Augmented Reality in Aircraft Manufacturing Processes

Jessica Abele

Abstract— **This work studies the usability of Augmented Reality (AR) in aircraft manufacturing processes based on a commodity monocular handheld/head-mounted camera. The application aims to assist the assembly operator in the installation work-flow of aircraft parts. However, for this use-case well known pure marker-based approaches are less favorable due to the prior complex preparation of the scene. A webcam is used as an input for two separat AR systems. One of which detects markers, whereas the other detects natural features. This research aims to combine both systems into a semi-marker-based AR approach. The core task is the registration process between a feature-based AR system and a marker-tracking system to extend the feature-based system with global reference and scale. The reference of the feature-based system is defined during the initialization step and changes within every application. Therefore a global reference is missing. Furthermore a stereo setup is assumed, which does not provide the system with an accurate scale. With the constraint of an initial marker in a feature-based system the robustness can be enhanced and yet still retains its main independence of reference.**

To evaluate the overall performance of the proposed hybrid AR system different test scenarios were recorded with various movement patterns and lighting-conditions. By doing this the efficiency of the Marker-Tracking-System was evaluated and the dependency of the implemented algorithm on the choice of measurements.

I. INTRODUCTION

This research topic contributes to the area of Computer Vision. The primary objective is to connect two systems built on different conceptions of Augmented Reality. Both systems are visual tracking systems affected individually by various limitations.

The focus of this research is the open-source marker-free system PTAM which shows good performance in stable environments built for monocular galvanic moving cameras, similar to those of head-mounted or hand-held cameras. The system in its native form tolerates an inaccurate scale based on a stereo assumption applied during its initialization step. Furthermore the camera estimation is only defined relatively to a coordinate frame set during the initialization. The main goal is to provide the system with the missing scale information, to make it applicable in an installation process typically found in aerospace technology.

By combining both systems the disadvantage of the marker-free solution (no reference in the global world) is countered by the marker-based one. Therefore the final system will benefit from the accuracy of the marker-tracking system and the independence from prior installations which is provided by the native PTAM system. The problem of finding the scale and global reference information can be seen as an absolute orientation problem, described by Berthold K. P. Horn [7]. The module designed to accomplish this task is the registration interface, that either registers the PTAM feature map or every camera pose generated by the PTAM system independently from the system. However, the second strategy is expected to be predominant due to error accumulation which arises when integrating in PTAM.

II. STATE OF THE ART

A broad range of applications which combine real environments with computer generated content can be categorized as Augmented Virtuality (AV) or Augmented Reality (AR). In 1994 Paul Milgram and Fumio Kushino introduced the concept of Mixed Reality (MR) [12], which is shown in Fig. 1. The users subjective experience is placed "anywhere between the extrema of the virtuality continuum" [12]. The term AV is nowadays seldomly used and AR is often used as a synonym for MR.

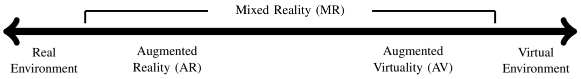


Fig. 1: Virtuality Continuum defining Mixed Reality [12].

Although the concept of AR is a young discipline the related research is extensive, as multiple algorithms and strategies originate from the field of Computer Vision and Robotics.

A. Tracking Sensors and Approaches

Before a virtual element can be placed in a real environment, the AR system must track the user's (at least relative) movement in the scene. The respective pose should be provided with 6 Degrees of Freedom (DoF). Three of them are used for positioning (x,y,z) and the others are used for orientation (roll,pitch,yaw). The problem of determining this camera pose is often defined as pose estimation or pose tracking. Techniques used in tracking can be subdivided into three groups: sensor-based, vision-based and hybrid methods.

Vision-based tracking is the most relevant area in regards to this research work. It uses image processing techniques to estimate the camera pose relative to the real scene. This area is divided into two main disciplines [16]: feature-based (ad-hoc) and model-based (a priori knowledge). The feature-based approach can be further divided as AR distinguishes between two sub strategies, those using features extracted from a maker as reference and those working solely on natural features.

Applications in which physical markers (known as *fiducial* markers) are placed in an environment in order to be detected by the AR system, are referred to as marker-based approaches. These systems have been gaining popularity in recent years, due to their easy implementation through available toolkits such as ARToolKitPlus [15], the successor of ARToolKit or ARTag [6]. The methods in [13], [9], which propose square planar fiducial markers have become popular methods for camera pose estimation in AR, because of their robustness, low-cost and real-time characteristics.

Marker-less strategies operate in a completely unknown environment. Within the area of AR, no marker-less solution has yet prevailed. Existing marker-less approaches are far more complicated and still less robust than the marker-based alternatives. However, they are more robust towards occlusions. The marker-less systems usually build maps storing the detected features. Two basic approaches are common in marker-less AR: extensible tracking and simultaneous localisation and mapping [14].

A linked tracking and mapping is less favourable for handheld cameras for multiple reasons, e.g. updating the whole map is expensive and data association errors accumulate into the map over time. To improve the process, Davison proposes to use a sparse map containing only high-quality features [3] and later an approach for monocular hand-held cameras [4]. Parallel Tracking and Mapping (PTAM) introduced by George Klein and David

This work was supported by FADA-CATEC

Antidio Viguria head of Department for Simulation and Software, FADA-CATEC, 41309 Sevilla, Spain aviguria@catec.aero

Université de Bourgogne (Le Creusot, France), Heriot-Watt University (Edinburgh, Scotland) and Universitat de Girona (Girona, Spain)

Murray [10] combines the idea of natural feature tracking and map building in 3D. This approach uses keyframes to obtain low quality features stored in dense maps instead of using every video frame. The decoupled extensible tracking and mapping approach enables the computation of expensive bundle adjustments to refine the map continuously.

B. Registration Problem

To register an AR system in the ”global” reference frame a registration method is needed. Different kinds of strategies can be found in literature. Some rely solely on geometrical characteristics, others use a least-squares method as a linear solving strategy, furthermore there exist iterative methods or non-linear optimization approaches. The authors of [5] provide a comparative analysis of four efficient and common algorithms for estimating 3-D rigid transformations. However, none of the analysed algorithms showed a salient performance in all evaluation tests. Based on this comparison the research work applies the Horn method to solve the registration problem. The presented algorithms use either Singular Value Decomposition (SVD) from an orthonormal matrix [1], unit quaternions [7] or polar decompositions [8] to represent rotations.

III. GENERAL FRAMEWORK

A. PTAM

Parallel Tracking and Mapping, known as PTAM [10], is especially designed to track the motion of a hand-held camera in small AR workspaces. The motion tracking runs parallel to a mapping thread which builds an extensible 3D map of point features (e.g Fig.2). Whereas most AR systems require initial knowledge of the working scene, PTAM avoids dealing with the typical limitations of these approaches by using natural features exclusively.

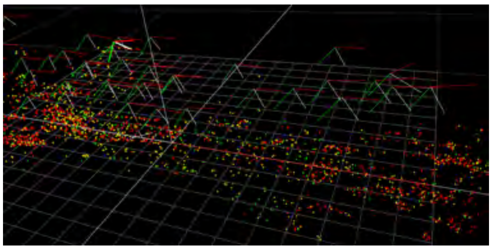


Fig. 2: A section of the initial map of 761 points and the two initial keyframes is extended to 3761 points and 46 keyframes. The grid on the ground is the dominant plane.

1) Mapping: The mapping thread is organised into two tasks. First a stereo-vision step is performed to obtain the initial map, which can be expanded and optimised for new keyframes. The PTAM initialisation uses two keyframes, selected manually by the user. From the first keyframe FAST corners are extracted and patches around these corners are instantiated. Through the patch tracking system the corresponding points in the second keyframe are found. Via five-point algorithm and RANSAC the essential matrix can be estimated for triangulation of the initial map, which is optimised through bundle adjustment. In general the bundle adjustment runs with low priority due to its computational complexity. Once the initial map is established a virtual ground plane is estimated using the map points. The best plane which is located at $z = 0$ of the world coordinate frame is found with RANSAC using a three point consensus set. Subsequently computer rendered objects dependent on the AR application are aligned with the real environment based on this virtual grid plane.

2) Tracking: The tracking can be seen as a two step algorithm, starting after the stereo-initialization. For every new frame a first rough camera pose is estimated for an assumed motion model. Based on this pose the map points are projected into the image plane. The tracking uses the Pyramid Levels and searches only for a small number of correspondences of the coarsest feature points. For these matches the camera pose can be updated with a more accurate estimation. Subsequently a fine search of a large number of points leads to a more accurate pose estimation.

B. EasyAR

To encode and decode the fiducial markers within the EasyAR application the hamming code technique is applied. There are two main matrices involved in handling hamming code patterns. The code generator matrix is used for construction and the parity check matrix is applied in the identification step.

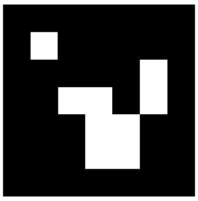


Fig. 3: Hamming coded fiducial marker (zeros in black, ones in white)

The specific hamming coded marker which is used for testing the system is presented in Fig.3. It has a size of 7×7, where the coded part measures 5×5. The real size of the marker used in this application after printing amounts to $4cm \times 4cm$, which refers to $266px \times 266px$.

1) Marker Generation: The marker is generated based on an incoming decimal ID, which is converted into a 12-bit integer. From this binary vector with help of the hamming code matrix, the hamming words are generated that can be converted into a fiducial marker. The result is given in Fig.3. Within the chosen implementation of the hamming code marker the white corner (1,1) of the inner square is used to identify the markers orientation.

2) Marker Identification: The identification step can be seen as the inverse strategy to the marker construction idea. Once the sub-image containing the marker is extracted from an incoming binary image it is divided into the 7×7 sub images in order to obtain the hamming code matrix. Each of the 49 squares is converted into one binary value in respect to its dominant pixel value. These candidates need to pass some test criteria before they can traverse the correction step by the parity check matrix.

IV. METHODOLOGY AND REALISATION

A. Combining EasyAR with PTAM

Since the real scale in the feature-based tracking approach PTAM is not known and to receive a more reliable orientation than obtained by the stereo-initialization, a marker-based system is introduced into PTAM. Global reference frames received by a precise tracking system allow PTAM to either register its feature map to the world frame or to transform the PTAM generated pose separately - without influencing the map. Both strategies are discussed.

PTAM provides a feature map in which a pose is defined by 6DoF, in which 3DoF refer to translation and 3DoF to rotation. An additional optical tracking system, such as a marker-based tracking system, can provide the missing scale information (the 7th DoF). This problem is known as the problem of absolute orientation. Originally PTAM assumes a translation of $10cm$ during the initialization in order to process scale information which leads to errors in the feature triangulation and therefore mainly in the z-axis estimation. The major drawback of PTAM is that the origin and orientation of its system depends on the features that are detected during the stereo initialization. Therefore the quality of the system is generally determined by the user and his ability to accomplish the initialization step. Furthermore the origin and the orientation change with every initialization.

The extension proposed by this paper is either adding further input parameters to the PTAM system, which will be considered as an ”internal” approach or in leaving PTAM as it is, solving the registration as an ”external” approach.

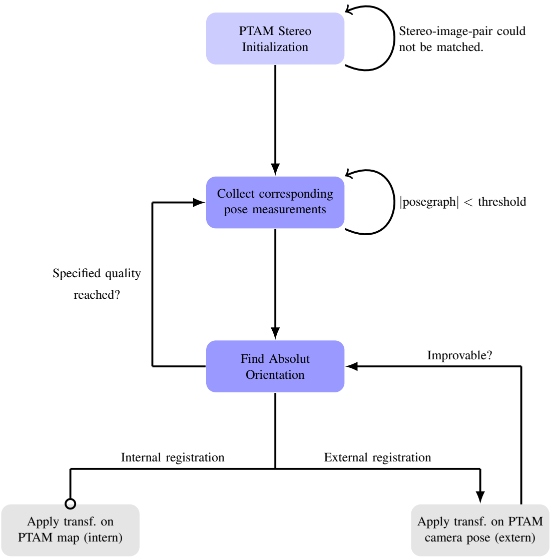

PTAM Stereo Initialization
Stereo-image-pair could not be matched.
Collect corresponding pose measurements
|posegraph| < threshold
Specified quality reached?
Find Absolut Orientation
Improvable?
Internal registration
External registration
Apply transf. on PTAM map (intern)
Apply transf. on PTAM camera pose (extern)

Fig. 4: Workflow of the registration interface.

- *Internal approach*
  Along with the intrinsic parameters the computed scaling factor, the rotation matrix and the translation vector are added to the PTAM process. These transformation factors need to be applied to the PTAM map and the current camera pose. It is important to ensure, that PTAM does not track the camera based on the inconsistent map, during the registration process. There are two approaches to solve this issue. The first method stops the tracking for the incoming keyframes during the registration time-slot and the second approach saves the old map before the registration process is initiated. By doing so, the tracking can continue using the old map and switches automatically to the new map, once it is fully registered. This should be adhered to, to prevent the PTAM tracker from loosing the reference and consequently triggering the recovery mode. In this case the registration is only done once in the PTAM system. Afterwards the PTAM system estimates the camera poses relatively to the global frame.
- *External approach*
  Alternatively an external approach could be chosen. In this case, the PTAM map remains as a relative map to its initial coordinate system. Only the estimated camera pose for each image frame is prompted by the main system and transformed respectively. This method applies a registration optimization strategy and the PTAM system does not further increase in its complexity.

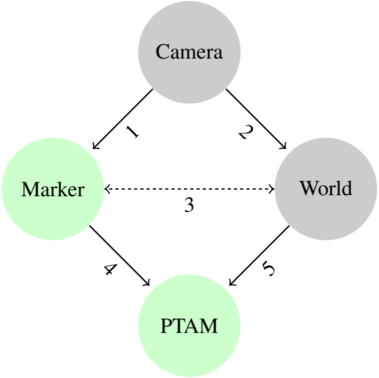


Fig. 5: Transformation-Relation-Graph of the registration problem.

Fig. 5 illustrates the relationship of the different coordinate systems involved in the registration problem. Each node within the graph can be seen as a cartesian coordinate system, an arrow as the transformation between two different coordinate systems.

The goal of the marker-tracking system (EasyAR) is to determine the transformation ${}^{W}T_{C}$ from the ”Camera” to the ”World” coordinate frame, represented by arrow 2, as well as its geometrical representation: ${}^{W}T_{C} = {}^{W}T_{M}{}^{M}T_{C}$. The marker is positioned in the ”World” coordinate frame and it is assumed, that its position and orientation in the ”World” is known up to a negligible error dependent on the tool used to measure the marker pose. Therefore the transformation ${}^{W}T_{M}$, described by arrow 3, is known. To conclude on the missing transformation between the marker and the camera ${}^{M}T_{C}$, EasyAR recognises the 4 marker corner points and identifies the ID of the observed fiducial marker. Knowing the 2D marker corners detected in the camera image and their corresponding global 3D locations the transformation from the camera to the marker can be estimated.

While the upper part of the graph keeps the coordinate system relationships in EasyAR, the lower part illustrates the integration of the marker-tracking-system into PTAM. Recalling the main target of the marker tracking system - the registration of the PTAM map into the "World" coordinate system - the overall goal of the project is to find the transformation described by arrow 5 and its geometrical representation: ${}^{W}T_P = {}^{W}T_M {}^{M}T_P$. The PTAM system keeps a map of cloud points, where 4 of these cloud points correspond to the marker corner coordinates. Through the stereo initialisation step a PTAM pose in respect to the fiducial marker is estimated. The transformation can be described with ${}^{M}T_P$.

B. Absolute Orientation

The Horn method [7] describes a method to find the absolute orientation for two sets of corresponding measurements in two different coordinate systems. The absolute orientation addresses the rigid-body transformation recovery between the two systems based on these measurements. It is a closed-form optimal solution considering all point measurements without applying outlier detection. In this work the handling of outlier measurements is constrained beforehand and separated from the absolute orientation algorithm.

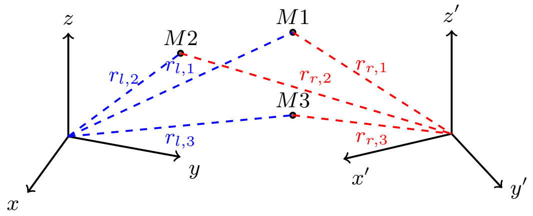


Fig. 6: Absolute Orientation Problem based on a minimum of 3 measurements M1, M2, M3.

In the Fig.6 the measurements $M_1 ... M_n$ and their references to each coordinate system $\{r_{l,i}\}$, $\{r_{r,j}\}$ are illustrated. In the following i, j are indices from $1 - n$ measurements, where $n \geq 3$.

The general formulation of the absolute orientation is given by the transformation from the left to the right coordinate system.

$$r_r = sR(r_l) + r_0 \tag{1}$$

The scale factor is described by s, $R(r_l)$ defines a rotated measurement from the left coordinate system and the offset is defined by r_0 .

This transformation can be split into a translation and a rotation plus a scale. It can be stated that the Horn method needs at least 3 corresponding pose measurements to solve the transformation. Three 3D points (x,y,z) provide 9 equations to find the 7 unknown parameters. Due to errors in a point measurement, the Horn method is not able to find a perfect transformation between the two coordinate systems. Therefore it is appropriate to involve more than 3 points. In addition the term in Equ.1 apparently will not suit all measurements. The objective is to minimize the sum of errors as shown in Equ.2.

$$\sum_{i=1}^{n} \|r_{r,i} - sR(r_{l,i}) - r_0\|^2 \tag{2}$$

All measurements are represented by their relation to the center point of each related cloud. In order to calculate the relative coordinates the centroids $\bar{r_l}$, $\bar{r_r}$ are computed first. A measurement is kept in its relative location to the centroids: r'_l, r'_r.

Horn describes three possibilities to find the appropriate scale between the two systems. The symmetrical term (Equ.3) is used since it can be computed independently from the rotation and translation.

$$s = \sqrt{\frac{\sum_{i=1}^{n} \|r'_{r,i}\|^2}{\sum_{i=1}^{n} \|r'_{l,i}\|^2}} \tag{3}$$

Rotations are described by unit quaternions (Hamilton). The matrix M contains the sum of the product S of pose measurements from the left with the right coordinate system. From this matrix the symmetrical matrix N is computed as a linear combination of M.

$$N = \begin{bmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yz} & -S_{xx}+S_{yy}-S_{zz} & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & -S_{xx}-S_{yy}+S_{zz} \end{bmatrix}$$

The unit quaternion refers to a unit vector in the direction of the Eigenvector corresponding to the highest Eigenvalue λ_{max} of matrix N.

As a final step of the absolute orientation algorithm the translation vector is computed, which is dependant on the rotation matrix and the scale factor.

$$r_0 = \bar{r_r} - sR(\bar{r_l}) \tag{4}$$

V. Results

To illustrate the registration process and the significantly different scale of the native 3D measurements taken either from EasyAR or PTAM the following 3D plots are provided (Fig.7, Fig.8). Fig.7 shows the measurements before applying the transformation on the PTAM measurements, and in Fig.8 the measurements are closely aligned due to the applied scale, translation and rotation on the PTAM measurements.

In Fig.9 and Fig.10 the AR visual result is shown. In Fig.9 the virtual element is aligned to the PTAM ground plane (before registration) and in Fig.10 it is globally aligned with a small rotational error. Fig.11 and Fig.12 reflect the measurement distribution in respect to their Distance To Marker (DTM). Tests have shown, that a better distribution with a small DTM performs best.

Throughout the experiments and development the hybrid system appeared sensitive in multiple dimensions. A good camera calibration ensures a higher accuracy of both systems and therefore more precise measurements for the registration process. The image resolution plays a further role in both systems.
Both systems add an uncertainty to the final hybrid system. Therefore a partly detached or bent marker might have a big influence

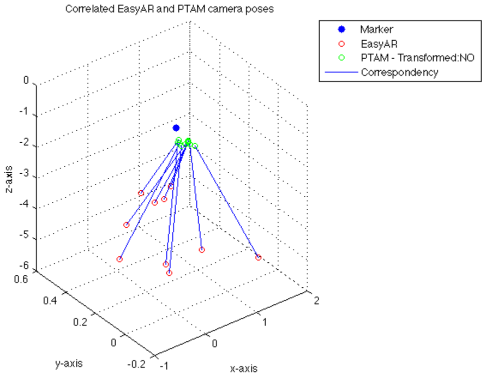


Fig. 7: Native correspondent 3D measurements taken from the PTAM system and EasyAR.

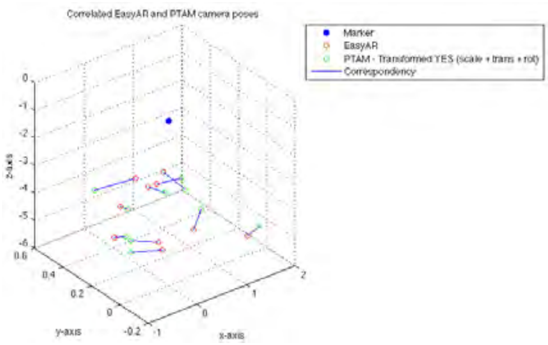


Fig. 8: 3D corresponding measurements after the alignment of PTAM using the absolute orientation transformation on the PTAM measurements.

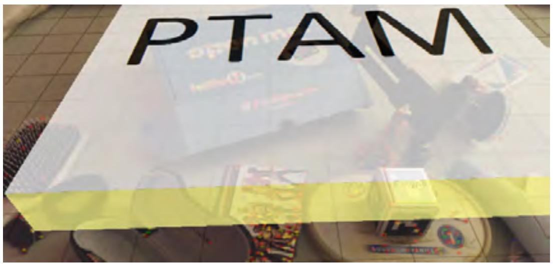


Fig. 9: PTAM virtual element fixed on the dominant plane defined in PTAM before the registration takes place.

in terms of pose estimation.

Furthermore the scale and quality of the marker are relevant. In general the detection errors within the pose calculation of EasyAR are higher in the direction of the z-, than in the x/y-translation based on the camera geometry. Similar to EasyAR, PTAM is a source of error, especially in the z-dimension. The system can be corrupted if the setup changes during the application or the environment contains repetitive patterns which can lead to location estimation jumps. In terms of registration, corresponding poses are not always available. An inexperienced user may not identify the best strategy of measurement selection which eventually results in a poor measurement distribution.

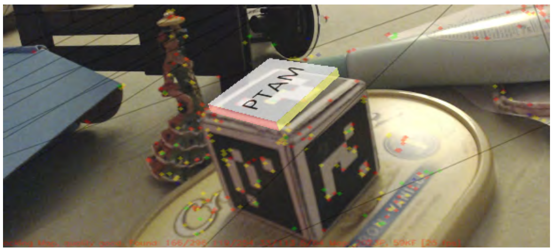


Fig. 10: Good PTAM alignment even in case EasyAR can not recognize the marker (in the image caused by reflections).

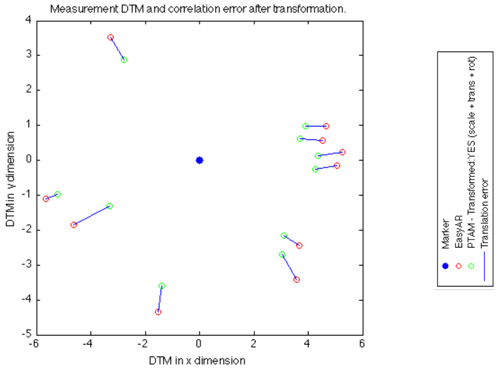


Fig. 11: DTM for SETUP 1. Translation error between the registration measurements after applying the transformation on the PTAM measurements: 1.25 (avg)

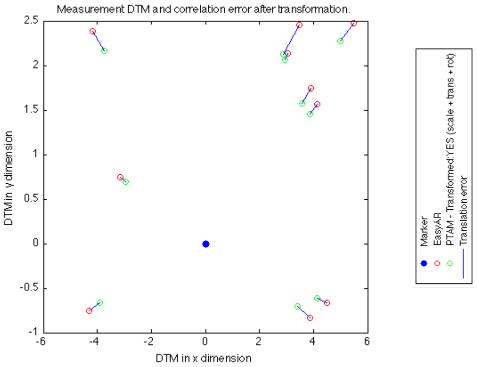


Fig. 12: DTM for SETUP 2. Translation error: 0.52 (avg)

VI. CONCLUSION

This research work presents a hybrid visual-camera-tracking strategy that combines the markerless-visual-tracking module PTAM with global reference information by introducing a single maker in the scene. The restriction introduced by the marker is reasonable due to the benefit it entails for industrial applications, where a global scale, position and orientation are imperative. Only the registration process requires the marker, afterwards the PTAM system proceeds without it. Experiments have shown that in robust maps, the marker can even be removed from the scene, once the initial map is build. However, in small maps, which cover only the regions close to the marker and the marker itself, the marker removal leads to tracking failure or complete loss of reference. The final solution does not require a preparation of the environment prior to the application, except for the marker installation. PTAM constructs its map on-the-fly. Therefore the proposed approach is suitable for industrial processes, which frequently change their setup.

VII. Future Work

Several ideas on possible extensions can be addressed. Linking PTAM with additional sensors might be one of them (fusion strategy). Combining PTAM with an accelerometer would replace the motion estimation by true acceleration values which leads to more robustness and higher speed in tracking, since it decreases the area to search for features.

To increase the flexibility of the registration process it would be interesting to work without a marker-system. This could be accomplished by using single positions with known global coordinates as references instead of the estimated camera pose. These features could be manually selected by the user in the PTAM feature map and the correspondent feature needs to be recognized in the current image frame.

Obviously the quality in terms of robustness and accuracy of the marker-tracking system *EasyAR* could be optimized by using more than one marker, more points per marker or in addition even natural features. A multi-marker system combines the information of different markers which can cope with partly occluded markers.

Furthermore the PTAM version, used in this research, only performs with an acceptable accuracy if applied in small workspaces and shows significant issues for 360° camera movements. At the time of this research there are two further developments of PTAM discussed: [11] (edgelets for motion blur), [2] (a multi-map framework), which could be of interest to optimize the performance and flexibility.

Finally it should be stated, that technical changes in terms of the equipment (e.g camera) and marker installation can also lead to tracking improvements. The use of a camera with higher resolution or a wide view objective could result in significant improvements in PTAMs feature detection process and also for the camera pose estimation on behalf of EasyAR.

VIII. Acknowledgments

First and foremost I want to express my gratitude towards my parents without whom this opportunity would not have been given to me. Furthermore I want to thank all the open-minded people at CATEC. Especially my supervisor Antidio Viguria, Marco Antonio Escalera Castro and Raul Quesada Pegalajar. Their guidance, support and cultural openness made me enjoy my working experience at CATEC. Looking back at two exciting years within the VIBOT Erasmus Mundus Program I am thankful for all of my fellow students and the nice people that make this program possible. Also special thanks extend to my two careful proof readers Sebastian Kirk and Christian Kniep.

References

[1] K.S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(5):698–700, Sept 1987.

[2] R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *Proc 12th IEEE Int Symp on Wearable Computers, Pittsburgh PA, Sept 28 - Oct 1, 2008*, pages 15–22, 2008.

[3] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410 vol.2, Oct 2003.

[4] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, June 2007.

[5] A. Lorusso D. W. Eggert and R. B. Fisher. A comparison of four algorithms for estimating 3-d rigid transformations. In *In Proc. British Machine Vision Conference*, 1995.

[6] M. Fiala. Artag, a fiducial marker system using digital techniques. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 590–596 vol. 2, June 2005.

[7] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[8] Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOURNAL OF THE OPTICAL SOCIETY AMERICA*, 5(7):1127–1135, 1988.

[9] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top AR environment. In *IEEE and ACM International Symposium on Augmented Reality, 2000.(ISAR 2000). Proceedings*, pages 111–119, 2000.

[10] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234, Nov 2007.

[11] Georg Klein and David Murray. Improving the agility of keyframe-based SLAM. In *Proc. 10th European Conference on Computer Vision (ECCV'08)*, pages 802–815, Marseille, October 2008.

[12] Paul Milgram and Fumio Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, E77-D(12), December 1994.

[13] J. Rekimoto. Matrix: a realtime object identification and registration method for augmented reality. In *Computer Human Interaction, 1998. Proceedings. 3rd Asia Pacific*, pages 63–68, Jul 1998.

[14] S. Siltanen and Valtion teknillinen tutkimuskeskus. *Theory and Applications of Marker-based Augmented Reality*. VTT science. 2012.

[15] Daniel Wagner and Dieter Schmalstieg. Artoolkitplus for pose tracking on mobile devices, 2007.

[16] Feng Zhou, H.B.-L. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*, pages 193–202, Sept 2008.

Evaluation of Bias Field Correction Methods on Breast MRI

S M Masudur Rahman Al-Arif

***Abstract*— Cancer is the most deadly disease in the world. The diversity of cancer is immense. In women, breast cancer is the most prominent cancer. One out of four female cancer patients is diagnosed with breast cancer. Mass screening programs have been initiated in many countries in order to detect breast cancer in early stage. Data show that, if breast cancer is detected in premature stage, the five year survival rate approaches 100%. Magnetic Resonance Imaging (MRI) is one of the best imaging modalities to detect breast cancer in terms of sensitivity and specificity. However, due to cost, time and lack of expert radiologists, MRI is not used as a primary device. Automatic image analysis techniques can be used in order to help the radiologist for reading and interpreting MRI images. In this work, we address an inherent problem of MR images called 'bias field' effect that stands in the way of any image processing application. We compare performances of state of the art fully automatic bias field correction methods on breast MRI. We have carried out evaluation in terms of novel and existing metrics. A ranking of the algorithms is made based on their performance in removing bias field from the image. A critical trade off between bias field removal and tissue distortion is found. Best algorithms have been identified based on this trade off. Finally, the quantitative result is also verified by qualitative analysis.**

I. INTRODUCTION

In women, breast cancer is by far the most prominent cancer and has the highest death rate. Statistics show that, in 2012, 1.7 million women were diagnosed with breast cancer and another 6.3 million had already been diagnosed with breast cancer in the previous five years. Since 2008, breast cancer incidence has increased by more than 20%, while death rate has increased by 14%. It represents one in four of all cancers in women [1]. If the tumor is still confined to the breast at the time of detection and has not spread through lymphatic system, five year survival approaches 100% [2]. MRI has a higher sensitivity than mammography and ultrasound. Its sensitivity varies from 89 to 100% over all ages where sensitivity of mammography falls to 40-50% for women with dense breasts. The MRI specificity varies from 82-96% [3], [4]. Although MRI provides acceptable sensitivity and specificity values, it is only used for reevaluation purposes in mass screening programs. The reason lies in the cost and time needed for evaluation. Screening using MRI involves large reading times compared to mammography, which would imply the need of more trained radiologists. Therefore, it is important to provide computer algorithms to automatically analyze breast MRI to aid in the interpretation and the detection of breast cancer. The first problem in any computer aided analysis of MR images is the high intra-tissue variability. The intensities measured from the same tissue smoothly vary across the image. This problem occurs due to practical limitations of the MRI scanners. The most visual effect can be seen in the fatty tissue inside a breast MRI and the difference in the intensity is usually severe in different part of the breast. An example of this is shown in Fig. 1. It is very crucial to correct this non-uniformity before processing the MRI for further image analysis. This non-uniformity of the signal intensity is also called bias field. Most of the existing research on bias field have been focused on brain MRI and many algorithms exist for this type of images. However, there is only a little work on applying or developing bias field correction algorithms to breast MRI. Moreover, brain and breast bias fields are different in nature, shape and smoothness, mainly due to differences in the coils used for image acquisition.

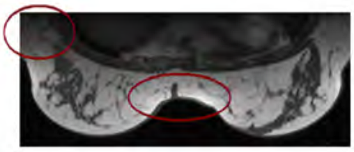


Fig. 1: Inter-tissue variability inside fatty tissue

In this work, we compare state of the art bias field correction methods for breast MRI with both quantitative and qualitative results on a dataset of 53 breast MRI scans. A novel evaluation metric is introduced which correlates with the qualitative results. This paper is divided into five sections. The main issue, its necessity and effects have been discussed in the introduction. A more detailed description of the problem, literature review and all the bias field correction methods are discussed in the Background (see Sec. II). In Sec. III, Methodology, evaluation procedures, metrics and base of comparison is discussed in details. All the results for all the cases has been summarized and discussed in Sec. IV. Finally, conclusions are given in Sec. V.

II. BACKGROUND

A huge number of bias field correction algorithms have been proposed in the last decade. Based on the initial approach, all the methods can be classified into two broad classes: prospective and retrospective. The classification tree is shown in Fig. 2 and it is based on the work of Vovk et al. [5]. While

This work is submitted for the degree of MSc Erasmus Mundus in Vision and Robotics (VIBOT)

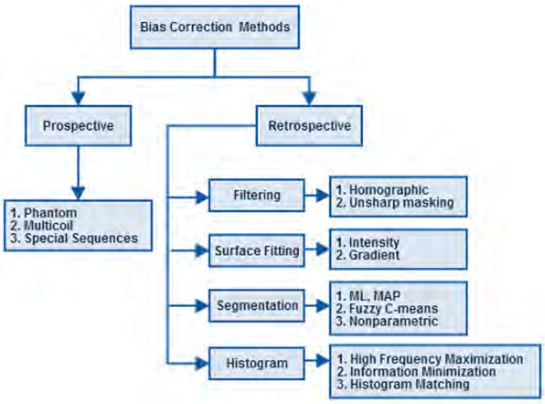


Fig. 2: Classification of bias field correction algorithms

all prospective methods require interaction with the MRI devices, retrospective methods are more general and more related with the image analysis research. These methods only use the acquired image to estimate the bias field in an MRI scan [6], [7]. Retrospective methods are further divided into four types: filtering-based methods, surface fitting-based methods, segmentation-based methods and histogram-based methods. Filtering-based methods assume the bias field to be a low frequency entity and removes it by low-pass filtering of the image. However, these methods inherently assume that no useful information is present in the low frequency band, which does not hold for most of the scanned anatomical structures; e.g., fatty tissue for breast MRI. Thus, the applicability of this method is limited [8].

Surface fitting methods do not apply the principle of low-pass filtering. Instead, surface fitting methods fit a parametric function corresponding to a set of image features that contain information on the bias field. The surface is usually polynomial-based or spline-based. Based on the kind of image features used to compute the surface, these methods can be classified into intensity-based [9] or gradient-based [10].

Image segmentation often requires bias field correction for better performance, but perfect segmentation can make the bias field estimation very easy. Thus, these two procedures can be thought as complementary of each other. Based on this idea, a set of bias field correction methods are developed to iteratively segment and perform bias field estimation at the same time. These algorithms are further divided into classes based on their segmentation procedure,e.g., Expectation Maximization (EM)-based [11], FCM-based [12] and nonparametric Mean-Shift, Max-shift algorithm-based [13], [14].

Finally, the most commonly used methods are histogram-based methods. These methods directly work on image histograms and needs very little or no initialization. This property makes these methods fully automatic and highly general for all type of MRI. Although a number of segmentation-based methods also operate on image intensity histograms, the distinction between the segmentation-based and histogram-based methods is that the latter provide no segmentation results. The most popular of this class is the nonparametric non-uniformity normalization (N3) method [15]. Having in mind the application to breast screening programs, the correction algorithm should be fully automatic. Thus among all the works we only chose seven fully automatic algorithms from the state of the art to compare their results with each other. These algorithms are Mean-Shift (MS) bias field correction [14], [16], modified Fuzzy C-Means-based Bias field Correction (BCFCM) [17], sparseness of gradient-based Non-Uniformity Correction (NUC) [18], Non-parametric Non-uniformity Normalization (N3) [15], improved N3 (N4) [19], FCM segmentation-based Coherent Local Intensity Correction (CLIC) [20] and Level-Set image segmentation-based bias field correction (LevSet) [21]. Of these seven methods, MS, BCFCM, N3 and N4 were defined to correct the bias field in 3D image volumes, while the rest only work with 2D slices for now. Based on our mentioned classification, NUC is a gradient-based surface fitting method; MS, BCFCM, LevSet and CLIC are segmentation-based methods and N3, N4 are histogram-based methods.

III. METHODOLOGY

In this work we compare the results of applying seven different bias field correction algorithms on 53 breast MR scans. As not much work has been previously done on breast MRIs comparing the performance of bias field correction methods, novel techniques for evaluation are proposed. The evaluation mainly focuses on three visible tissues in breast MRI: fatty tissue, dense tissue and pectoral muscle (see Fig. 3). Manual segmentations of these tissues are used as masks for evaluation.

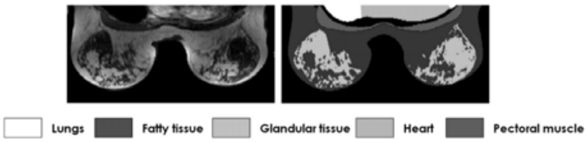


Fig. 3: Manual segmentation

A. Material

The data set used to evaluate the bias field correction results consists of 53 pre-contrast coronal T1-weighted MR breast volumes from 53 different patients. The age of the screened women ranged from 23 to 76 years (45.84 ± 11.97 of average). The cases were collected from 2003 to 2009. Breast MRI examinations were performed on either a 1.5 or 3 Tesla Siemens scanner (Magnetom Vision, Magnetom Avanto and Magnetom Trio), with a dedicated breast coil (CP Breast Array, Siemens, Erlangen). Clinical imaging parameters varied; matrix size: 256 x 128 or 256 x 96; slice thickness: 1.3 mm; pixel spacing: $0.625 - 1.25$ mm; flip angle: $8°$, $20°$ or $25°$; repetition time: $7.5 - 9.8$ ms; echo time: $1.7 - 4.76$ ms. Patients were scanned in prone position.

B. *Algorithms*

The evaluated algorithms are:

1) Mean-Shift (MS)
2) Modified FCM (BCFCM)
3) N3 bias field correction (N3)
   - Spline Fitting Levels 4 (N3 SF4)
   - Spline Fitting Levels 6 (N3 SF6)
4) Nick's Improved N3 (N4)
   - Spline Fitting Levels 4 (N4 SF4)
   - Spline Fitting Levels 6 (N4 SF6)
5) Sparseness of Gradient (NUC)
6) Level Set (LevSet)
7) Coherent Local Intensity Correction (CLIC)

Some of these algorithms are publicly available, some are collected through requests to appropriate authority. Implementations are in C++ and/or Matlab. The iterative framework for all cases is written in Matlab and all algorithms are called from this iterative framework. VTK, DCM and $NRRD$ file formats are used. Different parameters for the algorithms are empirically found and/or suggested by the authors to obtain good results. N3 and N4 are both evaluated with two different spline fitting levels.

C. *Evaluation Metrics*

Different metrics are considered for quantitative comparison of bias field correction algorithms. In previous bias field correction comparison studies [16], [22], [23], researchers used visual or qualitative results and Coefficient of Variation (CV) as quantitative measure. Visual results can be extremely hard to determine by human eyes and the metric CV also has disadvantages (see Sec. III-C.4). To overcome these limitations, in this work, along with CV, we propose and use different metrics:

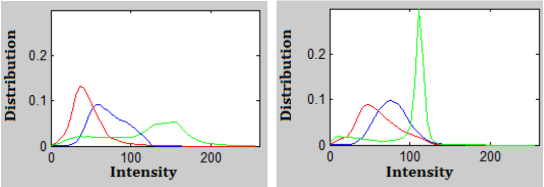
- Standard deviation (STD)
- Percentage Count (PC)
- Entropy (E)
- Coefficient of Variation (CV)
- Bhattacharyya Distance (BD)

1) Standard Deviation and Histograms: Bias field correction algorithms reduce the intensity variation within the same tissue. The standard deviation represents the spread of the intensity distribution of a certain tissue class. Therefore, after correcting the bias field, standard deviation of that tissue class should decrease. For a tissue class X with N voxels, standard deviation can be defined as:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \qquad (1)$$

where $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $x_i \in X$.

Fig. 4 shows normalized histograms of three tissues (dense tissue, pectoral muscle and fatty tissue) for a single case before and after application of bias field correction algorithm. It can be seen in Fig. 4 that dense tissue and pectoral muscles signal intensity distributions follow a unimodal Gaussian distribution both before and after application of bias field correction. However, the signal intensity distribution of the fatty tissue shows bi-modality. Performance of tissues with unimodal distributions can be easily measured by standard deviations, but standard deviation of a bimodal distribution is not useful. In order to measure the effect of the bias field correction algorithms other metrics are needed.



(a) Before correction (b) After correction

Fig. 4: Histogram of different tissues

2) Percentage Count (PC): Percentage Count (PC) is a novel metric proposed in this thesis. PC is a kind of local histogram-based metric which only computes the percentage of voxels having intensity near the peak histogram bin intensities. In other words, PC is the percentage of voxels present in vicinity of the peak histogram count. A range of 20 is chosen around the peak histogram count intensity. All the voxels that have intensities in that range are considered. Higher percentage count represents better result. Therefore, increase of PC from the original PC indicates improvement. PC is defined as:

$$Percentage\ Count(PC) = \frac{\sum_{i=x-\frac{r}{2}}^{x+\frac{r}{2}} P_i}{\sum_{j=1}^{N} P_j}, \qquad (2)$$

where P_i is the number of voxels having intensity level i; x is the approximate intensity which has maximum number voxels, N is the maximum intensity and r is the range. The range is empirically set to 20 for this work. Fig. 5 shows which part of a histogram is chosen to measure PC.

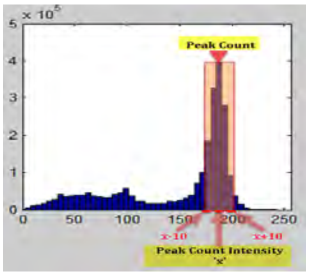


Fig. 5: Percentage Count

3) Entropy (E): Entropy is a statistical metric which represents the randomness of a distribution. As bias field correction reduces the dynamic range of a tissue, randomness of the tissue intensity distribution should also be decreased from the original image and such amount of decrease can be used as a metric. It can be expressed as:

$$Entropy(E) = \sum_{i=0}^{number\ of\ bins} \frac{bin\ count_i}{log_2(bin\ count_i)}, \tag{3}$$

where bin $count_i$ is the number of voxels counted in i-th histogram bin and $number\ of\ bins$ is taken as 50. While PC is a histogram-based local metric, entropy is a histogram-based global metric as the whole histogram is taken into account. Entropy has not been used before for evaluating bias field correction results.

4) Coefficient of Variation (CV): The third metric is Coefficient of Variation (CV). It has been used in [16], [23] to compare the performances of different algorithms. Mathematically it is expressed as:

$$CV(C) = \frac{\sigma(C)}{\mu(C)} \tag{4}$$

where σ and μ are the standard deviation and the mean of the tissue class C. CV quantifies the variation of tissue intensity and should be reduced after bias field correction. CV is invariant to multiplicative uniform intensity transformation (intensity scaling or contrast), which an intensity inhomogeneity correction method could introduce. A drawback of CV is its sensitivity to uniform additive intensity transformation (brightness), which changes the mean but not the variance of a tissue [5].

5) Bhattacharyya Distance: Variability between different tissue classes is very important for breast MRI analysis. Bias field correction algorithms should be able to increase the difference between signal intensity distributions of two different tissue classes. Fig. 4 illustrates how inter-class variability is affected by the bias field correction. In order to quantify the effect, Bhattacharyya distance [24] between the signal intensity distributions of fatty and dense tissue or pectoral muscle can be calculated. If p and q are two histograms for same intensity range X, then the Bhattacharyya distance between p and q is defined as:

$$D_B(p,q) = -ln(BC(p,q)) \tag{5}$$

where $BC(p,q) = \sum_{x \epsilon X} \sqrt{p(x)q(x)}$. Larger values of D_B indicate more separation between p and q.

D. Visual Inspection

Apart from the quantitative analysis, a qualitative comparison between the algorithms is also performed. In order to do this, a smaller number of cases are carefully chosen from the dataset. Of 53 cases, eight cases with strong bias field are selected. One slice from each view is chosen and given to 2 different experts for rating. Each inspector was asked to rate all the cases according to two different 5 point scale. One scale determines the amount of bias field present in a slice, the other scale determines the amount of distortion in tissues. The scales are shown in Fig. 7. Finally, based on the average rating, a ranking between the algorithms is made.

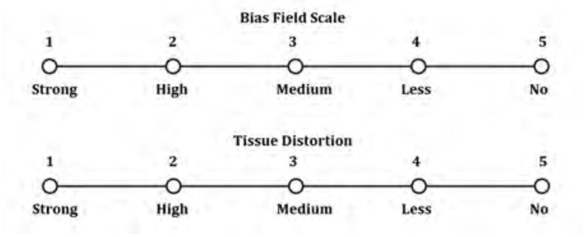


Fig. 7: Rating scales for visual inspection

IV. RESULTS AND DISCUSSIONS

The different bias field correction algorithms are evaluated in this chapter using dataset of 53 cases and the metrics explained in the previous chapter. To summarize the results, box plots are used. Box plot is a convenient way of graphically depicting groups of numerical data through their quartiles [25]. Box plots of the metrics on fatty tissue bias field removal are shown in Fig. 6. To recall the metrics from Sec. III-C, increase in PC and decrease in E & CV indicates better bias field removal. In terms of PC the performance is: $N4SF6 > LevSet > N4SF4 > N3SF6 > CLIC > N3SF4 > MS > BCFCM > NUC > ORG$. In terms of Entropy (E): $LevSet > N4SF6 > N4SF4 > N3SF4 > N3SF6 > CLIC > BCFCM > NUC > MS > ORG$. Ranking of algorithms in terms of CV is: $LevSet > N4SF4 > N4SF6 > CLIC > N3SF4 > N3SF6 > BCFCM > NUC > MS > ORG$. We can conclude that LevSet, N4's & N3's perform well in terms of bias field removal and NUC, BCFCM, CLIC & MS do not remove bias field much.

While bias field correction algorithms remove bias field from the image, it also cases distortion in other tissues. These distortion can be expressed in terms of standard deviations. The standard deviations of dense tissue and pectoral muscle before and after the application of algorithms are shown in Fig. 8. It can be seen that N4's, N3SF6 and LevSet, which perform good bias field removal, causes substantial increase in standard deviation for both tissues. N3SF4, MS, BCFCM, NUC and CLIC perform the best in terms of low distortion. N3SF4 is the only algorithms that performs well in both bias field removal and low distortion. Variability between tissue classes is a critical issue. This is also affected by bias field correction algorithms. Fig. 9 shows that all the algorithms except CLIC, NUC, BCFCM, N4SF4 and N3SF4 reduces the distance between fatty tissue and

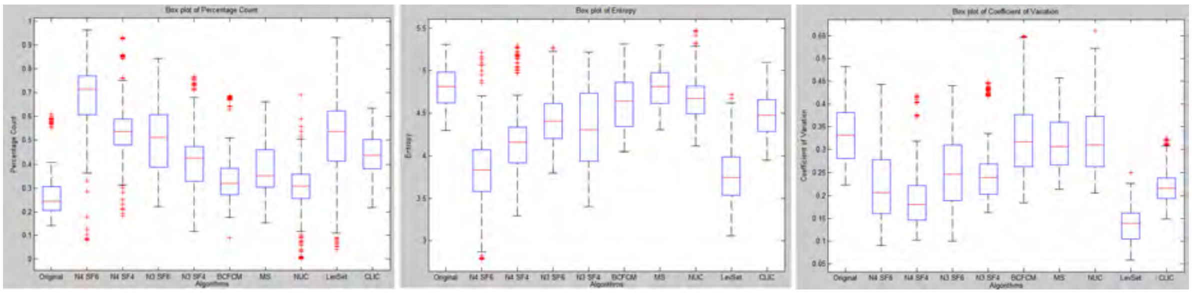
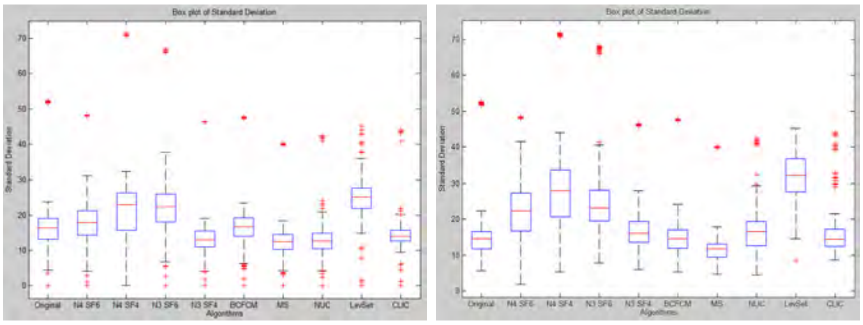


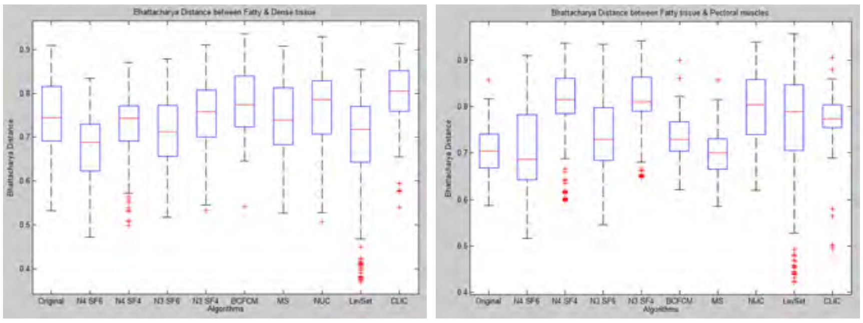
Fig. 6: PC (left), Entropy (middle) and CV (right) for all algorithms



(a)

(b)

Fig. 8: Standard deviations for (a) dense tissue and (b) pectoral muscle



(a)

(b)

Fig. 9: Bhattacharya distances from fatty tissue to (a) dense tissue and (b) pectoral muscle

TABLE I: Bias field correction scores

Methods	MS	BCFCM	N3SF4	N3SF6	N4SF4	N4SF6	NUC	CLIC	LevSet
Expert 1	2.33	2.33	3.05	3.16	4.37	4.21	1.5	2.75	4.5
Expert 2	3.70	3.75	3.38	3.38	4.75	4.95	2	3.75	5

TABLE II: Distortion scores

Methods	MS	BCFCM	N3SF4	N3SF6	N4SF4	N4SF6	NUC	CLIC	LevSet
Expert 1	4.08	4.08	4.08	3.67	3.91	3.2	4.13	4	3.12
Expert 2	4.29	4.25	4.38	3.54	4.41	2.38	3.87	3.87	2.87

dense tissue. The distance between fatty tissue and pectoral muscle shows that N4SF4, N3SF4, NUC, LevSet and CLIC increases the distance after bias field correction. After considering bias field removal, distortion and inter class distances, we can conclude that only N3SF4 performs well in all categories. Some of the resulted volumes were also sent to two experts for visual evaluation. Average scores for the algorithms are summarized in Table. I and II. If we consider score 3 to be the baseline score, then three algorithms unanimously outperform the baseline score in terms of bias field and distortion. They are: N3 SF6, N3 SF4 and N4 SF4. This trend is similar to what we observed during the quantitative analysis.

V. CONCLUSIONS

In this work, we have extensively studied the effect of seven different state of the art bias field correction methods on breast MRI. The evaluation process, which includes the metrics used, and a complete comparison of their performance have been described. The effect of bias field correction on tissue segmentation has been also investigated. The quantitative findings have been verified by blind qualitative analysis. A critical trade off between correction of bias field and tissue distortion has been found. Finally, the best algorithms have been identified based on this trade off.

Among seven algorithms, N4SF6 and LevSet methods obtain the best bias field removal results. The distortions caused by these algorithms are also very large. These algorithms also decreases the inter-tissue distances. On the contrary, MS, BCFCM, NUC and CLIC cause least amount of tissue distortions. MS and BCFCM also increase the distances between tissue classes. However, none of them correct the bias field completely. Considering the trade off, N3SF4 is found to be moderate in both ends. This algorithm also passed the baseline score in qualitative analysis by the two experts.

REFERENCES

[1] I. A. for Research on Cancer *et al.*, "Global battle against cancer won t be won with treatment alone," *Effective prevention measures urgently needed to prevent cancer crisis (Press Release No. 224)*, 2014.

[2] "National Insititute of Health. Breast Cancer.." `http://www.nlm.nih.gov/medlineplus/ency/article/000913.htm`, 2010.

[3] C. Kuhl, "The current status of breast mr imaging part i. choice of technique, image interpretation, diagnostic accuracy, and transfer to clinical practice 1," *Radiology*, vol. 244, no. 2, pp. 356–378, 2007.

[4] H. Vainio and F. Bianchini, "Iarc handbooks of cancer prevention: breast cancer screening," *Lyon: IARCPress*, 2002.

[5] U. Vovk, F. Pernus, and B. Likar, "A review of methods for correction of intensity inhomogeneity in mri," *Medical Imaging, IEEE Transactions on*, vol. 26, no. 3, pp. 405–421, 2007.

[6] H. Mihara, N. Iriguchi, and S. Ueno, "A method of rf inhomogeneity correction in mr imaging," *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 7, no. 2, pp. 115–120, 1998.

[7] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, P. Boesiger, *et al.*, "Sense: sensitivity encoding for fast mri," *Magnetic resonance in medicine*, vol. 42, no. 5, pp. 952–962, 1999.

[8] E. B. Lewis and N. C. Fox, "Correction of differential intensity inhomogeneity in longitudinal mr images," *Neuroimage*, vol. 23, no. 1, pp. 75–83, 2004.

[9] B. M. Dawant, A. P. Zijdenbos, and R. A. Margolin, "Correction of intensity variations in mr images for computer-aided tissue classification," *Medical Imaging, IEEE Transactions on*, vol. 12, no. 4, pp. 770–781, 1993.

[10] S.-H. Lai and M. Fang, "A new variational shape-from-orientation approach to correcting intensity inhomogeneities in magnetic resonance images," *Medical Image Analysis*, vol. 3, no. 4, pp. 409–424, 1999.

[11] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens, "Automated model-based bias field correction of mr images of the brain," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 10, pp. 885–896, 1999.

[12] J. C. Bezdek, L. Hall, L. Clarke, *et al.*, "Review of mr image segmentation techniques using pattern recognition," *MEDICAL PHYSICS-LANCASTER PA-*, vol. 20, pp. 1033–1033, 1993.

[13] B. Likar, J. Derganc, and F. Pernus, "Segmentation-based retrospective correction of intensity nonuniformity in multispectral mr images," in *Medical Imaging 2002*, pp. 1531–1540, International Society for Optics and Photonics, 2002.

[14] J. Derganc, B. Likar, and F. Pernus, "Nonparametric segmentation of multispectral mr images incorporating spatial and intensity information," in *Medical Imaging 2002*, pp. 391–400, International Society for Optics and Photonics, 2002.

[15] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, "A nonparametric method for automatic correction of intensity nonuniformity in mri data," *Medical Imaging, IEEE Transactions on*, vol. 17, no. 1, pp. 87–97, 1998.

[16] A. Makarau, H. Huisman, R. Mus, M. Zijp, and N. Karssemeijer, "Breast mri intensity non-uniformity correction using mean-shift," in *SPIE Medical Imaging*, pp. 76242D–76242D, International Society for Optics and Photonics, 2010.

[17] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriarty, "A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 3, pp. 193–199, 2002.

[18] Y. Zheng, M. Grossman, S. P. Awate, and J. C. Gee, "Automatic correction of intensity nonuniformity from sparseness of gradient distribution in medical images," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pp. 852–859, Springer, 2009.

[19] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee, "N4itk: improved n3 bias correction," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 6, pp. 1310–1320, 2010.

[20] C. Li, C. Xu, A. W. Anderson, and J. C. Gore, "Mri tissue classification and bias field estimation based on coherent local intensity clustering: A unified energy minimization framework," in *Information Processing in Medical Imaging*, pp. 288–299, Springer, 2009.

[21] C. Li, R. Huang, Z. Ding, J. Gatenby, D. N. Metaxas, and J. C. Gore, "A level set method for image segmentation in the presence of intensity inhomogeneities with application to mri," *Image Processing, IEEE Transactions on*, vol. 20, no. 7, pp. 2007–2016, 2011.

[22] J. B. Arnold, J.-S. Liow, K. A. Schaper, J. J. Stern, J. G. Sled, D. W. Shattuck, A. J. Worth, M. S. Cohen, R. M. Leahy, J. C. Mazziotta, *et al.*, "Qualitative and quantitative evaluation of six algorithms for correcting intensity nonuniformity effects," *NeuroImage*, vol. 13, no. 5, pp. 931–943, 2001.

[23] Z. Hou, "A review on mr image intensity inhomogeneity correction," *International Journal of Biomedical Imaging*, vol. 2006, 2006.

[24] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: The Indian Journal of Statistics*, pp. 401–406, 1946.

[25] Y. Benjamini, "Opening the box of a boxplot," *The American Statistician*, vol. 42, no. 4, pp. 257–262, 1988.

Ego-motion estimation with low-power processing using a smart panoramic compound camera

Gerard Bahi Vila, Ramon Pericet Camara and Dario Floreano

Abstract— **This work focuses on developing an ego-motion estimation method using light weight embedded sensors in GPS denied environments. The approach presented fuses inertial information and optic flow information to provide a velocity estimate which aerial vehicles can use for navigation in indoor cluttered environments. The solution proposed in this work is targeted for vehicles weighting less than 100 grams. Using the I2A algorithm, up to 20,000 optic flow vectors per second can be obtained from the sensor, which then perform voting to determine the direction of motion. This direction of motion estimate is used to correct the velocity estimate obtained through an extended Kalman filter using the inertial information.**

I. INTRODUCTION

In a scenario of a natural disaster there are many situations in which the use autonomous vehicles for exploring the area is much safer. One example of these situations is the earthquake that took place in May 2012 on the North of Italy [1].

Teleoperated flying and ground robots were used to analyze the state of half collapsed buildings. Flying robots were proven to have some advantages with respect to ground vehicles. They were capable of providing views from several attitudes and they are able to access areas regardless of stairs or wreck in their path. Even though teleoperated robots were proven useful, to drive the robots without any collision was found not only difficult but also stressful for the pilots, as it was known that the robots probably would not be able to recover from such a collision.

One possible solution in such a situation is to utilize flying robots with embedded sensors that are able to navigate autonomously and avoid collisions. In addition the flying platform should be as small as possible as this reduces the risk of collision and increases the areas the robot can reach.

For flying robots to navigate autonomously in GPS denied environments sensors have to be embedded in the robot so that ego-motion can be computed reliably. Ego-motion describes the amplitude and direction of the robot velocity vector. This information can be used in closed loop controllers to maintain a position or follow a trajectory.

Many solutions in the literature use monocular vision sensors in order to compute optic flow based ego-motion estimation. However these sensors present some ambiguity, as they are not able to distinguish between scale and speed changes as explained more in detail in [2]. In order to overcome this ambiguity, several solutions use inertial sensors and monocular sensors and they combine them using simultaneous localization and mapping (SLAM) [3], [4]. However, these SLAM based solutions usually require a large amount of computation and memory not suitable for the targeted platforms processing power.

Some other proposed solutions use feature tracking and matching. These methods require cameras with a resolution high enough to capture those features, and also enough computing power to compute matching between them. A survey of several techniques using these approaches can be found in [5]. The payload generated by the cameras needed makes these methods not suitable for the desired hardware.

There are also solutions which use optic flow based methods in order to find a direction of motion constraint using several optic flow vectors. Lim et al. [6] developed a method using catadrioptic devices and fish eye lenses whose image plane can be approximated by a sphere. These lenses, which provide a very large field of view, allow for getting translational optic flow data using information from antipodal points to cancel the rotational component of the optic flow [6]. Given the translational optic flow, an estimate of the direction of motion can be computed. In [6], two methods for doing so are compared: RANSAC and a coarse-to-fine voting algorithm. The last option has shown better performance with respect to accuracy and frame rate. This approach if changing the sensor for a more lightweight option it would be a suitable option for computing a direction of motion constraint. This approach would need to be combined with a solution able to provide the magnitude of the velocity vector in order to define ego-motion.

Recently, methods that fuse optic flow and inertial sensor measurements have been developed. Kendoul et al. [7] use a 3 nested Kalman filtering approach which fuses visual and inertial information in order to obtain the velocity and the scene structure. Briod et al. [8] use an extended Kalman filter approach using accelerometers to integrate the velocity and position. For reducing the uncertainty and drift of the estimation, they use the direction of motion constraint given by the translational optic flow information obtained by high precision sensors.

The aim of this paper is to provide a method capable of estimating ego-motion using very lightweight sensors in order to assist autonomous navigation in GPS denied environments, adding the minimal amount of payload possible. The method is designed for modern micro-air vehicles which can only carry a limited amount of payload as well as a limited amount of computational power due to their small structure.

The paper describes the method as follows. Section II describes the sensor used for developing the method. Section III explains the details of the implementation. The outcome of the method is described in section IV followed by the conclusions in section V.

II. Curvace sensor

The compound eyes of insects are very efficient for local and global motion analysis over a large field of view (FOV). This feature makes them an excellent sensor for accurate and fast navigation in 3D dynamic environments.

The Curvace is a sensor which mimics the insect compound eyes [9]. For doing so the sensor is composed of micro-lens arrays integrated with adaptive photoreceptors. Using this technique, the sensor is able to provide a much larger field of view than conventional cameras and also a limitless depth-of-field.

Figure 1 shows the Curvace sensor developed. The sensor has a volume of 2.2 cm^2, a weight of 1.75 g and consumes 0.9 W at maximum power. The resolution of the sensor is 15 rows by 42 columns. What is more it also contains a gyroscope and an accelerometer in order to provide inertial information.

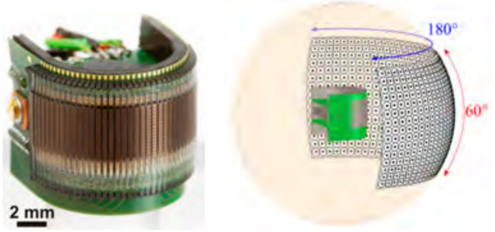


Fig. 1: Representation of the Curvace sensor. Left: an image of the prototype. Right: Representation of the FOV of the sensor. [9]

III. Approach

The method developed should extract optic flow information from the Curvace sensor. The optic flow data is then combined as in [6] to output a direction of motion estimate which is robust to noise from the optic flow sensor. The estimate found may then be used to provide ego-motion estimation in a way similar to in [8] but with the direction of motion estimate used in place of high-precision optic flow sensor readings. This method provides an ego-motion estimate that has a sensor payload of 2 grams.

A. *Optic flow extraction*

The method used for extracting optic flow from the sensor is the Mandyam V. Srinivasan method [10], commonly known as I2A. The code used is an assembly language implementation of this method programmed by Andreas Steiner [11].

The code was modified in order to be able to use the data given by the Curvace sensor, and also to make it capable to support different window sizes. This implementation was chosen since it can use architecture-specific assembly calls of the microchip, making the algorithm faster than any other implementation in C, thus allowing a higher frame rate.

The method involves approximating the frame at time step n by computing an interpolation from the frame at time step $n-1$ shifted ± 1 units horizontally and vertically. The x and y shift which provide the interpolation of the current frame from the previous frame with the minimum error is the optic flow observed in the time step n.

Some implementations also take into account a diagonal shift, however the version described which only uses horizontal and vertical interpolation is widely used in the literature achieving good performance [12] [10].

In the original method Srinivasan in his work uses a 2D Gaussian window function. However, Taylor et al. [12] state that no advantage was found on using the Gaussian function instead a rectangular window which has clear efficiency advantages.

B. *Direction of motion estimation*

Once the translational optic flow is obtained, the next step is obtaining the direction of motion constraint. For doing so, firstly the image plane is approximated as a unit sphere. This constraint is actually very suitable for the Curvace sensor since the ommatidia of the sensor have spherical shape (see figure 1). As shown in the literature, having the image plane represented by a sphere has the advantage the translational component of the optic flow vector, given a position on the sphere surface, generates a great circle, whose plane contains the direction of motion [13].

In an ideal situation without noise, two optic flow vectors that are not contained in the same great circle would be enough to determine the direction of motion, as the intersections of the great circles would give the focus of expansion and the focus of contraction. The focus of expansion is going to be the point whose angle between it and the optic flow vectors is higher than 90°.

However, since the optic flow data is noisy, an algorithm to compute an estimate of the direction of motion is needed. Lim [13] proposed a coarse to fine voting strategy which outperforms other techniques such as RANSAC or least mean squares in terms of accuracy or frame rate according to the results found in the literature. The approach taken in this step is based on his work.

For performing a voting algorithm, firstly a voting space needs to be defined. Since the image plane is approximated by a unit sphere, spherical polar coordinates can be used. Using this coordinate system has the advantage that only voting on θ and ϕ is needed since r is always 1, reducing the voting space for the direction of motion which is a 3D vector to instead only two dimensions.

In order to determine if a bin is crossed by a great circle the method proceeds as follows:

- The great circle plane is defined by the normal vector of the plane, computed by doing the cross product between the position vector and the optic flow vector.
- The bin corners are converted into Cartesian space.
- The sign distance between the bin corners and the great circle plane is computed.

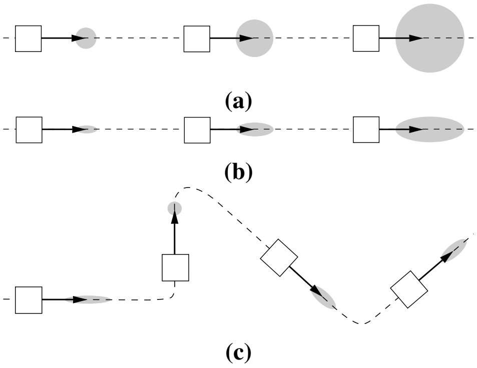
- If the sign of the distance is not equal for all the bin corners it means the great circle goes trough the bin so the vote for that bin is increased in one unit.

C. Ego-motion estimation

In this section, the sensor fusion method used for obtaining the velocity estimation is going to be described. The direction of motion estimate obtained from the visual information is fused with the accelerometers data using an extended Kalman filter. This approach is based on [8] where an extended Kalman filter is used to fuse inertial data with very precise optic flow measurements.

It has been stated in [14] and [8] that this approach keeps one degree of freedom which is unobserved, hence subject to suffer from drift. This degree of freedom corresponds to the direction of motion which means that it will change every time the direction of motion is changed, allowing to keep the velocity uncertainty bounded if sufficient movements are undertaken by the robot.

Figure 2 shows graphically how the velocity uncertainty changes when no direction of motion constraint is applied versus when it is applied, giving two different paths as example: one which follows a straight line and another one which keeps changing the direction of motion. As expected, the last case is the one that gives more reliability of the velocity measurement.



(a)

(b)

(c)

Fig. 2: 2D representation of the effect that a direction of motion constraint has over the velocity uncertainty when fused with inertial sensors. (2a) shows the case when no direction of motion constraint is applied. (2b) shows the case when direction of motion constraint is applied on a straight path. (2c) shows the case when direction of motion constraint is applied on a path with turnings. Black arrow: velocity vector. Grey ellipse: velocity uncertainty. White square: robot. Dashed line: path. [8]

For estimating ego-motion using EKF the state is defined by a 6 element vector containing the three velocity components in body frame and the accelerometer biases: $\mathbf{x}$ = $(v_x^B, v_y^B, v_z^B, b_x^B, b_y^B, b_z^B)^T$. The estimation of the biases allows for the compensation of calibration errors or temperature changes, as well as orientation and estimation errors which affect the velocity integration indirectly. The Kalman filter state and the prediction equations are based on [8].

In the context of the algorithm presented in section III-B, it is not necessary to implement such a direction of motion constraint given by [8], as the algorithm determines an estimate of the direction of motion directly. However, the concept of using the unit vector to compare the results while disregarding the magnitude may be applied. As such, the proposed measurement used to update the EKF is:

$$\mathbf{z_k} = \hat{\mathbf{m}}^B \tag{1}$$

with the non-linear measurement model relating the expected state after the prediction step $\tilde{\mathbf{x}}_\mathbf{k}$ to the expected measurement of

$$h[\tilde{\mathbf{x}}_\mathbf{k}] = \hat{\mathbf{v}}^B \tag{2}$$

where $\hat{\mathbf{m}}^B$ represents the 3D Cartesian direction of motion in the body frame determined from the algorithm, and $\hat{\mathbf{v}}^B$ is the unit velocity vector in the body frame estimated by the first three elements of $\mathbf{x}$. This measurement model has a corresponding observation matrix found from the Jacobian $\mathbf{H_k} = \frac{\partial \mathbf{h}[\mathbf{x}]}{\partial \mathbf{x}}$

The Kalman filter update is given then by the following equations:

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{3}$$

$$\mathbf{x}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - h[\tilde{\mathbf{x}}_\mathbf{k}]) \tag{4}$$

$$\mathbf{P}_k = (I - \mathbf{K}_k \mathbf{H}_k)\tilde{\mathbf{P}}_k \tag{5}$$

where $\mathbf{z}_k$ is the appropriate measurement, $\mathbf{R}_k$ is the measurement noise, $h[\tilde{\mathbf{x}}_\mathbf{k}]$ and its Jacobian $\mathbf{H}_k$ correspond to the non-linear measurement model

IV. RESULTS

A. Optic flow extraction

This section describes the outcome of several tests performed in order to analyze the response of the I2A implementation for optic flow extraction. To do so rotational movements were applied to the sensor in different orientations so that the linear response to changes in angular speed could be measured in both of the optic flow components.

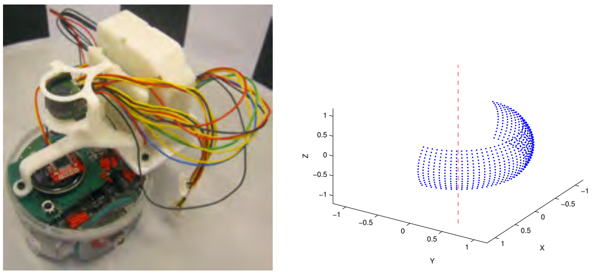
One of the main reasons for choosing this method was the performance of the assembly implementation available. Some tests were carried out in order to know the maximum frame rate possible using this implementation.

For doing so a known amount of regions of interest and a known amount of frames to be computed was configured to the Curvace sensor. Table I shows the data measured and the frame rate computed with that information. It can be observed that this method is able to compute around 20000 optic flow vectors per second.

N ROIs	N frames	Time (s)	Optic flow vec./sec.
180	6000	50.9	21218.075
160	6000	46.2	20779.22
140	6000	40.2	20895.52
120	6000	35	20571.43

TABLE I: Frame rate measured of the optic flow extraction method.

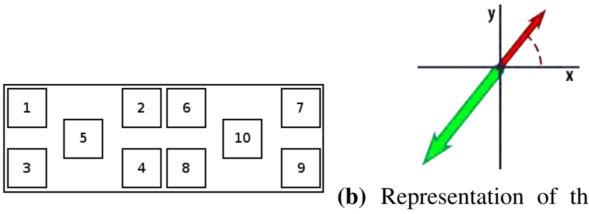
For the following experiment the sensor was placed horizontally on an e-puck (see figure 3a) which would then spin meaning that the image plane was perpendicular to the axis of rotation (see figure 3b). The e-puck was placed the n inside a cylindrical tube which had painted vertical black and white stripes. The optic flow was extracted from 10 different regions distributed along the image plane, see figure 4a. This experiment should provide a defined regression line from all regions of interest. Given that the optic flow data has a linear relation with the angular speed. Figure 4b shows this relationship on a 2D window of the image plane.



(a) Experimental setup used for characterizing the optic flow with the Curvace mounted on an e-puck.

(b) Image plane orientation with respect to the axis of rotation. Blue dots: Image pixels. Red dashed line: axis of rotation

Fig. 3: Experiment setup



(a) ROI distribution.

(b) Representation of the optic flow given by one ROI.

Fig. 4: Region of interest distribution in the image plane for the optic flow calculation (Left). Detail of the optic flow computed in a ROI given a certain angular speed (Right). Red arrow: optic flow vector. Red arch: optic flow vector angle with respect to the horizontal axis. Green arrow: angular speed.

A linear regression between the optic flow vectors magnitude and the angular speed was computed using the x component of the optic flow vector and the angular speeds, see figure 5. The x component is used as it should be the only component varying with horizontal rotation. Table II shows the goodness of fit of the regression values obtained for each region of interest. All values obtained are over 90%, which shows a very high proportion of the variance in the data is explained by a linear model. However, it is also important to know the distribution of the values given a certain speed, which may provide additional insight into the sensor's performance.

ROI id	1	2	3	4	5
R^2 value	0.9207	0.9472	0.9088	0.9538	0.9298
ROI id	6	7	8	9	10
R^2 value	0.9533	0.9519	0.9460	0.9299	0.9343

TABLE II: R^2 values for linear regression of optic flow x component against angular speed, for each of the region of interest, using the stripes pattern.

For showing the variability of the data and how much overlap exists between different optic flow vectors at different speeds, a box plot for each region and speed was computed. For simplifying the process of comparing the data for each region, the standard deviation was computed for each speed, giving one standard deviation value for each speed and each region. From these standard deviation values, the mean of the standard deviation for each region at different speeds was computed. The region with the lowest average standard deviations gave a value of 0.069277. The region with the highest mean of standard deviations got a value of 0.094817. Figure 5 shows the boxplot of the region with the highest mean of standard deviations.

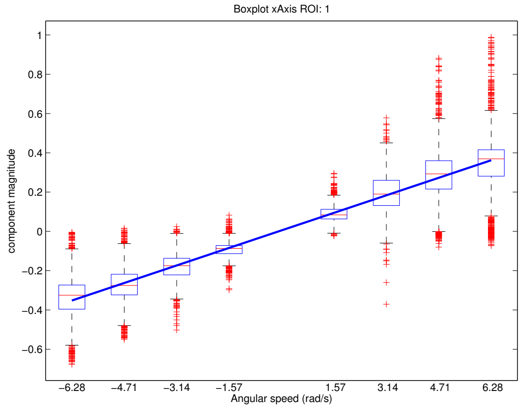


Fig. 5: Optic flow x component (horizontal sensor orientation) regression with respect angular speed of Region 1

Using this setup the optic flow vectors measured should have an angle of 0 or 180 degrees depending on the side of the image plane. For analyzing if that behavior was produced, the atan2 of the optic flow measured vector components was computed. The function atan2 provides the angle of a given vector within the range of $[-\pi, \pi]$. In order to be able to check the variability, abs() was applied to the results since $-\pi$ and π would be very similar, however those results would be considered very different by the statistical measurements without applying abs().

In order to have an idea of the variability of the angle, the same procedure of getting the mean of the standard deviation as before was applied. The region with the lowest average standard deviation across all angular speeds gave a value of $19.0359498°$. Figure 6 shows the region with the highest average standard deviation, with a value of $24.5025401°$. Again in this case the difference on the variability between minimum and maximum standard deviation values is quite low, around $5.4°$.

The results presented show a high linear relationship between the angular speed and the optic flow data, since all the R^2 values are over 0.9. The outcome of the algorithm ensures a reliable optic flow implementation that can be used

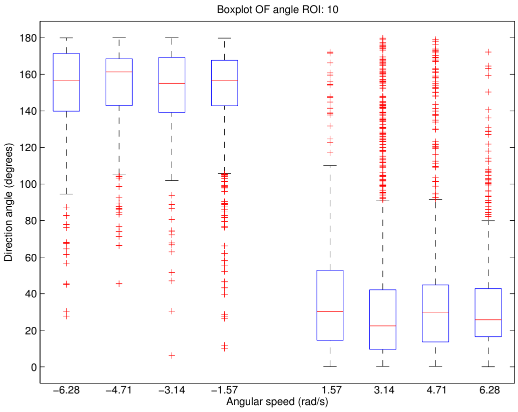


Fig. 6: Optic flow angle variability with respect angular speed of Region 10

as a basis for the following steps of this work in terms of efficiency and accuracy.

B. *Direction of motion estimation*

This section analyzes the results of the direction of motion estimation algorithm. The algorithm follows a coarse to fine voting strategy using the optic flow data from the Curvace sensor.

For evaluating the accuracy of the voting algorithm, several tests were run with the simulator since the output can be compared with ground truth information. Using the optic flow data acquired for characterizing the optic flow extraction method it was possible to get a noise model. For adding noise to the optic flow the following procedure was used. Firstly a noise with the same variability as the optic flow variability when not moving was added. After that a noise with higher variability was added to the optic flow vector proportionally to the magnitude of the optic flow. That means that if one component represented 70% of the magnitude then the same percentage of error would be added to it.

Since the optic flow needs to be de-rotated using the gyroscopes, noise also needs to be added on those sensors. In order to do so, gyroscope values were measured without moving the device which allow for modeling the bias and the variability as well as the distribution of this noise.

In order to know the robustness of the method with several amounts of noise, 3 types of noise (A,B,C) were tested. A is the realistic noise approximation with a standard deviation of 0.05, B is a pessimistic approximation with a deviation of 1 and C has more variability than B with a deviation of 1.5 in order to check the robustness of the method.

Table III shows the mean and the standard deviation error compared with ground truth of the three datasets at the three noise levels.

The results presented in this section show a reliable method for computing an optic flow based direction of motion estimate which is robust to noise. Further experiments using the output of this algorithm for ego-motion estimation need to be carried out, in order to assure the accuracy given by the algorithm is high enough for a reliable ego-motion estimation.

Noise	Datasets	Error mean		Error std	
		θ	ϕ	θ	ϕ
A	tx90XY	-1.539	1.040	3.876	3.766
	tx90YZ	2.057	-4.668	4.429	2.798
	tx90XYZ	-0.971	-2.542	3.994	3.609
B	tx90XY	-2.103	1.576	3.922	4.041
	tx90YZ	-0.333	-2.803	5.873	3.915
	tx90XYZ	-1.966	-1.437	5.097	4.297
C	tx90XY	-1.851	1.624	4.020	3.772
	tx90YZ	-0.847	-1.649	6.082	13.242
	tx90XYZ	-2.462	-1.117	4.926	3.897

TABLE III: Mean and standard deviation of the error when estimating direction of motion, applying noises A,B and C.

C. *Ego-motion estimation*

This section shows the results of the ego-motion estimation algorithm which uses an EKF based approach in order to fuse inertial information with the direction of motion estimate obtained from the optic flow.

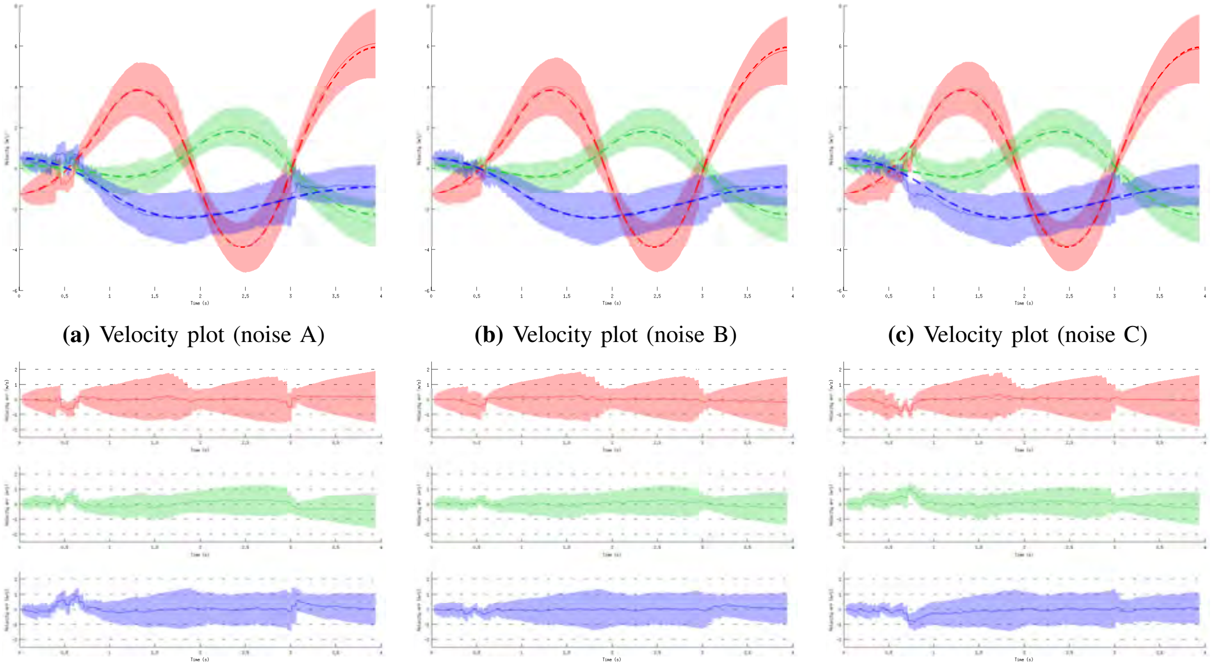
For analyzing the sensor fusion data, a simulated data set containing inertial and optic flow information of the sensor performing different movements was recorded. As the simulator provides ground truth for the inertial sensors as well as the optic flow, noise needed to be added to the data. For the gyroscopes, the same distribution used for the previous stage was used. In order to model the noise of the accelerometers, real data from accelerometers was recorded having those steady, which then allowed to model the noise for the three axes. For the optic flow data, the same procedure described in the previous section was followed and the three noise types where used for the two datasets. The direction of motion estimation algorithm was run at the same specifications used for generating the results of the previous section.

Figure 7 shows the output of the algorithm, as well as the error against the ground truth. It can be observed that the components that present less movement are the ones with lowest uncertainty. This is expected, as explained in section III-C, since the update cannot reduce the uncertainty along the movement direction as the magnitude of the motion is not known.

After successfully testing the algorithm with simulated data the next step would be to test it on a flying platform. This test will assure that the uncertainty given is not too high for being used as a closed loop controller input and have a fully validated method.

V. Conclusions

In order to assist flying platforms to navigate autonomously in GPS denied environments, ego-motion estimation algorithms have been proven a reliable solution [8] [7] [6]. Presented is an alternative which lowers the amount



(a) Velocity plot (noise A) **(b)** Velocity plot (noise B) **(c)** Velocity plot (noise C)

(d) Velocity error plot (noise A) **(e)** Velocity error plot (noise B) **(f)** Velocity error plot (noise C)

Fig. 7: Velocity and velocity error plot of the sensor fusion algorithm when applying three different levels of noise to the dataset taccx90XYZTurning. RGB solid lines: XYZ components of the velocity estimate, respectively. RGB dashed lines: XYZ components of the velocity ground truth, respectively. RGB semitransparent area: uncertainty of the XYZ components, respectively.

of sensor payload needed for ego-motion estimation by using the Curvace sensor.

One of the most remarkable features of the optic flow extraction method implemented in the Curvace is the efficiency. The algorithm allows to compute up to 20000 optic flow vectors per second.

On the direction of motion estimate method chosen, one of its main advantages is its versatility. The method allows to have a trade-off between computational time and accuracy achieved. This characteristic allows to adapt to the needs of the flying platform in terms of precision.

The approach presented on this work shows a very lightweight solution to this problem as the payload added for the sensors needed is of 2 grams. As the payload is very low, the method could be used in aerial vehicles as small as the LadyBird Quadcopter. This quadcopter has a total weight of 25 grams appropriately and a width of 12.5 centimeters. Such devices, coupled with the system presented in this work, show the potential for autonomous navigation in cluttered, GPS-denied environments.

REFERENCES

[1] G.-J. Kruijff, V. Tretyakov, T. Linder, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, E. Pianese, S. Corrao, *et al.*, "Rescue robots at earthquake-hit mirandola, italy: a field report," in *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pp. 1–8, IEEE, 2012.

[2] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.

[3] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[4] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Fraundorfer, E. Kosmatopoulos, A. Martinelli, M. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, *et al.*, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments," *IEEE Robotics & Automation Magazine*, pp. 1–10, 2013.

[5] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II: Matching, robustness, optimization, and applications," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 2, pp. 78–90, 2012.

[6] J. Lim and N. Barnes, "Estimation of the epipole using optical flow at antipodal points," *Computer Vision and Image Understanding*, vol. 114, no. 2, pp. 245–253, 2010.

[7] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, vol. 57, no. 6, pp. 591–602, 2009.

[8] A. Briod, J.-C. Zufferey, and D. Floreano, "Optic-Flow Based Control of a 46g Quadrotor," in *Workshop on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments, IROS 2013*, no. EPFL-CONF-189879, 2013.

[9] D. Floreano, R. Pericet-Camara, S. Viollet, F. Ruffier, A. Brückner, R. Leitel, W. Buss, M. Menouni, F. Expert, R. Juston, *et al.*, "Miniature curved artificial compound eyes," *Proceedings of the National Academy of Sciences*, vol. 110, no. 23, pp. 9267–9272, 2013.

[10] M. Srinivasan, "An image-interpolation technique for the computation of optic flow and egomotion," *Biological Cybernetics*, vol. 71, no. 5, pp. 401–415, 1994.

[11] A. Steiner, "Hardware implementation of optical flow algorithm, UZH-ETH Zurich." `http://n.ethz.ch/~andstein/optical\_flow\_DSP\_steiner10.pdf`, 2012.

[12] J. D. Taylor, ADFA, atthew Garratt, and reenatha Anavatti, "A Practical Algorithm for Optic Flow," 2008.

[13] J. Lim, *Egomotion Estimation with Large Field-of-View Vision*. PhD thesis, 2010.

[14] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 5, pp. 731–747, 2001.

A Method for Dynamically Balancing Kicking Motions for the NAO Robots

Mariela De Lucas Alvarez

Abstract— **Balancing moving robots has always been a challenging task in the field of robotics. Within the Robocup challenge, this is a present concern for soccer playing NAO robots. A method for solving balance pertaining kicking motions is proposed. In order to accomplish this task, the Zero-Moment Point (ZMP) is addressed to generate balancing motion patterns to sustain a kick. This Inverse Dynamics problem is controlled with a ZMP observing Linear Quadratic Regulator.**

I. INTRODUCTION

The NAO humanoid robots, as used within the RoboCup, are expected to perform a wide variety of maneuvers ranging from walking, standing up from a fallen down position, diving to save the ball, etc. Within the Standard Platform League, the soccer competition comprises many tasks that represent a great challenge to accomplish. There is a limitation related to the kicking execution, where there is a need to effectively observe the motions of the robot in order to perform in stable manner. With this in mind, there is a pressing need to design a method for observing balance and stability, taking into account external disturbances to generate fluid and dynamic motions that suit the characteristics of a soccer match.

Along these lines, the proposed solution is to design a balancer that acknowledges the dynamic characteristics of the NAO. The suggested solution follows the groundwork of many other researchers who have explored this problem with the development of linear models to pattern motions of bipedal robots. Such is the Cart-Table model which uses a notion to determine a location for stability as a mixture of a Zero-Moment point and Inverted Pendulum based approach. This method is approached along with a linear quadratic controller that compensates for motion errors. This is integrated in a module within an existing software framework to balance movements considering external disturbances for efficient control needs pertaining kicking tasks. Conclusively, satisfying the dynamic nature of a game play and this way enhancing the balancing capabilities of the NAO robot.

II. BALANCING HUMANOID ROBOTS

There have been many strategies to create balanced motions for bipedal robots. From the point of view of control and walking pattern generation, there can be two ways of approaching these goals. One is by knowing details of the dynamic infrastructure of the robot, such as mass, location of the center of mass and inertia of the links to construct the walking patterns, relying on the accuracy of the models. Another one will use a contrary approach where it will work using limited knowledge of the dynamics of the system, relying on a feedback control.

The first can be called zero-moment-point-based approaches and the second ones inverted-pendulum-based approaches, since they frequently model the zero-moment point and an inverted pendulum respectively. Throughout the chronicles of the RoboCup competition, many teams have conjugated both categories by using zero-moment-point-based linear inverted pendulum models to accomplish efficient balanced walking motions.

Works on biped walking by Kajita *et al* have reviewed both approaches. In [2] they analyzed the dynamics of a linear three-dimensional inverted pendulum in which motion is constrained to move along an arbitrarily defined plane to synthesize a model for biped walking generations. Also, in [1] they introduced a new method for biped walking pattern generation by using a preview control of the zero-moment point. Both approaches laid the groundwork for the implementation of walking and kicking engines of RoboCup team B-Human from University of Bremen and the German Research Center for Artificial Intelligence.

This last work lays the foundation of this solution, [5]. It covers two techniques of dynamically balancing the NAO robot by estimating the motions of the robot and using them to determine its zero-moment point. Consequently, offering a solution to the inverse dynamics problem.

A. Zero Moment Point

The concept of Zero Moment Point (ZMP) was first introduced by Vukobratovic and Borovac [3] as a method to localizing the point within the supporting polygon at which the reaction forces of will keep a robot balanced. The acceptance of this concept is based on its importance for biped gait analysis, synthesis and control. Consequently, the significance of the Zero Moment Point is built on more than three decades of diverse applications related to various humanoid locomotion devices.

The most important task of a locomotion system is to maintain the balance or stability, for instance, while performing a gait . This is achieved by maintaining the whole support polygon area in contact with the ground. The foot relies freely on the support and the only contact with the

This work was supported by the Erasmus Mundus Grant from the European Commission

M. De Lucas Alvarez was with Institute of Action, Perception and Behaviour from University of Edinburgh (UK), Heriot-Watt University (UK), Universitat de Girona (Spain) and Université de Bourgogne (France) as an MSc student of Erasmus Mundus Master in Compter Vision and Robotics (ViBot). mariela.delucas@outlook.com

environment is accomplished via the friction force of the vertical force of the ground reaction. As previously explained, the zero-moment point lies within the supporting foot. If it is not located in the foot, the fictional ZMP, is outside the foot sole and will cause the robot to tilt over and fall eventually.

When existent, the zero-moment point provides the location at which the reaction force of that ground exerts on the foot must act, such that all torques are canceled out. The ZMP, p_0, then satisfies the equality,

$$zmp \times f_{r,z} + \tau_{xy} = 0 \tag{1}$$

and hence,

$$zmp = \frac{n \times (-\tau_{xy})}{n \cdot f_{r,z}} = \frac{1}{f_{r,z}} \begin{bmatrix} \tau_y \\ -\tau_x \\ 0 \end{bmatrix} \tag{2}$$

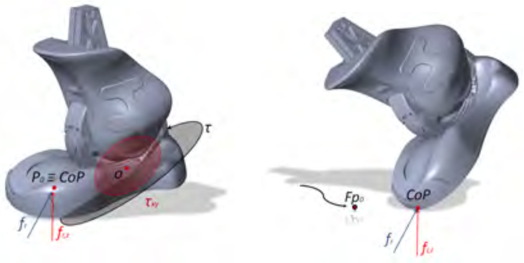
with $f_{r,z}$ being the vertical component of the reaction force, τ_{xy} the total torque of the robot relative to the projection of origin of the support polygon on the ground, n the normal vector, also on the ground and × denoting the cross product of the two vectors.

It is important to explain how the notion of the zero-moment point relates to the ground surface. It is crucial to note that a zero-moment point lying outside the foot polygon makes no sense, since in fact it does not exist. To understand the meaning of this, lets assume that the ZMP is no longer within the support polygon. In view of the fact that this point was obtained from the condition where horizontal and vertical moments are zero, we can consider it as a fictitious zero-moment point (FZMP), Fig.1(b). Therefore, if the ZMP can only exist within the support polygon, we call it a fictitious location the calculated ZMP that lies outside the support polygon .

Another issue that must be clarified is the difference between center of pressure (CoP) and ZMP. The pressure between the foot and the ground can be translated in to a force acting at the center of pressure. If this force balances all active forces acting on the robot during the gait, its acting point will be the ZMP. Hence, when the robot is dynamically balanced the CoP and the ZMP will coincide. If the robot is not balanced the ZMP will not exist and the robot will collapse about the foot edge where the CoP is localized.

From (2), it is inferred that the position of the ZMP depends on the dynamics of the machine, and therefore, determining the proper dynamics of the mechanism above the foot is essential to ensure a desired zero-moment point position. Figure 2.1 shows the how reaction forces act on a dynamically balanced gait and the relations between the center of pressure and the zero-moment point.

The concept of ZMP has had an essential role throughout the years in theoretical approaches and practical development of humanoid robots and biped locomotion, and with the accelerated research to incorporate humanoid robots close to humans, it is highly expected that its notion will not cease to be present.



(a) (b)

Fig. 1. **ZMP and FZMP.** a) In a dynamically balanced gait, the vertical component $f_{r,z}$ of the reaction force f_r acts on the ZMP, p_0, to cancel the horizontal torque, τ_{xy}, of the total, τ, relative to the origin of the foot, o. b) In an unbalanced gait, the ZMP does not exist, and the ground forces act on the CoP. The FZMP, Fp_0, lies outside the foot sole.

III. DYNAMIC MODELS FOR BALANCE ESTIMATION

A. ZMP and 3D Linear Inverted Pendulum Mode

To apply the Cart-Table model, to a balancing robot we must consider controlling an inverted pendulum constrained to a certain defined plane where a mass is moved along in equilibrium. This linear dynamics scheme is called Three-Dimensional Linear Inverted Pendulum mode (3D-LIPM) [1], [2], [11]. The cart represents the motion of the CoM and the table, the supporting foot. The cartesian position of the CoM is defined by,

$$p_{com} = \begin{bmatrix} x \\ y \\ z_c \end{bmatrix} \tag{3}$$

where z_c is de vertical position of the CoM defined at a constant height. This constrained height is represented with normal vector $(k_x, k_y, -1)$ and z intersection, z_c as,

$$z = k_x x + k_y y + z_c \tag{4}$$

If we consider the plane to be horizontal $(k_x = k_y = 0)$ the dynamics under the constraint control is given by,

$$\begin{aligned} \ddot{x} &= \frac{g}{z_c} x + \frac{1}{m z_c} \tau_y, \\ \ddot{y} &= \frac{g}{z_c} y - \frac{1}{m z_c} \tau_x \end{aligned} \tag{5}$$

where m is the CoM, g is gravitational acceleration and τ_x, τ_y are the torques around the x-axis and y-axis respectively. For the 3D-LIPM under a constrained plane, the ZMP can be calculated,

$$\begin{aligned} p_x &= -\frac{\tau_y}{mg} \\ p_y &= \frac{\tau_x}{mg} \end{aligned} \tag{6}$$

where $p_{x,y}$ are the locations of the ZMP on the ground. Consequently, we can substitute from (5) to (6) to have the Cartesian positions of the ZMP,

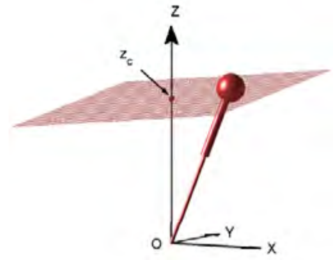


Fig. 2. **Inverted Pendulum.** This diagram shows an inverted pendulum under a constrained plane z_C.[1]

$$
\begin{aligned}
p_x &= x - \frac{z_c}{g}\ddot{x} \\
p_x &= y - \frac{z_c}{g}\ddot{y}
\end{aligned}
\tag{7}
$$

The concept behind the Cart-Table model is that if the cart accelerates, *i.e.* the CoM, at a proper rate, the table, *i.e.* the support leg, can sustain it upright for a moment. At this point the ZMP exists within the constraint plane and the moment around the ZMP is zero and can be verified below using set of equations (7),

$$
\tau_{zmp} = mg(x - p_x) - m\ddot{x}z_c = 0 \tag{8}
$$

IV. ZMP Pattern Generation

A. Cart-Table Model for ZMP tracking

To represent the NAO as the Cart-Table model, the cart motion gives the trajectory of the CoM. The constraint factor is approached by first settling the level of the support leg to the constant plane z_c before initializing the balance. The ZMP is easily calculated then from the equations in (7). This linear model uses the ZMP reference of a certain period and generates the corresponding CoM trajectory. To control the ZMP a variable, u_x, is used as the time derivative of the horizontal acceleration of the CoM,

$$
u_x = \frac{d}{dt}\ddot{x} \tag{9}
$$

Observing this u_x as the input of the ZMP (7), the set of equations can be translated into a dynamical system,

$$
\begin{aligned}
\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \\
p_x &= \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}
\end{aligned}
\tag{10}
$$

The positions, velocities and accelerations for the CoM are Kalman-filtered in a combined state, to obtain values for the x and y components. This dynamic system yields the next CoM point in the trajectory so that the resulting ZMP follows the reference specified.

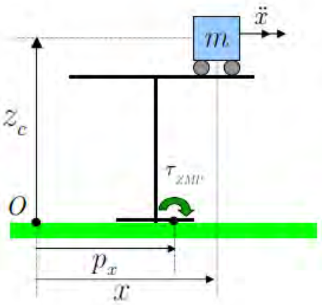


Fig. 3. **Cart-Table Model.**[1]

B. Pattern Generation by Linear-Quadratic Regulation

To control the motions of the NAO, an optimal linear discrete-time finite-horizon quadratic regulator similar to [5] was designed. The system is discretized with sampling time of T as,

$$
\begin{aligned}
x_{k+1} &= Ax_k + B\bar{u}_k, \\
p_k &= Cx_{x+1},
\end{aligned}
\tag{11}
$$

where,

$$
\begin{aligned}
x_k &\equiv \begin{bmatrix} x(kT) & \dot{x}(kT) & \ddot{x}(kT) \end{bmatrix}^T, \\
\bar{u}_k &\equiv u_x(kT), \\
p_k &\equiv p_x(kT), \\
A &\equiv \begin{bmatrix} 1 & T & T^2\frac{1}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \\
B &\equiv \begin{bmatrix} T^3\frac{1}{6} \\ T^2\frac{1}{2} \\ T \end{bmatrix}, \\
C &\equiv \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix}.
\end{aligned}
$$

With this model the controls to be applied at kT are,

$$
u_k = -(R + B^T P_k B)^{-1} B^T P_k A x_k \tag{12}
$$

$$
P_k = Q + A^T(P_{k-1} - P_{k-1}B(R + B^T P_{k-1}B)^{-1}B^T P_{k-1})A
$$

The LQR is essential a method for finding an ideal state-feedback controller. The cost or performance index, J, can be interpreted as an energy function, so that by making it as small as possible keeps small the total energy of the system. It is defined as a quadratic cost function,

$$
J = \sum_{k=0}^{N} \left(x_k^T Q x_k + u_k^T R u_k \right). \tag{13}
$$

The energy function weighs both the state x_k and control input u_t, so that if J is small, then neither x_k nor u_k can be very large. The two matrices Q and R were selected based on the terms that to keep J small the state must be smaller is opting for a large Q. For a large control matrix R, then u_k should be smaller than J.

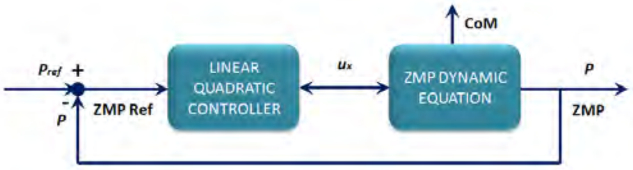


Fig. 4. **CoM pattern generation by ZMP control.**

From the dynamics of (11) and the controller (12) a CoM observing pattern generator as a ZMP tracking control system can be designed as in Fig. 4. The commands of the dynamic update yield new positions for the CoM which can be easily approached by following an inverse kinematic solution to calculate the joint angles for the support leg.

V. Pose Transformations for Inverse Kinematics

It was necessary to calculate several pose transformations in order to construct the linear model described above. The software framework that handles the motions of the NAO integrates its own coordinate system. To observe specific body locations of the NAO lets take into account the next rulings on the CoM and foot limbs with regards to the torso,

$$^{T}\mathbf{T}_{C}\ ,\ ^{T}\mathbf{T}_{F} \tag{14}$$

The transformations above do not have to be calculated since they are known in the system and are given as homogeneous matrices. The superscript, T, refers to the torso of the NAO, F refers to the foot, and C to the CoM. These are obtained through the a torso matrix provided within the software framework, which provides a transformation from the ground to the origin of the NAO.

To observe the motions of the CoM, we must relate it to the ground projection of the support foot. This is given by the following transformation sequence,

$$^{G}\mathbf{T}_{C} = \left(^{T}\mathbf{T}_{F}\right)^{-1} \cdot \ ^{T}\mathbf{T}_{C} \cdot \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{15}$$

where, G, refers to the ground, $\mathbf{I}$ corresponds to an identity matrix, and $\mathbf{t}$ refers to a translation vector $\mathbf{t} = [0, 0, -z_{fo}]$ constructed with the robot model defined in the software framework. Lets recall that, according to the coordinate system of the NAO, the foot origin, fo, is located in the ankle of the robot. Hence, it needs to be translated to the ground. After obtaining the dynamic update, the new CoM position needs to be in reference to the torso of the NAO given that the inverse kinematics solution only takes positions for end effectors with that reference,

$$^{F}\mathbf{T}_{C} =^{G} \mathbf{T}_{C} \cdot \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{16}$$

where $\mathbf{t} = [0, 0, z_{fo}]$,

$$^{T}\mathbf{T}_{F} =^{T} \mathbf{T}_{C} \cdot \ \left(^{F}\mathbf{T}_{C}\right)^{-1} \tag{17}$$

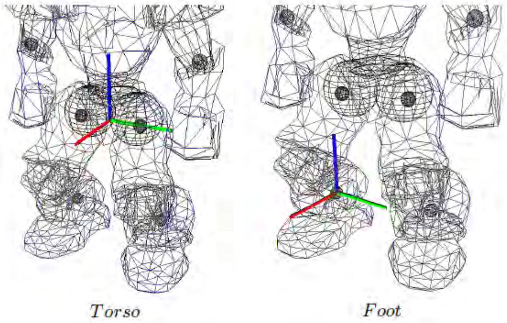


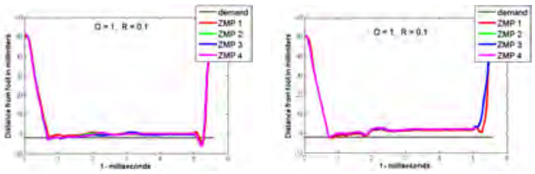
Fig. 5. **Coordinate sytem of the NAO.** Visualization of the coordinate system references as used in the inverse kinematic solver. Red is for x-axis, green is for y-axis and blue is for z-axis.[4].

VI. Results

To evaluate the performance of the controller a series of preprogrammed kicking sequences were run to observe the behavior of the ZMP with regards to the demanded ZMP. The experiments consisted in executing the following tests,

- A sequence where the NAO lifts the kicking leg and sustains it to balance in the support leg
- A sequence where the NAO executes a straight kick. This was tried out using different Q and R parameters
- A series of sequences where the NAO executes a 30°, 45°, 60°, and a 70° kick.

A kick sequence consist of three stages. First the NAO assumes the kicking stance, where it leans its body towards the support leg in order to reach the desired constrained height as established by the 3DLIPM. Second, the kicking leg starts its motions while the support leg balances. Third, the kicking sequence finalizes and the NAO assumes the stand pose.



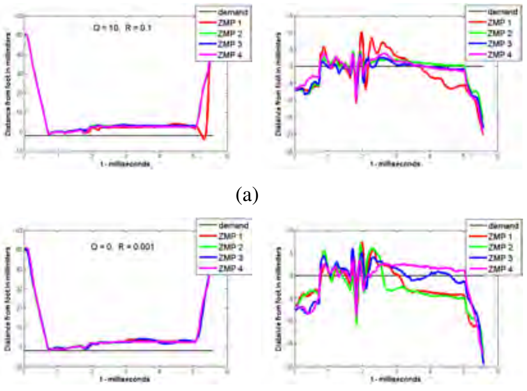
(a)

(b)

Fig. 6. a) Balance with no kick. b) Balance with straight kick.

All the plots show 4 consecutive motion executions. Results above are to compare the disturbances generated with a kick motion and without, where the ZMP references are tracked. The pattern shows how the NAO leans, sustains balance or kicks, and assumes the standing position again. To asses how the model fits, the dark line shows the desired ZMP location.

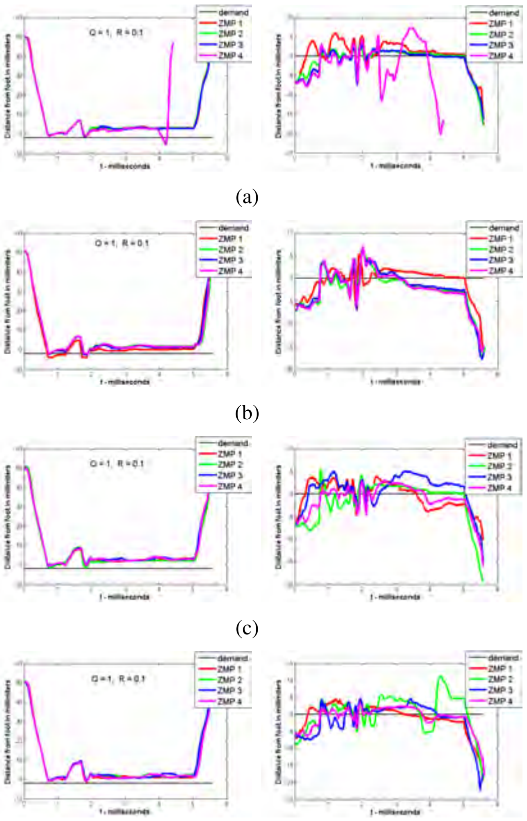
The plots below show the tests for straight kicks with different LQR parameters, show for both y-axis and x-axis. It was observed that most of the overshoots are present in the saggital motion during a straight kick. This makes sense given that the NAO strives to sustain its balanced posed while it swings the kicking leg backwards and forwards. The lateral motions in general appear to be smoother given that there are virtually no disturbances for this motion. Some small overshoots are observed, however, when the NAO re-assumes the standing position



(a)

(b)

Fig. 7. **Straight kick testing different parameters.** a) First test kick in Y-axis and X-axis, and b) second test kick in Y-axis and X-axis.



(a)

(b)

(c)

(d)

Fig. 8. **Angular kick executions.** a) 30°, b) 45°, c) 60°, and d) 75°.

A somewhat different scenario is observed for the ZMP patterns in the angular kick test above. The saggital motions still overshot, but appear more compact and closer to the demanded ZMP for the motions with a wider kick [8(c),8(d)]. In spite of these overshoots, it is important to note that their notation is in *mm* and some do not represent a significant disturbance for the balance of the NAO. The lateral motions are shown to be more affected but still managed in an acceptable manner by the controller as the NAO widens the kick motion away from its support leg to hit a ball in a certain angle. Regardless of the type of kick, the ZMP shows considerable disturbance while the kick foot is moved.

VII. CONCLUSIONS AND FUTURE WORKS

A. *Conclusions*

In the previous sections, a method was presented for tracking the ZMP trajectory of a kicking NAO in order to regulate the desired position to sustain a balanced pose. The Cart-Table model was presented as a reliable method for representing the ZMP. After reviewing other methods that incorporated this notion to generate motion patterns, the solution was given as a ZMP observing Linear-Quadratic Regulator. It was reinforced that, as in any optimization problem, the choice of parameters play a key role in the robustness of a controller.

To show the efficiency of the solution, the balancer was tested with a series of preprogrammed kicks. In here it was observed that each kick affected differently the response behavior of the regulator from which was useful to observe its robustness. The regulating response from the controller efficiently corrected the pose of the NAO for the straight kicks, while the angular kicks manifested larger disturbances difficult to control.

B. *Discussion and Future Works*

The results obtained in from the proposed approach were sufficient given that the cost parameters were chosen without a design procedure and selected through recommended guidelines that were found throughout the research. It makes sense that by choosing larger values for Q and smaller for R the state control efforts sufficiently kept the calculated ZMP close to the demanded ZMP. Theoretically, larger values of Q result in poles of the system matrix A, from the Riccati equation (12), being further left in the s-plane so that the state reaches decay faster. On the contrary, a large R means that less control effort is used, so the poles are slower and the state values of x become larger.

In comparison to the results shown in [5] where a preview controller is used for the Cart-Table model, this solution appears to have an acceptable response and smaller overshoots across time. Thus, it is safe to say that the Cart-Table model used to generate patterns for the ZMP can improve the robustness of the system using different controllers.

This approach includes the use of estimating the cost of the controller using the Algebraic Riccati Equation to solve for the state-variable feedback to determine the cost before the control is applied to the system. Although generally, obtaining a guaranteed solution to stabilize a system will seldom offer meaningful understanding of the robustness of the system. In addition, the improvement to angular kick responses would enable the testing on an online-generator for kicking motions for the enhanced adaptability of the NAO robot in the soccer pitch.

References

[1] S. Kajita and F. Kanehiro and K. Kaneko and K. Fujiwara and K. Harada and K. Yokoi and H. Hirukawa, Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point", *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2003.

[2] S. Kajita and F. Kanehiro and K. Kaneko and K. Yokoi and H. Hirukawa, The 3D Linear Inverted Pendulum Mode: A Simple Modelling for Biped Walking Pattern Generation, *Proceedings of the 2001 IEEE International Conference on Intelligent Robots and Systems*, 2001.

[3] Miomir Vukobratovic and Branislav Borovac, Zero-Moment Point - Thirty Five Years of its Life, *International Journal of Humanoid Robotics*, 2004.

[4] B-Human, Team Report and Code Release 2011, 2011.

[5] Felix Wenk and Thomas Roefer, Online Generated Kick Motions for the NAO Balanced Using Inverse Dynamics, *In RoboCup 2013: Robot Soccer World Cup XVII, Lecture Notes in Artificial Intelligence*, 2013.

[6] Pierre R. Belanger, Estimation of Angular Velocity and Acceleration from Shaft Encoder Mesurements, *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 1992.

[7] Anthony C. Fang and Nancy S. Pollard, Efficient Synthesis of Physically Valid Human Motion, *Proceedings of the ACM SIGGRAPH 2003 Conference*, 2003.

[8] F.L.Lewis, Linear Quadratic Regulator (LQR) State Feedbak Design, *University of Texas Arlington Lectures*, 1998.

[9] Tohru Katayama and Takahira Ohki and Toshio Inoue and Tomoyuki Kato, Design of an optimal controller for a discrete-time sytem subject to previewable demand, *International Journal of Control*, 1985.

[10] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, *Transactions of the ASME - Journal of Basic Engineering*, 1960.

[11] Wael Suleiman and Fumio Kanehiro and Kanako Miura and Eiichi Yoshida, Improving ZMP-Based Control Model Using System Identification Techniques, *9th IEEE-RAS International Conference on Humanoid Robots, 2009.*

Visual activity recognition for the humanoid robot REEM

Mariia Dmitrieva mari.dmitrieva@gmail.com
Master in VIsion & roBOTics
PAL Robotics, C/ Pujades 77-79, 4^o4^a 08030 Barcelona, Spain

***Abstract*— Vision-based human activity recognition is a state-of-the-art task, though it has been developed already for more than twenty years. The task is on high demand in monitoring systems, human-computer interaction and robotics.**

This article describes an action recognition module which was developed as a one camera based solution. An action is presented by a space-time volume and described by a movement history in the neighbouring frames. The history is accumulated in a static vector-image, called Motion History Image. The pixel values of the image represent the motion properties.

Thereupon the action representation image is specified by a shape descriptor. Afterwards a machine learning technique, in particular the Support Vector Machine, is used to classify actions. The accuracy of the approach is improved by an implementation of a buffer to store the intermediate results for the final decision.

The solution for the visual activity recognition was implemented for the REEM robot with the possibility to be extended to other robots.

I. Introduction

Nowadays robotics is a fast developing industry. Even so the most of the robots are still used for research and study, there are robots in our daily life, like vacuum cleaner robots, robot-pets and a big field of robotics aims at the industry sector. The robot-human interaction, with the perception and analysis of people behaviour is essential for the robots to leave the laboratory accommodations and work in a realistic environment. Visual activity recognition in robotics aims to help in communication.

The activity recognition application is developed for the wheeled robot REEM by PAL Robotics. The robot has a wide range of sensors: microphones, stereo-camera, lasers, ultrasounds, accelerometers and gyroscopes [1]. REEM is a ROS based robot, the Hydro Medusa version of ROS was used in this work.

Fig. 1: The human-robot interaction with the REEM robot

This work was supported by PAL Robotics

A. Related work

Activity recognition can be considered as a process of frame labelling according to an action in the video sequence [2]. The process contains two main phases: image representation and classification. All the approaches, designed by representation and classification can be categorized into two main directions: *single-layered approaches* and *hierarchical approaches*. The single-layered approaches recognize actions based on a frame sequence. While hierarchical approaches recognize human activity by a set of simple events.

1) Single-layered approaches: any activity is represented as a set of images, which can be categorized into classes. The methods are divided into two groups via human activity interpretation.

First group is the *Space-time approaches* which consider a video as a 3D volume (x, y and time dimensions). An action can be described by a space-time trajectory here, as it is made in the work of Sheikh et al. [3]. Space-time volume is another way to describe an action. In the Temporal Template method, presented by Bobick and Davis [4], the volume is described by the neighbour frames difference. Also Space-Time Features can be extracted from the XYT volume to describe and recognize an action. Good examples are a HOG3D descriptor of Alexander Klaser [5], or Ivan Laptev's HOGHOF descriptor [6].

Another group of the single layered approaches is *sequential approaches*. They consider the action representation as a sequence of observations (feature vectors). The exemplar-based recognition approaches of the group compare a sequence of features vectors with a template sequence. For instance, Yacoob and Black [7] consider the input video as a set of signals and describe only the changes of feature values. On other hand, the activity can be described as a trained model, which corresponds to sets of the feature vectors. Yamato et al. [8] was the first who adopted HHMs from speech recognition to action recognition.

2) Hierarchical approaches: are suitable for recognition of a complex high-level activity. These approaches can be divided into three main groups.

Statistic approaches use statistical state-based models for the activity representation. Multiple layers of state-based models, such as HMMs and DBNs, are used in methods to recognize activities. The good example of the HMMs using is the approach presented by Oliver et al. [9].

Syntactic approaches use a grammar syntax to describe an activity. Stochastic context-free grammars (SCFGs) are widely used in the class of the approaches. For instance, a method based on SCFGs was proposed by Bobick and Ivanov [10].

Description-based approaches describe the high-level activity by sub-events. For example, the approach presented by Walterio and Sundaram [11] models human activity by processing of a global motion in the frame.

All the approaches can be processed offline and online, depending on computational complexity and working platform. Commonly, offline solutions give more accurate results, but most of the applications, like a robot-human interaction, security monitoring and other systems which require direct reaction, demand online solutions. The huge challenge is to achieve high accuracy with a real-time solution.

B. Outline of the paper

The goal of the work is to develop an activity recognition application, which will identify a set of actions in real time. Hand waving is the most common action, that the REEM robot faces in events. Therefore the results of the developed approach are presented for the hand waving activity in the first place. Nevertheless the chosen strategy is able to scale up to another actions. The solution must be applicable to other robots, including robots with a single camera. Therefore, the algorithm has to work with a monocular camera only. The most important requirement for the developed solution is the robustness and real-time processing.

The overview of the developed solution is presented in Section II. The approach is described in details in Sections III, V and IV where the Regions of Interest location, the Action Recognition Module, the Final Decision Module are described. The experiments are presented in Section VI. Section VII contains conclusions and future work.

II. Algorithm outline

The developed recognition module describes the space-time volume by a Motion History Image to recognize an action. The approach is based on Temporal Template method, presented by Bobick et al. [4].

The outline of the implemented strategy is presented on the figure 2. Before the recognition process, to reduce the amount of processing data and remove most of the background, the ROI is placed for each frame in a sequence. A sequence of ROIs is accumulated in a buffer for processing by the action recognition module.

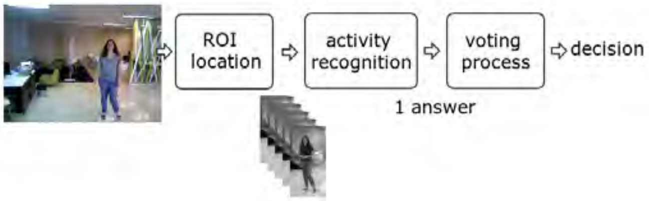


Fig. 2: Scheme of the activity recognition approach

The activity recognition module contains an action representation block and an action classifier. One output of the classifier is a decision made according to one ROI sequence. This result is not the final decision about the action. To increase the robustness of the approach, the classifier outputs are stored in a buffer, called the voting vector. When the vector reaches a given size, the voting process takes place to have the final result.

III. Region of Interest module

The region of interest (ROI) is detected as a region of frame where the action is more probable, in other words, the area where people are. Therefore, people in the frame are the main target to build a ROI around and the location of a ROI is defined by a people detector.

The placed ROI is stored in a buffer, because the space-time approach needs a sequence of frames to recognize an action. When the amount of people in the frame is more than one and people move, the ROIs need to be tracked. The number of buffers is equal to the number of tracked ROIs.

Moreover the size of ROIs changes from frame to frame. Robustness of the action recognition depends on the stability of the tracked ROI. The ROI is expected to have a constant size, as long as it is possible. Meanwhile the position can be changed in case of people movement.

A. ROI location

The location of a ROI is defined by a people detector. They can be detected by a full-body detector or by a face detector, depending on the distance between a robot and a human.

1) Full-body ROI: the full-body detector based on Histogram of Oriented Gradients (HOG) [12] integrated in REEM is used. The HOG descriptor evaluates an image and describes the shape of it by the distribution of local intensity gradients. The people detector allows to recognize a human when the full-body is seen on the frame. The detector runs at 4 Hz in REEM.

The width of the ROI provided by the person detector is enlarged by 20%. This extension gives a space for arm movements (Figure 3a). The full-body detector is applicable only when the body is seen on the frame completely. For the cases of a close person location, ROI is built based on a face detector.

2) ROI based on face detection: the face location is the main criteria for the ROI position. The face detector is implemented in a separate node. It uses Haar cascade classifier for the detection [13].

The size of the calculated ROI is proportional to the face. The region has 6 times the width and 4 times the height of the face region. Figure 3b illustrates the ROI.

B. ROI tracking

Position of the moving person is changing from frame to frame and so the ROIs. To observe an action the ROI is tracked by its location. A buffer is a storage of tracking ROIs. Each track has its own slot. According to the coordinates of the new rectangles center the tracker assigns the ROI to a slot or creates a new one. The tracker has a threshold for a position choice. It is a value of an Euclidean distance

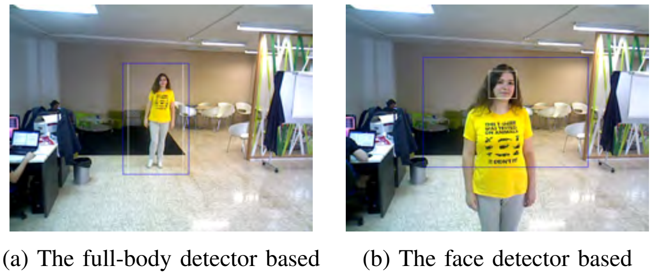

(a) The full-body detector based (b) The face detector based

Fig. 3: ROI size and location

between centres of the current and the buffered ROIs. The value is sensitive to the frame rate, distance to the person and speed of the people movement.

The tracking of ROIs are not saved directly to the buffer, but need to be stabilized first.

C. ROI stabilisation

The tracked ROI's size and position depend on the face or full-body detectors, so the accuracy of the detection impacts the ROI position. The ROI location is stabilised to compensate the frame to frame variation in the detected ROI size. ROI stabilisation is implemented using the data from the REEM's laser sensor.

1) Leg detector: it processes the positions of all the reached object edges to find possible leg positions. The proper detection of the leg positions is a sophisticated task, because characteristics of the leg patterns can be very contrasting (figure 4).

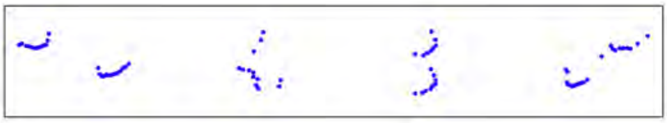


Fig. 4: Clusters of points from laser range scans which correspond to person legs [14]

The detector is based on the work of Arras et al. [14]. The approach uses AdaBoost to train a classifier.

The detected leg positions have to be expressed in image frame coordinates, to place it on the image and process the data with the camera data.

2) Leg position projection: the laser frame is 10 cm above the robot reference frame. The positions can be projected to the image coordinate system by the intrinsic and extrinsic matrices as show in eq. (1).

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} {}^{c}T_L{}^{L} \begin{bmatrix} X \\ X \\ Z \\ 1 \end{bmatrix} \tag{1}$$

Where the ${}^{c}T_L$ is the homogeneous transform from the laser frame to the camera frame and f_x, f_y and C_x, C_y are the camera intrinsic parameters considered.

3) Stabilization technique: the stabilization strategy depends on leg positions. The scheme contains three steps presented on Figure 5.

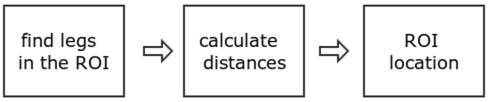


Fig. 5: ROI stabilization scheme

The leg position is searched in the bottom half of the ROI, the legs closest to the ROI vertical centre are selected. It is possible, that the legs are not detected, in this case, the ROI has the size of the last ROI from the buffer and its own position.

If the legs position is found, the second step will be to calculate a difference between the previous position and the new one. Two values are calculated. The difference between robot - person distances of the previous ROI position and the current one, eq. (2). The second value is the distance between the leg locations in 2D image frame. Euclidean distance estimates the value eq. (3).

$$D_1 = |d_1 - d_2| \tag{2}$$

$$D_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}$$

If the differences are less than the thresholds, the position stays the same, otherwise the position is changed. For the tracked ROI its size is fixed with the value in the buffer. The stabilization is needed to achieve activity recognition by limiting the impact of the surrounding area.

The stabilised ROIs are stored in the allocated spot of the buffer to be processed by the action recognition module.

IV. Action Recognition Module

The activity recognition approach is based on the Temporal Template method, where an activity is described by motion images [4].

A. Temporal Templates

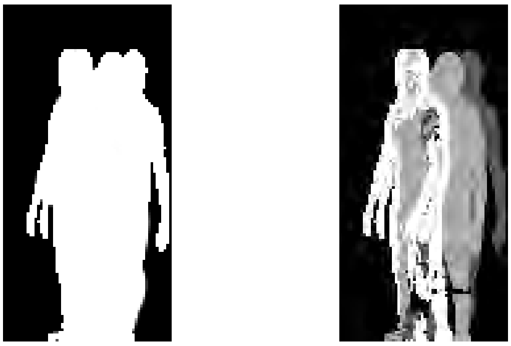
The origin of the method is based on the representation of a movement by a static vector-image. Each pixel is a vector value which represents motion properties. The assumption of the method is that the background is static. Therefore the action can be registered by a subtraction of a frame sequence.

The temporal template, in general, can be represented by two types of images: Motion Energy Image (MEI) and Motion History Image (MHI), see Figure 6.

The MEI is a binary picture. It indicates a region of motion and is defined as follows.

$$E_\tau(x, y, t) = \bigcup_{i=0}^{\tau-1} D(x, y, t - i) \tag{4}$$

Where D(x,y,t) is a sequence of binary images which indicates a region of the motion. In other words, each image



(a) Motion-Energy Image (b) Motion-History Image

Fig. 6: MEI and MHI, buffer size 5 frames, 4 frames/sec

of the sequence is a binary result of the neighbour frames subtraction. The resulted shape of the Motion Energy Images blob guidances to estimate the view condition and the action itself.

The Motion History Image (MHI) is a grayscale image which shows the way an object moves. The intensity of a pixel represents the temporal history of the motion at the pixel position. Pixel H can be presented by

$$H_{\tau}(x,y,t) = \begin{cases} \tau, if D(x,y,t) = 1 \\ max(0, H_{\tau}(x,y,t-1) - 1), otherwise \end{cases} \tag{5}$$

The MHI images provide information about a direction of the motion.

Motion History and Motion Energy Images can define variety of movements, describing its direction, area of movement and even point of view.

The original matching method proposed by the authors is based on the comparison of Hu moments computed for each components [4].

The method doesn't require time consuming calculations and can be run in real-time, so the Motion-History Image is used as a basis in our work. However a different way to describe the motion images is adopted and a machine learning algorithm is used for matching.

B. Solution details

In the implemented approach the Motion History Image is calculated for a sequence of tracked ROIs. A number of frames in the sequence defines a depth of MHI (how many frames will be described by the image). On this stage a difference of the two neighbour frames is calculated and transformed into a binary image, using a thresholding technique. The binary result of the subtraction is used to update the Motion History Image. One important parameter of the update is the observation time (duration time), it is the time interval of the history representation.The observation time is made flexible to the time step value and the size of the frame buffer, it is set by the formula (6).

$$t_{obs} = (Nframes - 2) * timestep \tag{6}$$

The image normalisation for the thresholding and calculation of the observation time allows to estimate the proper Motion History Image. The image is the resulting description of an action.

The MHI is a gray scale image and it can be represented by a descriptor. The descriptor has to robustly capture salient information from the image patch. Histogram of Oriented Gradients is used for the purpose.

To achieve a constant size of the descriptor for all the MHIs, each image is resized to a pattern size. After the resizing the MHIs have size 64x128 pixels for the full-body ROI and 128x128 pixels for the face based ROI. The HOG descriptor size is 3780 and 30870 correspondingly. The HOG descriptors of the Motion History Images is the final representation of an action.

To compare the HOG descriptors and recognize an action, the Support Vector Machine (SVM) is used [15]. For the activity recognition task the binary SVM has two classes: the descriptors of the action sequences and the descriptors of the sequences without the action.

V. Final decision module

Action recognition is a long term continuous process. When a new person appears in front of the REEM robot, the ROI buffer takes some time to be filled. After the accumulation process the buffer has enough images to transmit them to the action recognition module. The ROI's sequence is processed and an answer comes up from the recognition module. Therefore it takes a few frames to have the first recognition result after a new ROI is localized. Afterwards, each new frame with the tracked ROI brings one more result.

The process becomes a sliding window along the time axis, see Figure 7. In the implemented approach, a single decision (action recognition module output) is not considered as a final answer. The final decision is made conforming to a sequence of the frame decisions.

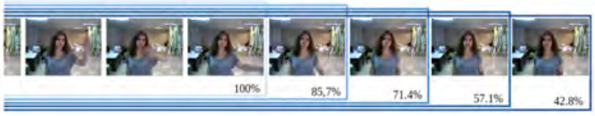


Fig. 7: Time scale of the voting process

The local answers are stored in a buffer, called the voting vector. When the vector size reaches the fixed value the decision process takes place. One voting vector is a set of binary answers (yes or no). The decision process considers a percentage of the positive answers, A_{pos} , and the last answer value. When the calculated percentage is higher than a given threshold and the last answer is positive, the final decision is positive and the movements are counted as an action.

The last positive answer is an important constraint for the comparison. It reduces the false positive results, when a person stops doing an action but is still standing in front of the robot. In this case, the tracking buffer is still full, that forces to continue the evaluation process. The voting vector in turn is filled with the positive answers and even the action is not recognized any more, the percentage of the positive answers in the vector is still higher than the threshold and the final decision is still positive. Hence there is the condition for the last positive answer that doesn't happen.

The output answer of the module is the final decision.

VI. Experiments

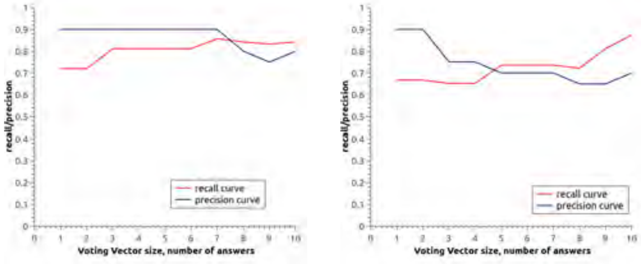
The hand waving is the main action we aimed to detect in this work. Nevertheless to evaluate the approach and compare the results of the hand waving with some other action, the sitting down detection results are also evaluated in the Section.

A. Activity recognition parameters

There are two important parameters which have a big influence on the final decision accuracy. *The size of the voting vector* defines the number of considered answers and *the threshold of positive values* characterizes the needed percentage of positive answers in the voting vector for the positive decision.

To tune the parameters a set of tests were run. The dataset of already tracked ROIs is used for the tests. The results present the accuracy of the recognition approach without the tracking part, so they are more stable and the influence of the parameters is more clear.

1) Voting vector size: it is changed from 1 to 10 answers. The threshold value is fixed up to 50 %. The data is presented on the Figure 9a and Figure 9b). The maximum values of the recall and the precision point to the most accurate solution.



(a) Hand waving action

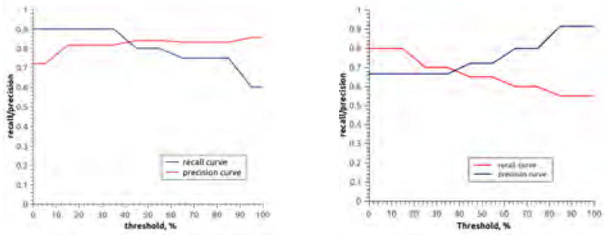
(b) Sitting down action

Fig. 8: Recall and precision for the different voting vector sizes

According to the plots, the best voting vector sizes is equal 7 for the hand waving action and 5 for the sitting down action.

2) Positive decision threshold: the parameter is studied for the voting vector size of 7 answers in hand waving case and 5 answers in case of sitting down. The threshold is changed from 0 to 100 percentages. The results are evaluated by the recall and the precision for each threshold case, Figure 10.

According to the plot, the maximum values of the recall and the precision are achieved for the 40% threshold value for both actions.



(a) Hand waving action

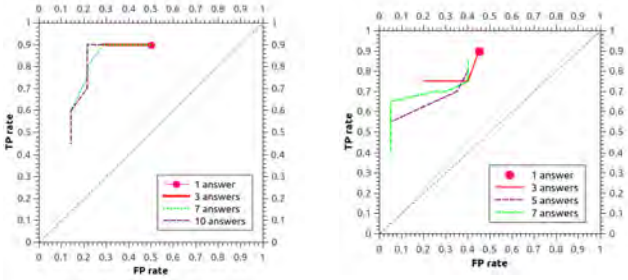
(b) Sitting down action

Fig. 9: Recall and precision for the different threshold values

B. Accuracy evaluation

The experiments are aimed to test accuracy of the approach. The results of the experiments are evaluated by the ROC curve, which presents the correspondence between True Positive rate (sensitivity or recall) and False Positive rate (specificity). The closer the curve follows the left and the top borders of the ROC space, the more accurate the test. [16].

The tests are run for the created ROI sequences databases. The ROC curves are built for the four different sizes of the Voting Vector and the threshold is changed from 0 to 100 %. The results are presented for the two different actions: hand waving and sitting down.



(a) Hand waving detection

(b) Sitting down detection

Fig. 10: ROC curve for the different voting vector sizes

The test also were run for the ROS bag file datasets. The files are recordings of the published data in ROS.

The hand wave detector was set up for 7 answers in the voting vector and 40 % threshold in the decision process. The results are presented in the table I.

TP	TN	FP	FN	TP rate	FP rate
18	19	3	3	0.857	0.136

TABLE I: Results of the waving action dataset. Voting vector size = 7, threshold=40%

The sitting down detector was run for the 5 answers in the vector size and the threshold 40%. The results are presented in the table II.

True Positive appears when at least one positive result is detected during the hand waving action (5 seconds). False

TP	TN	FP	FN	TP rate	FP rate
5	8	2	6	0.5	0.2

TABLE II: Results of the sitting down action dataset. Voting vector size = 5, threshold=40%

Negative is defined when no positive answers are detected during the hand waving action (5 seconds). False Positive is the detected hand waving when there was no hand waving action. True Negative appears when no positive results are found for not-waving action.

C. Discussion

The tests were run for ROI sequences and ROS bag files databases. The ROI sequences allow to test the accuracy of the implemented approach and define an influence of the parameters. The values of the ROC curves are above the guess line, therefore the approach is applicable for the hand waving as well as for the sitting down action recognition.

The running tests with the rosbag datasets consider influence of the ROI allocation and tracking on the final decision. The results for the hand waving is more accurate comparing to the sitting down action results.

The sitting down and hand waving actions are different in the way of movement. The hand waving action is using hands only, it can be characterised as a gesture. While the sitting down is the movement of the whole body. Hence the experiments show that the developed approach is applicable for both type of activities.

VII. Conclusions and Future Works

The developed approach is based on the Temporal Template representation of movements. An action is represented as a space-time volume and projected by the Motion History Image (MHI). Histogram of Oriented Gradients (HOG) descriptor characterises the image. The classification task is solved by a Support Vector Machine.

Before the action recognition process, Regions of Interest (ROI) are located for each frame in the sequence. The ROI is defined according to a data from the full-body and the face detectors. The regions are tracked and stabilized using a data from the leg detector and the buffer which contains previous ROI positions.

The solution is integrated inside a ROS-based environment and it can be ran in real time on the REEM robot. An action can de detected after 3 seconds of observation, assuming that the ROI is detected correctly.

Two types of actions were considered to test the approach on the robot: hand waving and sitting down. The set of experiments were run to define correct values of the voting vector size and the threshold for the final decision. Those parameters are different for the actions, while the observation time parameter, which defines the Motion History Image is set automatically and doesn't depend on the action type.

The accuracy of the action recognition module was investigated with the different parameter sets. The results for both actions show good accuracy.

The proposed method can be migrated as a visual activity recognition module to different robots, on conditions that the robot is equipped with at least one camera.

In spite of the positive results there are drawbacks of the solution. First of all, the number of the possible recognisable actions are limited by the Temporal Template method. To leave out the limitation the concept can be extended to a combination of the simple actions. For instance, the combination of MHI can describe more sophisticated actions, versus the presented solution.

Moreover the activity recognition application is constructed by modules, so there is a possibility to use another action representation, instead of the Temporal Templates. Depending on the representation, the solution can achieve higher accuracy and be able to detect more actions.

The approach can be also improved by the background subtraction in advance to the Region of Interest detection. The algorithm can be extended to use depth information. As REEM has a stereo camera, the depth image provided by dense stereo could be used to filter out the background in the Regions of Interest.

BIBLIOGRAPHY

References

[1] "REEM technical specification. PAL Robotics.," 2013.

[2] R. Poppe, "A survey on vision-based human action recognition," *Nucl. Phys.*, vol. B528, p. 35, 2011.

[3] M. Y. Sheikh and M. Shah, "Exploring the space of a human action," *the IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2005.

[4] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23(3), pp. 257–267, 2001.

[5] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conference*, pp. 995–1004, sep 2008.

[6] A. K. I. L. H. Wang, M. M. Ullah and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," *BMVC'09*, 2009.

[7] Y. Yacoob and M. Black, "Parameterized modeling and recognition of activities," *the IEEE International Conference on Computer Vision (ICCV)*, 1998.

[8] J. O. J. Yamato and K. Ishii, "Recognizing human action in time-sequential images using hidden markov models," *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 379–385, 1992.

[9] E. H. N. Oliver and A. Garg, "Layered representations for human activity recognition," *the IEEE International Conference on Multimodal Interfaces (ICMI)*, pp. 3–8, 2002.

[10] Y. Ivanov and A. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 8, pp. 852–872, 2000.

[11] S. Sundaram and W. W. Mayol-Cuevas, "What are we doing here? egocentric activity recognition on the move for contextual mapping," *Department of Computer Science, University of Bristol*, 2013.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *CVPR*, pp. 886–893, 2005.

[13] P. I. Wilson and D. J. Fernandez, "Facial feature detection using haar classifiers," *JCSC*, vol. 21, 4, 2006.

[14] M.-O. Arras, K.O. and W. Burgard, "Using boosted features for the detection of people in 2d range data," *the IEEE International Conference Robotics and Automation*, 2007.

[15] J. S.-T. N. Cristianini, "An introduction to support vector machines: and other kernel-based learning methods," *Cambridge University Press, New York*, 2000.

[16] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *In ICML 06: Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, ACM Press, 2006.

Sparse Coral Classification Using Deep Convolutional Neural Networks

Mohamed E. Elawady, Neil M. Robertson, and David Lane

Abstract— **Autonomous repair of deep-sea coral reefs is a recent proposed idea to support the oceans ecosystem in which is vital for commercial fishing, tourism and other species. This idea can be operated through using many small autonomous underwater vehicles (AUVs), swarm intelligence techniques, and machine vision algorithms to locate and replace chunks of coral which have been broken off, thus enabling re-growth and maintaining the habitat. We present an efficient sparse classification for coral species using supervised deep learning method called Convolutional Neural Networks (CNN). in which it is evaluated using computation of Weber Local Descriptor (WLD), Phase Congruency (PC), and Zero Component Analysis (ZCA) Whitening to extract shape and texture feature descriptors, which are employed to be supplementary channels (feature-based maps) besides basic spatial color channels (spatial-based maps) of coral input images from two different coral datasets (University of California San Diego's Moorea Labeled Corals, and Heriot-Watt University's Atlantic Deep Sea), we also experiment state-of-art preprocessing underwater algorithms for image enhancement and color normalization and color conversion adjustment.**

I. INTRODUCTION

Coral reef ecosystems provide for over half a million people: they create substantial socioeconomic benefits from tourism and fisheries while providing coastal protection, enhancing biodiversity and contributing to carbon sequestration that mitigates global warming [1], [2]. Global conservation of reefs and their resources in a world characterized by multiple stressors and disturbances will require unified efforts to create international marine and climate policies alongside local adaptive community management tools [3].

A. Coral Threats

Based on mid-90's statistics [4], 10% of coral reefs were destroyed and can't be recovered again, and there are only less than 30% healthy coral reefs around the world. Figure 1 shows human activities that threaten the coral reefs around the world starting from Caribbean coast of Atlantic ocean; passing through east Africa coast & Red sea; ending to central part of Pacific ocean, those activities involve coastal development (sun blocking from eroding soil over aquatic world), underwater pollution (oil, gas, and mineral exploration & extraction), destructive fishing methods (very popular in south Pacific and southeast Asia using poison fishing and dynamite fishing), and unsustainable tourism (i.e. touching during diving sessions), such that they damage both cold deep and warm shallow corals physically and don't allow them to grow again or recover in decades [5], [6].

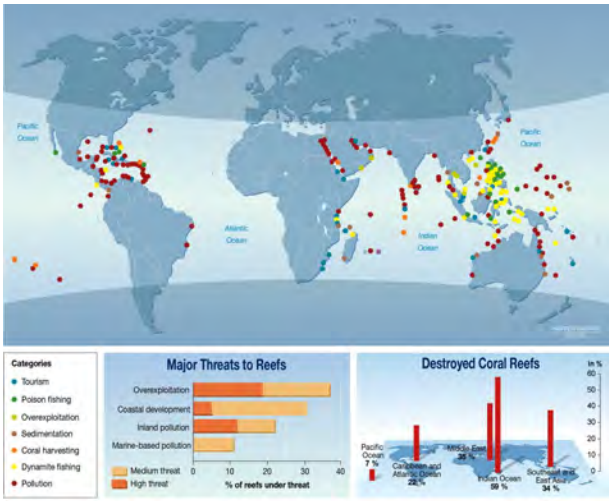


Fig. 1: Threads to Coral Reefs [5]

B. Coral Transplantation

Some types of coral reef have a slow survival ability for recovering or re-growth using small healthy coral wreckage resulting their artificial coral colony after some decades. Possible strategies are provided for coral gardening through involvement of SCUBA divers in coral reef reassemble and transplantation. Although, some limitations (restricted time and depth per diving session respect to human abilities) are introduced a small survival rate in transplanted corals (especially cold sea corals due to their deep depth conditions). Coral ecologists investigate new robot-based strategy in deep-sea coral restoration in such that autonomous underwater vehicles (AUVs) grasp cold-water coral samples and replant them in damaged reef areas. Successful transplantation trail [7] is already occurred in 2008 for cold-water coral Lophelia (at 82m water depth) in Kosterfjord, Sweden.

As deployment of single AUV operation is time limited. Inspired from behavioral of natural swarms of insects (bees, wasps and termites) in building complex colonies, team of marine biologists and robotics experts introduced an innovative underwater project 'coralbots' to speed-up the regenerated coral process using intelligent swarms of interconnected AUVs. Proposed work consists of two stages: offline data training and online identification. offline training

This work was supported by Coralbots

M.E. Elawady is with Erasmus Mundus Master Course in Vision and Robotics, Heriot-Watt University, Riccarton Campus, Edinburgh EH14 4AS, United Kingdom. mea30@hw.ac.uk

N.M. Robertson is with the Visionlab, Heriot-Watt University, Riccarton Campus, Edinburgh EH14 4AS, United Kingdom. N.M.Robertson@hw.ac.uk

D. Lane is with the Ocean Systems Lab, Heriot-Watt University, Riccarton Campus, Edinburgh EH14 4AS, United Kingdom. D.M.Lane@hw.ac.uk

will be on surface workstation for fast computation and long execution time which extracts features from coral-labeled images besides spatial information and then apply deep learning process (supervised, unsupervised, or hybrid) to get well-trained parameters for further successful classification. However, online identification will be on remotely operated underwater vehicle (ROV) which collect images from several AUVs and find out which species are included, and detect their coordinates in real-time processing for further coral transplantation.

II. RELATED WORK

This chapter discusses most-recent research in classification for coral species using optical camera sensors, then explains convolutional neural networks (deep learning method) as a feature extraction and classification technique.

A. Coral Classification

Marcos [8] developed an automated rapid classification (5 classes: coral, sand, rubble, dead coral, and dead coral with algae) for underwater reef video, he used color features based on histogram of normalized chromaticity coordinates (NCC) and texture features from local binary patterns (LBP) descriptor, those features feed into linear discriminant analysis (LDA) classifier. In case of using more classes [9], his method output inaccurate classification. Strokes [10] described an automated algorithm for the classification of coral reef benthic organisms and substrates which divides image into blocks, then finds distance between those blocks and identifies species blocks based on color features (normalized histogram of RGB color space) and texture features (radial samples of 2D discrete cosine transform) by using inconvenient distance metric (manually assigned parameters) after unsuccessful results of well-known mahalanobis distance. Beijbom [11] introduced Moorea Labeled Corals (MLC) dataset and proposed multi-scale classification algorithm for automatic annotation, he developed color stretching for each channel individually in L*a*b* color space as pre-processing step, then used Maximum Response (MR) filter bank approach (rotation invariant) as color and texture feature, followed by applying Radial Basis Function kernel (RBF) of Support Vector Machines (SVM) classifier, this method seeks all possibilities (time-consuming) to find a suitable patch size around selected image points for species identification.

Rather than depending on human-crafted features to get a proper coral classification, the proposed work decides letting the feature mapping to be done automatically by deep convolutional neural networks regardless to any under-water environment condition. by feeding new images, the network can learn and adapt the constructed feature maps respect to desired class outputs.

B. Convolutional Neural Networks

Traditional architecture firstly extracts hand-designed key features based on human analysis for input data, secondly applies those features in form of data vectors to generic classifier in order to get predicted target classes (in other words, classifier is totally dependent on how features constructed not input data). Deep architecture trains learning features across hidden layers; starting from low level details (i.e. edges, corners) up to high level details (i.e. shape, texture); to get better data representation for simple classifier (please see figure 2 for graphical details).

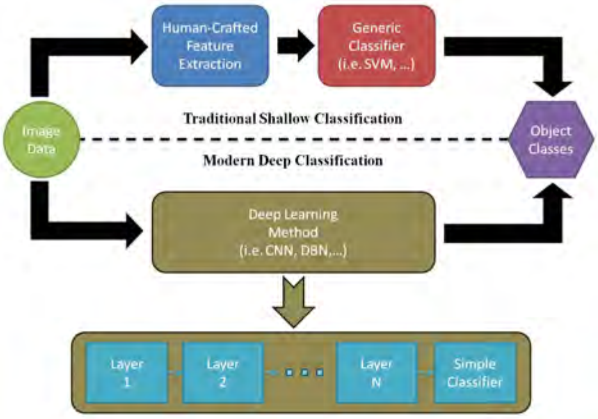

Human-Crafted Feature Extraction
Generic Classifier (i.e. SVM, ...)
Image Data
Traditional Shallow Classification
Modern Deep Classification
Object Classes
Deep Learning Method (i.e. CNN, DBN,...)
Layer 1
Layer 2
Layer N
Simple Classifier

Fig. 2: Difference between shallow traditional and deep modern classification architectures

A convolutional neural network (CNN) [12], [13] is a type of feed-forward back-propagation neural networks respect to biological-based visual processes. it consists of trainable multiple convolutional stages [14], in which input and output of each stage are variant representation of one/multi-dimensional array (i.e. 1D for audio, 2D for image, 3D for video, ...), the output data is learned to extract high-receptive features from all sides of input one. A typical CNN is composed of two or three hidden stages, followed by a classification layer. LeCun presented first back-propagation CNN entitled "LeNet-5" for handwritten digit recognition, which is a large network which contains 6 layer hidden layers whose its input is 28x28 input image of single handwritten character and its output is multi-invariant feature map of input character.

Each hidden stage/layer consists of four steps: trainable convolution, non-linearity activation, contrast normalization, and pooling/sub-sampling. Convolution filters an input map into translation-invariant maps with different trainable weights and biases, Non-linear activation function (i.e. hyperbolic, sigmoid,) adds independent relationship within objects inside, contrast normalization keeps output maps in pre-defined range measures, final feature map is subsampled or max-pooled from output maps of the last stage (make it small in size to further faster calculation in next layers).

III. METHODOLOGY

The proposed classification framework (as shown in figure 3) contains three main levels (input layer, hidden layers, output layer). Input layer consists of three basic channels of color image plus extra channels for texture and shape

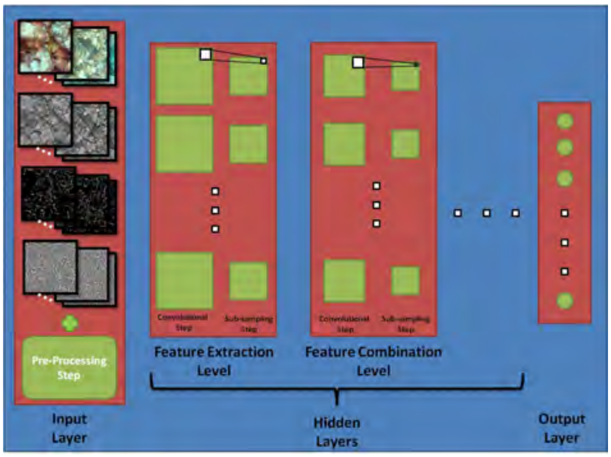


Fig. 3: Architecture overview of proposed CNN

descriptors consisting of following components: zero component analysis whitening, phase congruency, and Weber local descriptor (as shown in figures 4), and preprocessing step (color correction/enhancement, smoothing filter) can be applied for further classification improvement. Hidden layers contains one or more layer(s) [usually 2 or 3] in which each layer consists of convolution layer followed by down-sampling layer in such way that the network can find suitable weights of convolutional kernel and additive biases. Almost first layer represents feature extraction by finding visual strokes, edges, and corners, and up-coming layers starting from second layer show how those features can combine in different aspects to get a discriminative output map for each target class. Output layer acts as a classification layer and symbolize reconstructed maps from last hidden layer into binary vector (placement of number one in specific element corresponding to desired class and number zero in the rest elements).

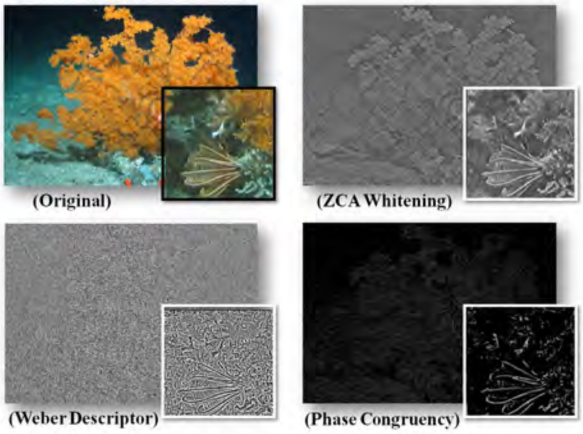


Fig. 4: Example of feature maps for Crinoid coral using ADS dataset

A. Preprocessing

1) Hybrid Patching: Three different-in-size patches are selected across each annotated point (61x61, 121x121, 181x181), then unified size scaling step is applied to those patches by scaling them up to size of the largest patch (181x181) allowing pixel randomization (blurring) in inter-shape coral details and keeping up corals edges and corners (please see figure 5), or scaling them down to size of the smallest patch (61x61) for fast classification computation over small data representation of different scaling selections.

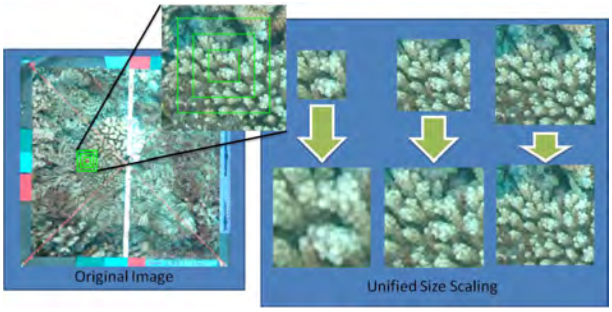


Fig. 5: Example of hybrid patching

2) Zero Component Analysis Whitening: Zero Component Analysis (ZCA) whitening [15] makes data less-redundant by removing any neighboring correlations in adjacent pixels in such that output data removes amplitude information and keeps recognizable edges. it stimulates image scanning retinal process which decorrelates similar intensity values of contiguous pixels (high correlated adjacent pixels) after few moments of eye-focusing. It requires one smoothing parameter (very small number) preventing division of zero in its calculation with respect to tiny eigenvalues which leads to a better-visual output features (dispatching off the inter-process aliasing artificats).

3) Weber Local Descriptor: Weber Local Descriptor (WLD) [16] is inspired from psychological law in 19th century "Weber's Law" and represents human perception of a pattern depending on ratio between change in image pixel and original pixel value. it consists of two components: differential excitation and orientation. The differential excitation component computes the salient micro-patterns relative to nearby neighboring pixels by calculating a function of the ratio between ratio between the relative intensity differences of a current pixel against its neighbors and the intensity of the current pixel. The orientation component constructs statistics on the computed salient patterns along with the gradient orientation of current pixel by building histograms of dominant orientations. This method shows a robust edge representation of high-texture images against high-noisy changes in illumination of image environment. WLD has proven promising results in different object recognition issues [17], [18], [19].

4) Phase Congruency: Phase Congruency [20], [21] represents image features in such format which should be high in information and low in redundancy using Fourier transform, rather than set of edges (sharp changes in intensity). in other words, Phase Congruency [22] is a dimensionless measure for the of a image structure independently of the signal amplitude which is based on Kovesi's work [23]. Those features are better than gradient-based features which are fully invariant to image illumination and contrast, and also partially invariant to scale and rotation transformation in case of application of suitable normalization process in frequency domain [24].

B. Network Architecture

1) Kernel weights & bias initialization: The network [25] initializes biases to zero, and kernel weights using uniform random distribution using the following range:

$$\begin{aligned} rng = \pm\sqrt{6 * (f_{in} * f_{out})}. \\ f_{in} = N_{in} * K^2. \\ f_{out} = N_{out} * K^2. \end{aligned} \tag{1}$$

where N_{in} and N_{out} represent number of input and output maps for each hidden layer (i.e. number of input map for layer 1 is 1 as gray-scale image or 3 as color image), and k symbolizes size of convolution kernel for each hidden layer.

2) Convolution layer: Convolution layer constructs output maps by convoluting trainable kernel over input maps to extract/combine features for better network behavior using the following equation:

$$x_j^l = f(\sum_{i \epsilon m_j} [{x_i}^{l-1} * k_{ij}^l + b_j^l]). \tag{2}$$

where ${x_i}^{l-1}$ & x_j^l are output maps of previous $(l-1)$ & current (l) layers with convolution kernel numbers (input i and output j), $f(.)$ is activation function for calculated maps after summation, and b_j^l is addition bias of current layer l with output convolution kernel number j.

3) Down-sampling layer: The functionality of down-sampling layer is dimensional reduction for feature maps through network's layers starting from input image ending to sufficient small feature representation leading to fast network computation in matrix calculation, which uses the following equation:

$$y_j^l = h_n({}^l * x_j^l). \tag{3}$$

where h_n is non-overlapping averaging function with size nxn with neighborhood weights w and applied on convoluted map x of kernel number j at layer l to get less-dimensional output maps of kernel number j at layer l (i.e. 64x64 input map will be reduced using n=2 to 32x32 output map).

4) Activation function: The logistic (sigmoid) function which is the most common activation function for classical neural networks and very useful in gradient decent training due to existence of function's derivatives. the function's equation is as follows:

$$f(x) = \frac{1}{1 + e^{-\beta x}}; [-\infty, +\infty] \Rightarrow [0, 1]. \tag{4}$$

where input x can be infinite value, and output $f(x)$ will be in bounded range [0,1].

5) Learning rate: Inspired from Lawrence's convergence learning rate in CNN application for face recognition [26], an adapt learning rate is used rather than a constant one with respect to network's status and performance as follows:

$$\alpha^n = g(\frac{\alpha^{n-1}}{[n/(N/2)] + 1} + e^n). \tag{5}$$

where α^n & α^{n-1} are learning rates of current & previous iterations (if first network iteration is the current one, then learning rate of previous network iteration represents initial learning rate as network input), n & N are number of current network iteration & total number of iterations, e^n is back-propagated error of current network iteration, and $g(.)$ is linear limitation function to keep value of learning rate in range $(0, 1]$.

IV. RESULTS

This section shows the results of sparse classification with hybrid patching around annotated points using convolutional neural networks initially referring to Palm's toolbox for deep learning [14], in which it discusses results for best configuration selection, and shows output representation of the proposed method with respect to selected configuration.

A. Evaluation Metrics

There are many popular assessment methods for quantitative measures in classification problems. The statistics of confusion matrix (contingency matrix)[**?**] is general quantitative representation of relationship between target classes and algorithm output classes, resulting of some important accuracy quantities (overall accuracy "OA", precision, recall, sensitivity, specificity, and F-score). training and test errors are also used to validate classification performance over different selection of network parameters.

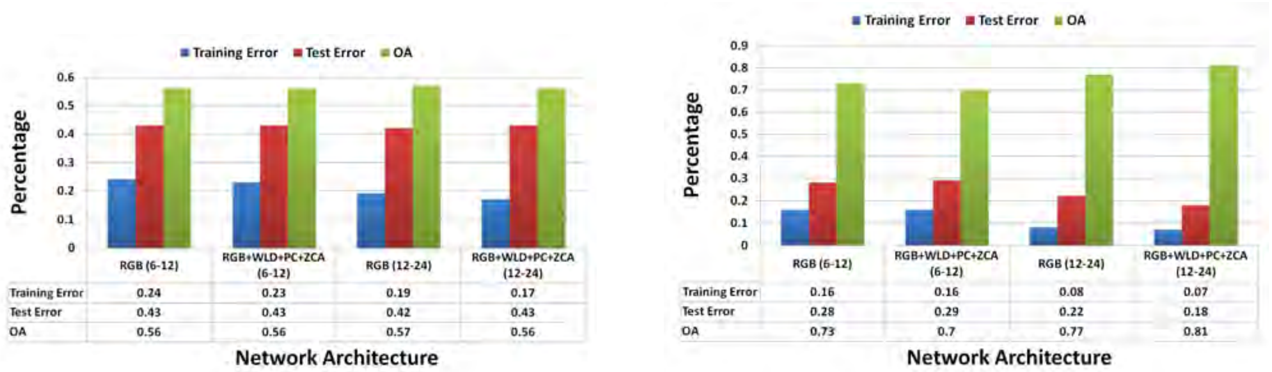
B. Network parameters

Finding best network architecture and validating its performance needs to compare quantitative results with keeping the rest network parameters constant (size of hybrid input image = 181x181, number of output classes = 9, number of samples per class = 300, normalization method = min-max with in range [-1,+1], initial learning rate = 1, network batch size = 3, and ratio of training/test sets = 2:1).

C. Experimental Results

In large-scale experiments of 50 network epochs, testing phase of MLC dataset has almost the same results across different configurations as shown in figure 6a, but training phase starts converging to correct target classes by increasing number of hidden output maps (12-24) and using additional feature-based maps as supplementary channels. Using ADS dataset in figure 6b, testing phase has best significant accuracy results with same selected configuration for MLC dataset.

Sub-figures 7a, 7d represent confusion matrices for MLC and ADS dataset, in which rows & columns represent the assignments of target classes & predicted output classes



(a) MLC dataset

(b) ADS dataset

Fig. 6: Comparison of network architecture

respectively. In MLC dataset, the highest classification rates are for Acrop (coral) and Sand (non-coral), and the lowest classification rates are for Pavon (coral) and Turf (non-coral), where misclassification occurred outputting Pavon as Monti/Macro and Turf as Macro/CCA/Sand due to similarity in their shape properties or growth environment. However in ADS dataset, non-corals has better classification rate then corals, where DRK (non-coral) has almost perfect classification rate due to its distinct nature (almost dark blue plain image), LEIO (coral) has excellent classification rate due to its distinction color property (orange), and LOPH (coral) & ENCW (coral) has lowest classification rates due to their color confusion with each other & with BLD (non-coral).

Sub-figures 7b, 7e show the evolution of training and test errors in MLC and ADS datasets across network epochs, such that the proposed method's errors have better convergence curves (almost half) with ADS dataset over MLC dataset. From epoch 30 in MLC dataset, increased gap starts to appear between training and test errors leading to algorithm over-fitting over training data. From epoch 35 in ADS dataset, training and test errors are almost stagnant (no major improvement) with respect to typical evolution of neural networks. MLC and ADS datasets have similar evolution curves (Sub-figures 7c, 7f) for learning rate across presented network epochs.

V. CONCLUSIONS AND FUTURE WORKS

The proposed framework presented first investigation of deep learning techniques (especially convolutional neural networks) in a supervised sparse-based classification method for coral species, investigated computation of supplementary channels (feature-based maps) besides basic spatial color channels (spatial-based maps) to act as coral input data, introduction of new coral-labeled dataset "Atlantic Deep Sea" representing cold-water coral reefs, and hybrid image patching procedure for multi-size scaling across different square-based windowing around labeled points. Although, the implementation of classification method lacks fast performance of proposed algorithm and handling large-sized input data.

Future work of proposed method will cover avoiding information loss of dimension reduction for convolutional neural networks, composition of multiple deep convolutional models for N-dimensional data, development of real-time image/video application for coral recognition and detection, code optimization and improvement to build GPU computation for processing huge image datasets and edge enhancement for feature-based maps, finally intensive nature analysis for different coral classes in variant aquatic environments.

ACKNOWLEDGMENTS

We would like to thank cold-water coral group from Heriot-Watt University for providing different types of proficient information about coral reefs and maintaining new cold-coral dataset. we would like also to thank Rasmus Berg Palm from Technical University of Denmark for publishing Deep Learning Toolbox online, Oscar Beijbom from University of California, San Diego for his public implementation of point-based coral classification.

REFERENCES

[1] N. Foley and C. W. Armstrong, "The ecological and economic value of cold-water coral ecosystems," pp. 1–39, 2010.

[2] A. Compilation, "Economic Values of Coral Reefs, Mangroves, and Seagrasses," 2008.

[3] E. V. Kennedy, C. T. Perry, P. R. Halloran, R. Iglesias-Prieto, C. H. L. Schönberg, M. Wisshak, A. U. Form, J. P. Carricart-Ganivet, M. Fine, C. M. Eakin, and P. J. Mumby, "Avoiding coral reef functional collapse requires local and global action.," *Current biology : CB*, vol. 23, pp. 912–8, May 2013.

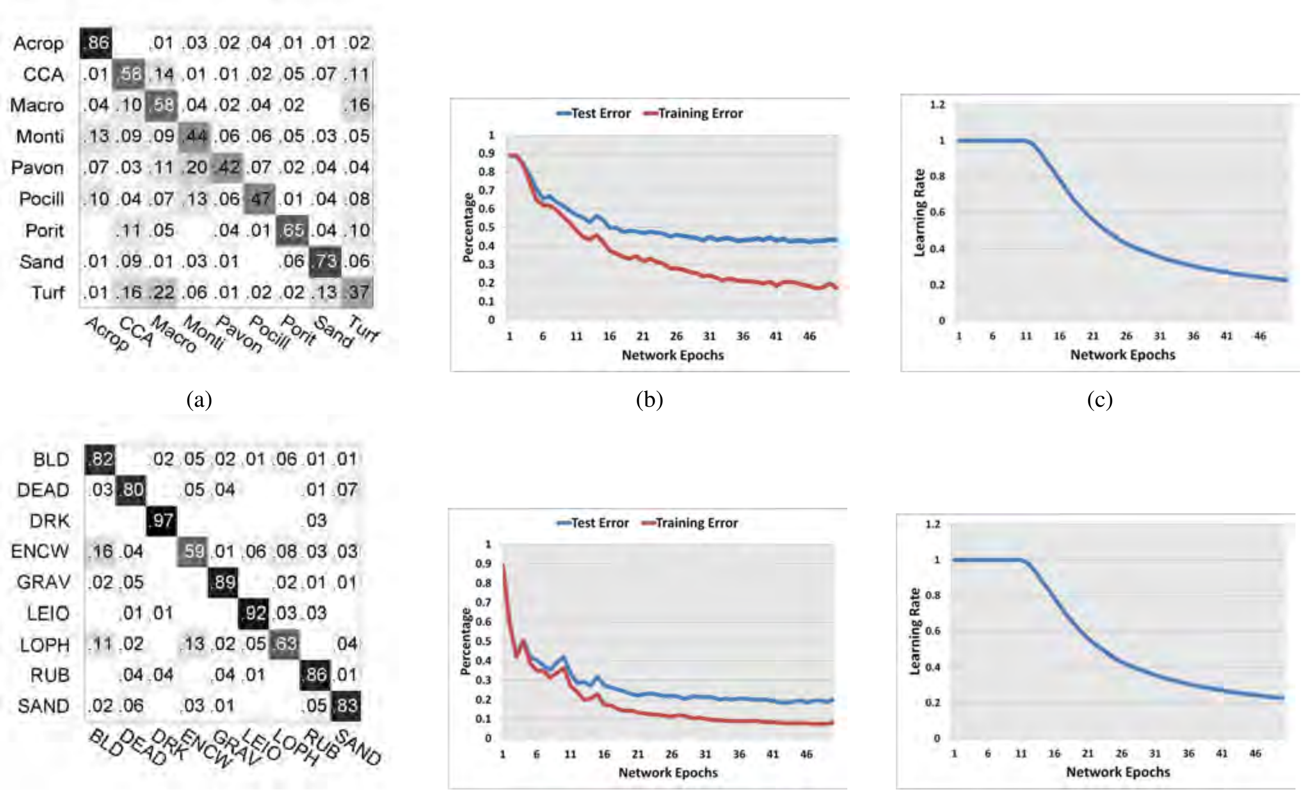
[4] I. C. R. A. N. ICRAN, "Coral Reefs: Ten Questions - Ten Answers." `http://www.icran.org/peoplereefs-tenquestions.html`. [Online; accessed 25-May-2014].

[5] S. Diop, P. M'mayi, D. Lisbjerg, and R. Johnstone, *Vital Water Graphics: An Overview of the State of the World's Fresh and Marine Waters*, vol. 1. UNEP/Earthprint, 2002.

[6] G. Death, K. E. Fabricius, H. Sweatman, and M. Puotinen, "The 27–year decline of coral cover on the great barrier reef and its causes," *Proceedings of the National Academy of Sciences*, vol. 109, no. 44, pp. 17995–17999, 2012.

[7] J. M. Roberts, *Cold-water corals: the biology and geology of deep-sea coral habitats.* Cambridge University Press, 2009.

[8] M. S. A. Marcos, L. David, E. Peñaflor, V. Ticzon, and M. Soriano, "Automated benthic counting of living and non-living components in Ngedarrak Reef, Palau via subsurface underwater video.," *Environmental monitoring and assessment*, vol. 145, pp. 177–84, Oct. 2008.



	Acrop	CCA	Macro	Monti	Pavon	Pocill	Porit	Sand	Turf
Acrop	.86		.01	.03	.02	.04	.01	.01	.02
CCA	.01	.58	.14	.01	.01	.02	.05	.07	.11
Macro	.04	.10	.58	.04	.02	.04	.02		.16
Monti	.13	.09	.09	.44	.06	.06	.05	.03	.05
Pavon	.07	.03	.11	.20	.42	.07	.02	.04	.04
Pocill	.10	.04	.07	.13	.06	47	.01	.04	.08
Porit		.11	.05		.04	.01	.65	.04	.10
Sand	.01	.09	.01	.03	.01		.06	.73	.06
Turf	.01	.16	.22	.06	.01	.02	.02	.13	37

(a)

(b)

(c)

	BLD	DEAD	DRK	ENCW	GRAV	LEIO	LOPH	RUB	SAND
BLD	.82		.02	.05	.02	.01	.06	.01	.01
DEAD	.03	.80		.05	.04			.01	.07
DRK			.97					.03	
ENCW	.16	.04		59	.01	.06	.08	.03	.03
GRAV	.02	.05			.89		.02	.01	.01
LEIO		.01	.01			.92	.03	.03	
LOPH	.11	.02		.13	.02	.05	.63		.04
RUB		.04	.04		.04	.01		.86	.01
SAND	.02	.06		.03	.01			.05	.83

(d)

(e)

(f)

Fig. 7: Evaluation Metrics for selected network architecture: (a,b,c) MLC dataset, (d,e,f) ADS dataset

[9] a.S.M. Shihavuddin, N. Gracias, R. Garcia, A. Gleason, and B. Gintert, "Image-Based Coral Reef Classification and Thematic Mapping," *Remote Sensing*, vol. 5, pp. 1809–1841, Apr. 2013.

[10] M. Stokes and G. Deane, "Automated processing of coral reef benthic images," *Limnol. Oceanogr.: Methods*, pp. 157–168, 2009.

[11] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1170–1177, June 2012.

[12] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5-6, pp. 555–559, 2003.

[13] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, pp. 9–48, Springer, 2012.

[14] R. Palm, "Prediction as a candidate for learning deep hierarchical models of data," *Technical University of Denmark, Palm*, 2012.

[15] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–38, 1997.

[16] C. J., S. S., H. C., Z. G., P. M., and C. X. . G. W., "WLD: A robust local image descriptor.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, 2010.

[17] A. Pal, N. Das, S. Sarkar, D. Gangopadhyay, and M. Nasipuri, "A new rotation invariant weber local descriptor for recognition of skin diseases.," in *PReMI* (P. Maji, A. Ghosh, M. N. Murty, K. Ghosh, and S. K. Pal, eds.), vol. 8251 of *Lecture Notes in Computer Science*, pp. 355–360, Springer, 2013.

[18] M. Ghulam, M. Hussain, F. Alenezy, A. M. Mirza, G. Bebis, and H. Aboalsamh, "Race recognition using local descriptors," in *ICASSP*, pp. 1525–1528, IEEE, 2012.

[19] S. Li, D. Gong, and Y. Yuan, "Face recognition using weber local descriptors.," *Neurocomputing*, vol. 122, pp. 272–283, 2013.

[20] M. C. Morrone and D. C. Burr, "Feature detection in human vision: a phase-dependent energy model.," *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, vol. 235, pp. 221–45, Dec. 1988.

[21] A. Oppenheim and J. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 69, no. 5, 1981.

[22] L. Zhang, L. Zhang, D. Zhang, and Z. Guo, "Phase congruency induced local features for finger-knuckle-print recognition," *Pattern Recognition*, vol. 45, pp. 2522–2531, July 2012.

[23] P. Kovesi, "Image features from phase congruency," *Videre: Journal of computer vision research*, no. June, 1999.

[24] A. Burlacu and C. Lazar, "Image features detection using phase congruency and its application in visual servoing," in *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, pp. 47–52, IEEE, 2008.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Nov 1998.

[26] S. Lawrence, C. L. Giles, a. C. Tsoi, and a. D. Back, "Face recognition: a convolutional neural-network approach.," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 8, pp. 98–113, Jan. 1997.

[27] J. Roberts and S. Party, "Changing Oceans Expedition 2012 RRS James Cook 073 Cruise Report," tech. rep., Heriot-Watt University, 2013.

5D scene reconstruction using crowdsourcing images.

Waldemar Franczak, In So Kweon, Cédric Demonceaux, Pascal Vasseur

VIBOT Erasmus Mundus Master Program in Computer Vision and Robotics

I. Introduction

Growing popularity of the scene reconstruction from image resources available online influenced not only the scalability of the solutions as in for example [10] and [1], but also quality of obtained scenes. In one of the most recent works by Shan et al. [7] the quality of their reconstruction was tested with use of Amazon Mechanical Turk, where people were presented with real photo and image taken from the model. Then they had to choose the one that is more realistic, according to their perception. Surprisingly at low resolutions results were quite impressive in favor of the reconstructed model.

Taking all this into consideration, one can find multiple possibilities given by this form of scene reconstruction. Constructing real world levels for gaming industry, 3D scene monitoring, 3D localization using images just to name few. One of the most intuitive is creating 3D maps, which is already a quite common application of this techniques. Looking into the future, 3D maps are most probably the next step in the domain of world mapping. This is especially important in the case of documenting world heritage, such as famous buildings and monuments that might change or be destroyed in time. That is why efficient techniques in temporal changes detection are of very high importance for the development of this domain. And this is one of the main problems discussed in this work.

II. Related works

Although the large-scale reconstruction domain is still relatively young research direction, there have already been presented some promising algorithms that deal with detection of scene changes over time. One of the early works on this topic was done by Schindler et al. [6]. In their approach they try to probabilisticaly infer temporal order of images, based on the detected features in the available set. They continued their aproach in [5], where the framework is applied to reconstructed point cloud and the inference is based on occlusions between objects present in the scene. These solutions provide good framework for ordering the data with respect to time dimension, however require at least some amount of information about when the images were taken. It also does not address directly the problem of changes detection and their representation. Taneja et al. [8] presented an algorithm that explicitly deals with detecting and presenting the changes in urban enviroment. They use the reconstructed model to project points from old images, to new images creating an inconsistency map. The energy minimization approach is then used to remove redundancies and also to include semantic knowledge about objects, that should not be considered in the algorithm, such as cars for example. This technique gives quite impressive results with good scalability and was used in their later work [9], to locate changes in city-scale models. Sakurada et al. [4] uses image pairs from two different time stamps to compute per pixel change in depth. They obtained very high accuracy of the detection however the scalability of the solution might be questionable, due to their image registration technique that requires high number of new images and performing additional reconstruction. One of the main challenges in change detection for large-scale reconstruction is measuring the accuracy of the solution, where sometimes it might be hard to define what the change really is. Solutions by Schindler and Taneja might perform well for large geometrical changes, while work presented by Sakurada is able to detect relatively small differences in the scene. Even more important is the scalability of the solution, that has to be taken into consideration given the characteristics of reconstruction from large databases. In this work image-based algorithms will be presented, that aim for high scalability, but also to recover underlying data as in case of Partial Sum Minimization algorithm.

III. Methodology

A. General process pipeline

Given the set of images I_o taken at time t_0, scene is reconstructed using PMVS. In the result of the reconstruction we obtain a file containing all information regarding reconstructed model. This file serves as an input to the system along with a text file containing directories of q new images from set I_1 registered at time t_1. Once the system is feed with the input first step is to register new set of cameras. This is done by running the SfM process again with position of old cameras fixed. This way we obtain positions of new cameras with respect to the scene coordinate frame. Moreover as a side product of the process, matches between images in sets I_0 and I_1 are found. At this point, in order to perform image based change detection, $K \in \{1...n\}$ nearest neighbors for each image in set I_1 have to be computed. In case where matches between images are not known, this can be done efficiently with use of Kd-Tree structure. However given the matches between both sets we can determine the nearest neighbor $j \in I_0$ of image $i \in I_1$, as the image with highest number of features corresponding to i. In the result we obtain set U_p where $p = \{1...K * q\}$, of pairs of new images with corresponding nearest neighbors. After computation of the nearest neighbors for new cameras we can proceed to the step of change detection. Different algorithms applied for this step are described in details in later sections. The detection step produces a change mask which then has to be projected on the model in order to provide spatial change information. Different mask projection techniques are presented in the consequent section. This general process pipeline is presented on figure 1.

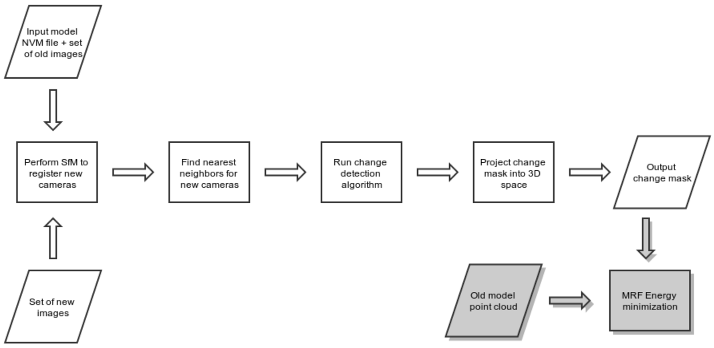

Input model NVM file + set of old images
Perform SfM to register new cameras
Set of new images
Find nearest neighbors for new cameras
Run change detection algorithm
Project change mask into 3D space
Output change mask
Old model point cloud
MRF Energy minimization

Fig. 1. The general change detection process pipeline. Optional energy minimization step was marked with gray color.

B. Change mask projection

The approach of determining temporal changes of spatial model using 2D image space, raises the problem of projecting the change information back to 3-dimensional space. Following subsections describe different methods used to solve this problem in this project. First two methods deal with the projection where correspondence between images and the 3D model is unknown, while the third method leverages information about correspondence between image features and 3D model points. In all cases we assume set of 2D binary change masks C_p where C_i is a mask obtained from the pair U_i and $i \in p$. We define set of 2D points x_p so that point $(x, y)_{iv} \in x_p \iff C_i(x, y) > 0$, where v is the number of elements in C_i.

1) Projection by triangulation: Given a 2D binary change masks C_i and camera parameters for the image pair U_i , we can use triangulation in order to compute change points position in 3D space. First we extract set x_{i1}. Next we compute homography between image pair and project the change mask to the second image in the pair. We extract coordinates x_{i2} from binary mask on second plane. This gives us two sets of corresponding points on two camera planes with known parameters. In the last step we use RANSAC method in order to find final 3D points coordinates.

2) Projection by ray shooting and voxelization: Using the reconstructed model point cloud, one can determine the changes by shooting outward rays from the change mask plane onto the 3D model and find points of intersection. In case of water-tight mesh resulting from for example Poisson Surface Reconstructien on PMVS point cloud the change point on the model is determined by intersection of the ray with the mesh. In case of using surfaceless model, we use point cloud voxelization which gives a possibility to approximate area of change in the model and find intersection with voxels where it would be immpossible for the case of point cloud. Given a binary change mask C_i first we extract detected points x_i. The direction of the ray on which lays the projection of the change point can be easily determined by using the camera matrix to back-project 2D point into 3D space. Once this direction is known we look for the intersection with voxel grid bounding box. When the point of intersection and corresponding voxel is determined, the fast voxel traversal algorithm [2] is applied in order to find first occupied voxel which is then considered as the changed space.

3) Projection using feature correspondence: If we are given with set of 2D features and their 3D correspondences as in the case of the output of SfM, the problem of change mask projection can be simplified to determining which 2D feature falls into change mask on the image plane. Then knowing the corresponding point in 3D model we can show exactly the parts that have changed.

C. Algorithm 1 - image absolute difference

In image change detection domain the simplest and most intuitive technique is to substract one image from another and take the absolute value of the result to avoid negative elements. Although the approach is prone to errors it is still widely used. The advantage in case of considered problem is that image difference is computed for multiple pairs of images representing the same scene. Therefore increased number of data can be used to overcome the downsides that come with this simple technique. Having the image pair U_i and the computed corresponding features we find the homography between images $i \in I_0$ and $j \in I_1$. Next we perform perspective transformation of the image j in order to align it with image i and we compute the absolute difference. We apply inverse perspective transformation to the resulting matrix in order to get rid of outlier areas resulting from transformation.

D. Algorithm 2 - feature grouping

The output of this change detection algorithm is composed of two group of features for considered local part of the model. First group consists of features that are detected and matched in set of old images in the considered area. Second group contains features that are detected and matched in all new images. The contraints for both groups are that the point will be counted as a change only if it appears in all images in given set.

E. Algorithm 3 - partial sum minimization

The last technique is based on work of Tae Hyun Oh [3] on Partial Sum Minimization approach for solving RPCA. The approach for the detection is the same as in the case of algorithm 1, with the difference that new images and computed neighbors are input to PSM solution in matlab. Once the error matrix E is computed, we convert it into a binary mask by setting to maximum all the values greater than zero.

F. Energy minimization

For the purpose of energy minimization we create an octree from the point cloud obtained in SfM process. The data term for each voxel is determined as follows:

$$D_i(L_i) = \begin{cases} 1 & \text{background} \\ \alpha * c_i & \text{change} \end{cases}$$

Where c_i is number of change points in the voxel. The edge potential between voxels $V_i(i, j)$ is determined in following way:

$$V_{i,j}(L_i, L_j) = \begin{cases} 1 & \text{if } c_j > 0 \text{ or } c_i = c_j \\ 0 & \text{otherwise} \end{cases}$$

For each voxel a 26-neighborhood is used. Then the optimization problem is solved with use of Kolmogorov-Boykov algorithm.

IV. Results

The change detection solutions were tested on 3 different datasets for which ROC curves were plotted. In order to compute True Positive Rate and False Positive Rate the the kd-tree was used. For each point in 3D change mask after energy minimization, we were looking for nearest neighbor from the GT point cloud. If the squared distance between this pair was

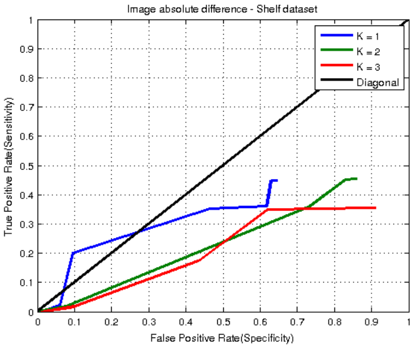


Fig. 2. ROC curve for IAD technique and Shelf dataset.

smaller than the threshold, the point was classified as true positive and false negative otherwise. The threshold was determined by average distance between points in the GT cloud. The values were computed for different parameters of K and α (section III-F). In all cases the change mask was projected on the model with use of point correspondence projection technique that yielded the best results. Figures 2 to 5 present ROC curves for different techniques and datasets. Figure 6 and 7 present time performance of developed solution. Figure 8-11 present exemple of the solution visualization for the KAIST campus dataset.

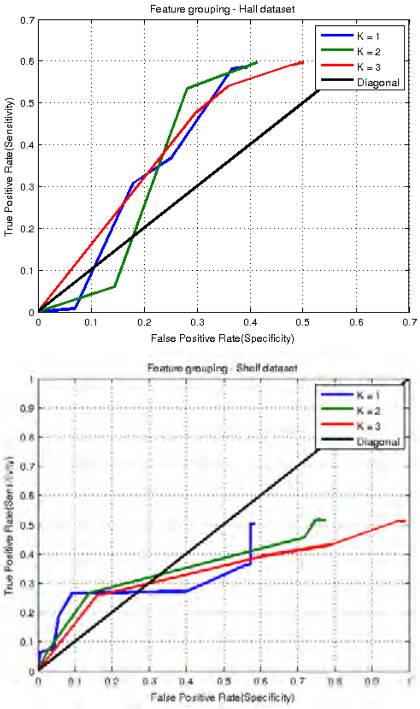


Fig. 3. ROC curves for feature grouping technique in Hall and Shelf dataset.

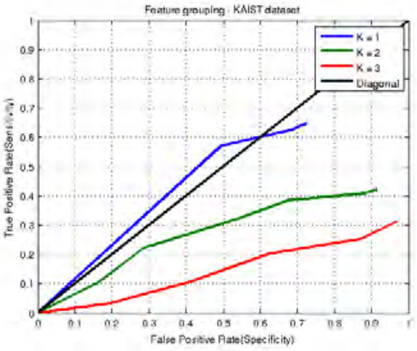


Fig. 4. ROC curve for feature grouping technique in KAIST dataset.

V. Conclusions

The main focus of this work, was the time dimension in form of changes detection in scenes reconstructed with SfM solutions. Three different techniques were implemented for this purpose including novel approach, with use of Robust Principal Component Analysis and results were evaluated quantitatively. The considered task was a very complex and challenging problem, due to the size of the whole process. Multiple factors along the change detection pipeline necessitate handling significant amount of parameters and accounting for number of possible errors in each step. This becomes even more difficult when the goal is a fully autonomous solution. Although the scalability of presented implementations is relatively good considering the time performance, the accuracy of presented algorithms related to the time dimension is in many cases far from desired. Best performance can be noticed for feature grouping algorithm. The IAD and PSM algorithms required computing the homography which, given specific characteristics of the problem, failed very often producing erroneous results. Nevertheless this work makes a an additional contribution through open software developed in the process of this project. It hopefully will help future researchers to avoid extremely time expensive implementation and also allow to be more rapidly introduced into the domain. The software allows to focus solely on the step of change detection, where different

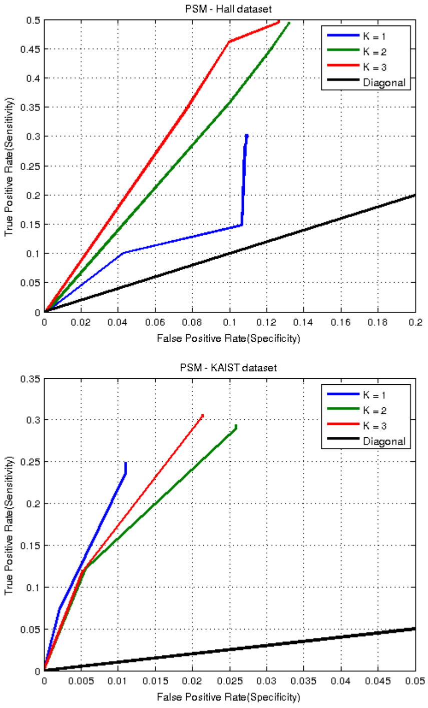


Fig. 5. ROC curves for PSM in Hall and KAIST dataset.

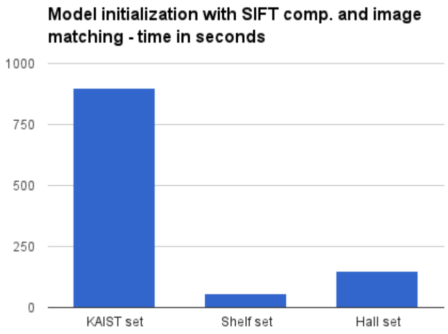


Fig. 6. Time performance for model initialization, SIFT feature extraction from new images and matching with old model.

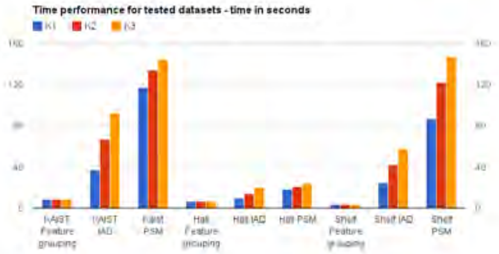


Fig. 7. Time performance of change detection algorithms for different datasets and values of parameter K.

Fig. 8. KAIST dataset dense reconstruction.

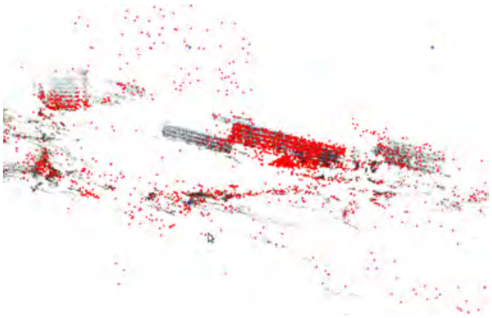


Fig. 9. Detected change points in red color.

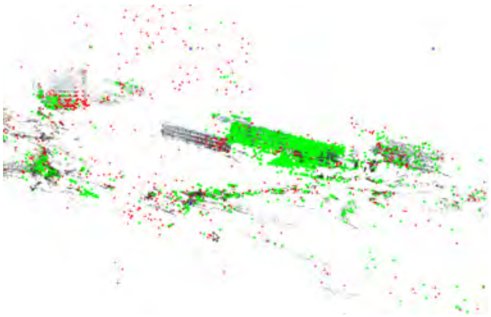


Fig. 10. Green points are the cloud resulting from energy minimization.

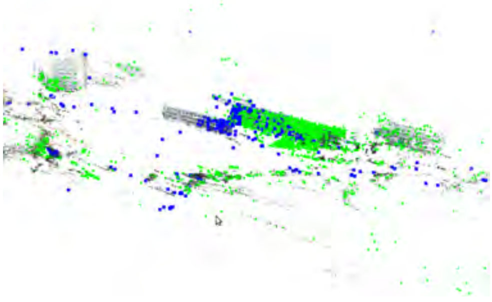


Fig. 11. Blue points present ground truth cloud.

approaches can be easily added and quantitatively evaluated.

References

[1] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle Adjustment in the Large. 2010.

[2] John Amanatides. A Fast Voxel Traversal Algorithm for Ray Tracing. i.

[3] Tae-Hyun Oh, Hyeongwoo Kim, Yu-Wing Tai, Jean-Charles Bazin, and In So Kweon. Partial Sum Minimization of Singular Values in RPCA for Low-Level Vision. *2013 IEEE International Conference on Computer Vision*, pages 145–152, December 2013.

[4] Ken Sakurada, Takayuki Okatani, and Koichiro Deguchi. Detecting Changes in 3D Structure of a Scene from Multi-view Images Captured by a Vehicle-Mounted Camera. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 137–144, June 2013.

[5] Grant Schindler and Frank Dellaert. Probabilistic temporal inference on reconstructed 3D scenes. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1410–1417, June 2010.

[6] Grant Schindler, Frank Dellaert, and Sing Bing Kang. Inferring Temporal Order of Images From 3D Structure. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2007.

[7] Qi Shan, Riley Adams, Brian Curless, Yasutaka Furukawa, and Steven M. Seitz. The Visual Turing Test for Scene Reconstruction. *2013 International Conference on 3D Vision*, pages 25–32, June 2013.

[8] Aparna Taneja, Luca Ballan, and Marc Pollefeys. Image based detection of geometric changes in urban environments. *2011 International Conference on Computer Vision*, pages 2336–2343, November 2011.

[9] Aparna Taneja, Luca Ballan, and Marc Pollefeys. CVPR 2013 Preprint CVPR 2013 Preprint. 2013.

[10] Changchang Wu. Towards Linear-Time Incremental Structure from Motion. *2013 International Conference on 3D Vision*, pages 127–134, 2013.

Complete Tool Allowing Semi-automatic MRI-PET Registration for Preclinical Studies

Hiliwi Leake KIDANE, Alain LALANDE (PhD), Stephanie BRICQ (PhD)
Le2I, Faculty of Medicine
University of Burgundy, Dijon, France

Abstract— **Images from Positron Emission Tomography (PET) deliver functional data such as perfusion and metabolism. On the other hand, images from Magnetic Resonance Imaging (MRI) provides information describing anatomical structures. Fusing the complementary information from the two modality is helpful in oncology. In this project, we implemented a complete tool allowing semi-automatic MRI-PET registration for small animal imaging in the preclinical studies. A two stage hierarchical registration approach is proposed. First, a global affine registration is applied. For robust and fast registration, principal component analysis (PCA) is used to compute the initial parameters for the global affine registration. Since, only the low intensities in the PET volume reveal the anatomic information on the MRI scan, we proposed a non-uniform intensity transformation to the PET volume to enhance the contrast of the low intensity. This helps to improve the computation of the centroid and principal axis by increasing the contribution of the low intensities. Then, the globally registered image is given as input to the second stage which is a local deformable registration (B-spline registration). Mutual information is used as metric function for the optimization. A multi-resolution approach is used in both stages. The registration algorithm is supported by graphical user interface (GUI) and visualization methods so that the user can interact easily with the process. The performance of the registration algorithm is validated by two medical experts on seven different datasets on abdominal and brain areas including noisy and difficult image volumes.**

I. INTRODUCTION

The use of small animal models in preclinical studies constitutes an integral part of testing new pharmaceutical agents and exploring new biological functions[11]. The mouse and the rat are the most widely used animals in medical research because of their small size, rapid breeding, and genetic similarities to humans; and they host a large number of human diseases[31]. Traditional model studies require the animals to be sacriced, prohibiting repeated studies with the same animal[7]. However, with the increased availability of imaging technologies originally developed for human medical diagnosis, it is now possible to study both anatomy and biological processes in the mouse repeatedly and non-invasively[7].

Each medical imaging modality has its own advantages and limitations and acquired information is actually complementary between them[11]. Consequently, multimodal approach is used to reveal both anatomical (MR or CT) and functional (PET, SPECT, or optical imaging) information. Alignment of these images requires the use of multimodal registration methods[7]. Among all combinations of modalities, PET-CT and PET MRI are the most mature combinations. However PET-CT has shortcoming due to the significant radiation dose to the small animal contribute by CT and MRI offers better contrast among soft tissues compared to CT [20]. As a result, PET-MRI which offers the combination of high resolution, soft tissue, anatomical information of MRI, and high sensitivity of PET[11] is a promising combination in preclinical research and will certainly progress to clinical application[20]. The common small animal biological studies involving PET and MRI acquisitions are tumour imaging, brain imaging and cardio-vascular imaging [8].

Numerous works has been done to register MRI-PET images. Woods *et al.*[29] developed automatic MRI-PET registration of human head of the same person by minimizing the standard deviation of the PET pixel values that correspond to each MRI pixel value. They suggested to exclude the non-brain structures of MRI images (scalp, skull, meninges) before applying the registration algorithm. Collignon *et al.*[6] used the concept of information theory to develop automatic multi-modality image registration for human head images and the mutual information (MI) of gray value pairs is proposed as new matching criteria. Pluim *et al.*[21] used combined mutual and gradient information to improve registration of multimodality. The algorithm was evaluated for human brain MRI-PET registration task. Ardekani *et al.* [2] presented an automatic algorithm for multimodality image registration that relied on minimizing the K-means variance criterion. Non-rigid registration methods have also been proposed for specific objects outside the head such as registration of abdomen, breast, lung. Mattes *et al.*[17] combined a rigid body deformation with localized cubic B-splines to capture the significant non-rigid motion in the chest between PET and CT images, using the mutual information as a similarity criterion. Rueckert *et al.* [7] used a non-rigid registration algorithm for breast MR images. They model the global motion with an affine transformation and describe the local breast motion with a free-form deformation (FFD) based on B-splines.

However, the methods reviewed above have been mostly applied to human images and later they start to be applied to small animal images. Vaquero *et al.* [27] investigated the MRI-PET registration algorithms developed by Woods *at al.*[28] and Collignon *et al.*[6] to register PET images to CT or MR images of the rat skull and brain. The latter was found to be more robust algorithm than the former method. Hayakawa *et al.* [23] modified the algorithm proposed by [2] to register PET and MR images of rat brains. Bernier *et al.* [4] proposed parallel multi-resolution and PCA initialization for MRI-PET registration of small animal bones. As many of the previous works on small animal MRI-PET registration are focused on the head and bones, i.e a rigid body registration, the non-rigid registration problem remains more open and an active area of research.

In this project, we introduce a new method to the work of Baisa[3]. A two-level hierarchical registration is proposed where in the first level a global affine registration is used to bring the volumes into alignment and then a local B-spline (elastic) registration is performed in the second level. In PET images, only the low intensities reveal the anatomical structure in MRI scan. Consequently, focusing on the PET range of intensity which reveal the anatomic structure in MRI will help to align the images perfectly. We apply a non-linear intensity transformation to the PET volume to enhance the contrast of the low intensity. The computation of initial parameters using PCA and the process of finding the optimal global affine transformation is performed using the intensity transformed PET. Then, the original PET volume is transformed using the final optimal global affine transformation matrix and given to the local registration as input. Moreover, we develop a visualization and GUI support for the registration algorithm so that the user can interact with the registration to select volume of interest and visualize the input/outputs files.

II. Method

The basic components of image registration framework are given in figure 1. The flow-diagram of the implemented hierarchical semi-automatic registration algorithm is shown in figure 2. The details of the block in the flow-diagram are discussed in the following sections.

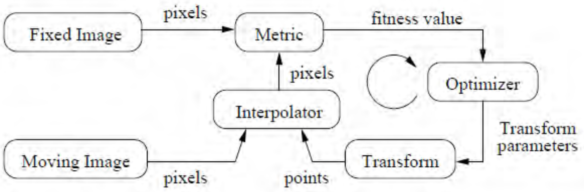


Fig. 1: The basic components of image registration framework. From[9]

A. Volume of interest (VOI) selection

Since the small animals are not cooperative like humans, anaesthesia is used through out the acquisition session to keep the animal in the same position, i.e the imaging session is governed by anaesthesia and it is limited. This limited time is enough to take PET scan of large part of body. In contrast, as different sequences and weighting are considered during the acquisition of MRI, it is difficult to take scan of large part of the body for each types of sequence and weighting. Consequently, the PET volume is always larger than MRI.

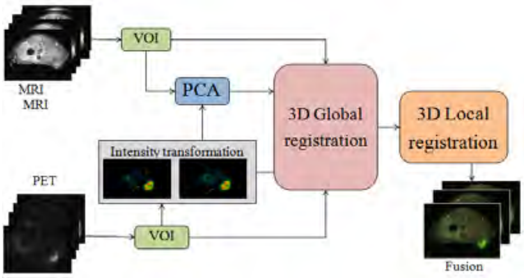


Fig. 2: Flow digram of the proposed algorithm. (VOI=volume of interest)

As the whole body images of small animal contain many articulated joints and the PET volume lacks spatial details, it is difficult to initialize the registration without avoiding the non-overlapping region. Moreover, the PCA is not useful if the two volumes do not refer to the same body. Selecting VOI will help to avoid unwanted objects present in the image volumes from affecting the registration outcome.

For better output and not to miss slices during the selection of VOI, we introduce a method to compare the slice thicknesses in both volume and add appropriate slice to the volume with thinner slice as shown in figure 3.

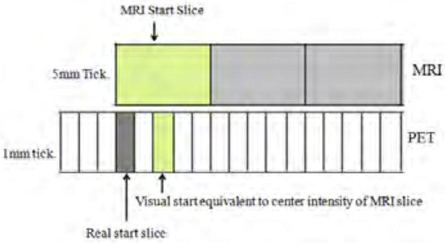


Fig. 3: MRI-PET slice selection offset

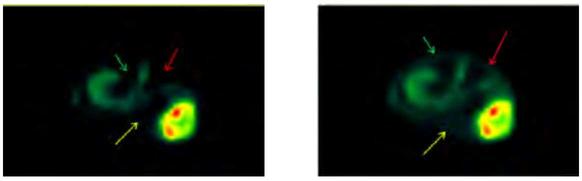
Based on the visualization displayed without knowing the slice thickness, the start slices of the MRI volume corresponds to the PET slice marked in green which corresponds to the center intensity of the MRI slices. However, the green marked PET slice corresponds only to the center part of the MRI slice and the correct starting slice corresponds to the MRI slice should be the one marked with black colour.

B. Intensity transformation of PET

When registration PET image with MRI, it is important to focus on the range of intensities which reveal the anatomic structure in the MRI scan. Normally, only the low intensities in the PET volume represent the full anatomical structure in MRI scan. Increasing the dynamic range of the low intensity will support to the PCA to compute the right centroid and principal axis by maximizing the density of 1s' in the PET volume to comparable level with the density of 1s' in the MRI volume. This can be alleviated by employing a sigmoid function[24]. It is a non-linear mapping which maps a specific range of intensity values into a new intensity range by making a very smooth and continuous transition in the borders of the range. Sigmoid function is given by:

$$I' = (Max - Min)\frac{1}{1 + e^{-(\frac{I-\beta}{\alpha})}} + Min \tag{1}$$

where I is the input intensity and I' is the transformed intensity, Max and Min the maximum and minimum of the expected output image, α defines the width of the input intensity range, and β defines the intensity around which the range is centered[9].


(a) Before

(b) After

Fig. 4: *PET slice before and after intensity transformation.*

C. Principal component analysis (PCA)

PCA is a technique that computes a linear transformation to map a high dimensional space into a lower dimensional space. The basis of PCA is computed by the eigen-decomposition of the data covariance matrix [1]. The idea of PCA initialization is derived from the theory of rigid body where a rigid body is uniquely located by knowledge of its center of mass (centroid) and its orientation (rotation) with respect to its center of mass[1]. PCA produces a single best line in such a way that the sum of the squares of the perpendicular distances from the sample points to the line is a minimum. The first principal component is the variable defined by the line of best fit which indicates the greatest amount of variation whereas the second principal component is the variable defined by the line that is orthogonal with the first and the center of the data set is the intersection of the two axes[14]. We have implemented PCA to find the centroid and orientation of image PET and MRI volumes to initialize the translation and rotation as described by Lu and Chen[14].

D. Global Transformation model

The global transformation model describes the overall motion of the animal body. An affine transformation parametrized by 12 degrees of freedom (DOF) is proposed for the global motion. Since the two modalities have different resolution, the rigid registration with 6 DOF is not sufficient to overcome the global motion. For 3-D images, the affine transformation can be written as:

$$T_G(x,y,z) = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \theta_{14} \\ \theta_{24} \\ \theta_{34} \end{pmatrix} \tag{2}$$

where the coefficients $\theta_{(.)}$ parametrize the 12 DOF of the transformation, i.e 3-DOF rotation (R), 3-DOF transformation, 3-DOF scaling (S) and 3-DOF shearing (H).

The initial rotation and translation are computed using PCA and the initial scaling and shearing are assumed to be identity, i.e initially there is no scaling difference between the volumes and there is no shearing problem.

E. Local Transformation model

The scope of the affine transformation is to align the two volumes globally. However, there is local deformation due to breathing and uncontrolled movement in the lower abdomen of the small animals during acquisition. A non-rigid Cubic B-spline free-form deformation (FFD) is used for the local registration. The motivation to choose Cubic B-splines for the local deformable registration is that, B-splines is the most adequate basis function to represent the deformation with very small overlap which makes it faster and reduce the interdependency between the parameters as demonstrated by Kybic and Unser[12]. In addition, Cubic B-spline have the least number of contributing factions with respect to the other methods like polynomials, radial basis functions, and wavelets[13].

A B-spline based FFD can be written as a 3D tensor product of one-dimensional cubic B-spline, producing a transformation separately for each axes. Let ϕ denote a uniformly spaced grid (lattice) of n_xxn_yxn_z control points $\phi_{i,j,k}$ with spacing of δ where, $-1 \leq i \leq n_x - 1$, $-1 \leq j \leq n_y - 1$,$-1 \leq k \leq n_z - 1$. Then the non-linear transformation for each point (x, y, z) in the volume is computed as[10] :

$$T_{local}(x,y,z) = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3}\beta_l(u)\beta_m(v)\beta_n(w)\phi_{i+l,j+m,k+n} \tag{3}$$

Here $i = \llcorner x/n_x \lrcorner - 1$, $j = \llcorner y/n_y \lrcorner - 1$, $k = \llcorner z/n_z \lrcorner - 1$, denote the index of the control point lattice containing (x, y, z), and u, v and w are relative positions of (x, y, z) in the three dimensions, where $u = x/n_x - \llcorner x/n_x \lrcorner$, $v = \frac{y}{n_y} - \llcorner y/n_y \lrcorner$, $w = z/n_z - \llcorner z/n_z \lrcorner$. $\beta_l, \beta_m, \beta_n$ represents the l^{th}, m^{th} and n^{th} function of the B-spline respectively. A one-dimensional cubic B-spline is given by[7]:

$$\begin{aligned} \beta_0(u) &= \frac{1}{6}(-u^3 + 3u^2 - 3u + 1) \\ \beta_1(u) &= \frac{1}{6}(3u^3 - 6u^2 + u) \\ \beta_2(u) &= \frac{1}{6}(-u^3 + 3u^2 + 3u + 1) \\ \beta_3(u) &= \frac{1}{6}u^3 \end{aligned} \tag{4}$$

F. Interpolation

Interpolation is a method of constructing new data points within the range of a discrete set of known data points. Image volumes are sampled at discrete grid points, P and when the image's grid points are transformed to align with other image, the grid point do not coincide with the other grid points. Hence interpolation must be applied to calculate the intensity values at the new grid points using the information from neighbouring pixel or voxel grid positions. Different interpolation methods are proposed for image registration. Among them, we applied B-spline interpolation because it is the most effective interpolation scheme having the superior performance than any other polynomial basis function of the same order and is highly recommended for multi-resolution registration strategy [22][26]. It produced an interpolated function that is continuous through to the second derivative. For 1-D, the equation of cubic B-spline is given by:

$$\beta^3(x) = \begin{cases} \frac{1}{6}(4 - 6|x|^2 + 3|x|^3) & \text{if} \quad |x| \leq 1 \\ \frac{1}{6}(2 - |x|^3) & \text{if} \quad 1 < |x| \leq 2 \\ 0 & \text{for} \quad |x| > 0; \end{cases} \tag{5}$$

G. Similarity metric

A normalized mutual information (NMI)[25] is used as a similarity metric in both stages of registration. Mutual information is a measure of the amount of information one random variable contains about another. In the context of image registration, image intensity is a random variable and MI measures how much image intensity in one image tells about image intensity in the other image and is defined in terms of entropy[15]. Entropy is a self information of a random variable or a measure of uncertainty of random variable. The mutual information of image X and Y is given by:

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \tag{6}$$

where $H(X)$, $H(Y)$ demote the marginal entropies of X, Y and $H(X,Y)$ denotes their joint entropies. Let $P_X(x)$ and $P_Y(y)$ are probability distribution of intensity values of x and y of image X and Y, then the marginal entropies are given as:

$$H(X) = -\sum_x P_X(x)log(P_X(x)) \tag{7}$$

and

$$H(Y) = -\sum_y P_Y(y)log(P_Y(y)) \tag{8}$$

Similarly the joint entropy $H(X,Y)$ of a pair of random variables (X,Y) with a joint probability density function $P_{XY}(X,Y)$ can also be defined as:

$$H(X,Y) = -\sum_{x,y} P_{XY}(x,y)log(P_{XY}(x,y)) \tag{9}$$

If both images are aligned, the mutual information is maximized. It has been shown by Studholme *et al.*[25] that mutual information is not independent of overlap between two images. To overcome any dependency on the amount of overlap, Studholme *et al.* suggested the use of normalized mutual information (NMI) as measure of image alignment. The NMI is given as:

$$NMI = \frac{H(X) + H(Y)}{H(X,Y)} \tag{10}$$

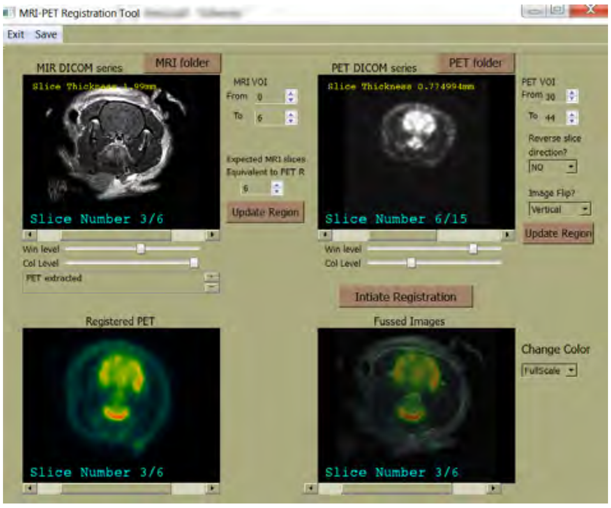
In practice, direct access to the marginal and joint probability densities is not possible and hence the densities must be estimated from the image data. The two most efficient techniques used for probability density estimation are discrete joint histogram and Parzen windowing[9][22]. In case of discrete joint histogram, the marginal and joint probability densities are computing by counting the number of occurrence of each intensity in the images. This method does not allow the similarity metric to be explicitly differentiated and can only be used with non-gradient based optimization methods. Whereas, in Parzen windowing, the marginal and joint densities are estimated by constructing a continuous density function by superimposing kernel functions centered on the intensity samples obtained from the images. Parzen windowing provides a continuous joint histogram which is a derivative function, so that gradient-based optimization method can be applied in the registration process[30]. In this project, mattes mutual information which uses Parzen windowing for estimation of the density distributions implemented in ITK[9] is used.

H. Optimization

Optimization algorithms find the optimal transformation parameters that can align volumes by minimizing the negated mutual information. Optimization methods are categorized into two groups as: derivative and non-derivative. Both methods iteratively select parameters such that the similarity metric is minimized. The strength of derivative optimization methods is that, if the initialization is quite close to the optimum, they converge rapidly and with high precision. The weakness is that they converge to a local minimum if the initialization is far from the optimum[19]. In our case, as both global and local registrations are initialized well, the local convergence is not problem. In addition, using the Parzen windows to estimate the probability density distributions allows to form a derivative continuous function. Hence, a derivative optimization method is used. A regular step gradient-descent, is used to optimize the mutual information of the affine global registration [16][17]. In case of the local registration where the B-spline transform has a high dimension of parameter space, optimization algorithm which handles memory-related problems should be used. Consequently, Limited memory Broyden-Fletcher Goldfarb-Shannon with bounds (LM-BFGS-B)[5] is used. Only a low-rank approximation is computed instead of the entire Hessian matrix during minimization allowing linear or super-linear convergence rates[19].

I. Visualization and GUI

The registration algorithm is developed using Insight Toolkit(ITK)[9]. Though, ITK provides advanced algorithms for performing image registration and segmentation, it does not provide support to perform image visualization, nor does it offer any graphical user interface (GUI) framework. Consequently, the Visualization Toolkit (VTK)[18] which is an open-source, freely available software system for 3D computer graphics, image processing and visualization is integrated with ITK for visualization purpose. The GUI is developed by integration Qt, another cross-platform application framework that is widely used for developing application software with a graphical user interface (GUI). The programs are written using Visual Studio C++. The developed GUI and visualization is given in Figure 5.



MRI VOI From 0 To 6

Expected MRI slices Equivalent to PET R 6

PET VOI From 30 To 44

Reverse slice direction? NO

Image Flip? Vertical

PET extracted

Change Color FullScale

Fig. 5: MRI-PET slice selection offset

III. Experiments and Results

The developed semi-automatic MRI-PET registration algorithm is tested and validated using seven different datasets. The experiment was carried out using processor of Intel(R) Core(TM) i3-2350M CPU@ 2.30GHz 2.30GHz, RAM 4.00GB(2.70 usable) running in Windows 7 32-bit.

A. *Experiment dataset*

The dataset used to investigate the performance of this developed algorithm were obtained from preclinical imaging laboratory at Dijon in the framework of the IMAPPI (Integrated Magnetic resonance And Positron emission tomography in Preclinical Imaging) project and consists of brain and abdominal MRI and PET images of rat and mice. The brain scan dataset used was deformed both globally and locally. On the other hand, the abdominal scan datasets used contain slightly deformed and noise volumes. All the dataset were axial images and their detail size is given in TableI.

Subject	Test	Modality	Dimension	Voxel size(mm)
Abdomen	test1	MRI T1	256x256x7	0.27x0.27x3
		PET	176x176x48	0.39x0.39x0.3875
Brain	test 2	MRI T1	256x256x6	0.2x0.2x2.01
		PET	176x176x12	0.39x0.39x0.7749
Abdomen	test 3	MRI T2	256x256x7	0.12x0.12x3.0
		PET	175x175x23	0.39x0.39x0.775
Abdomen	test 4	MRI T1	256x256x7	0.27x0.27x3
		PET	176x176x25	0.39x0.39x0.775
	test 5	MRI T1	256x256x7	0.27x0.27x3
		PET	176x176x25	0.39x0.39x0.775
	test 6	MRI T2	256x256x7	0.12x0.12x1.5
		PET	175x175x25	0.39x0.39x0.775
Brain	test 7	MRI T1	256x256x16	0.2x0.2x1.99
		PET	176x176x6	0.39x0.39x0.78

TABLE I: Details of the experimental dataset used

B. *Results*

A sample visual registration result of brain and abdominal are given in figure 6 and 7 respectively.

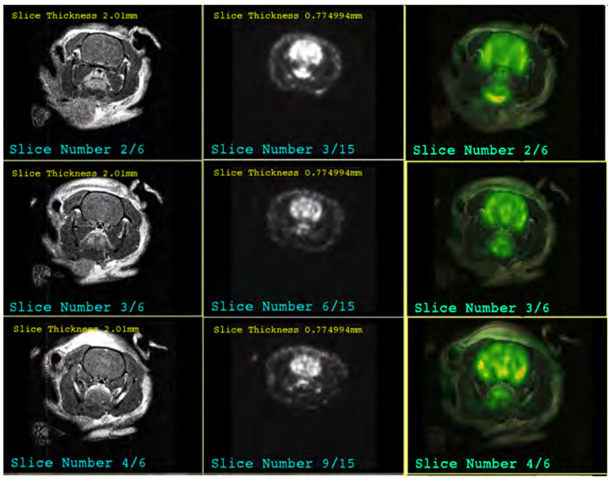


Fig. 6: Registration result of Brain MRI-PET volumes

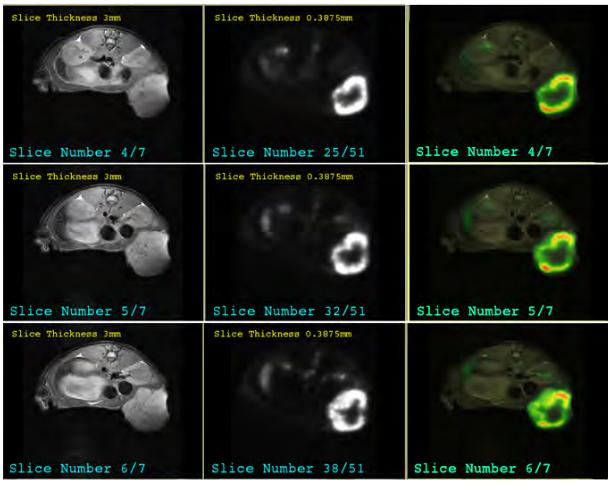


Fig. 7: Registration result of abdominal MRI-PET volumes

C. *validation*

Due to the lack of ground truth and landmarks to perform quantitative validations, the developed semi-automatic MRI-PET registration tool is validated by two experts using the two brain and five abdominal datasets. First, the experts set an evaluation criteria to validate the registration algorithm in terms of precision, accuracy, robustness, stability, reliability of the overall registration algorithm. Both experts set a common ranking ranges from 0 to 5, where 5= perfect, 4=very good, 3= acceptable, 2= Limited, 1= Wrong alignment and 0=completely error. Since, the registration comprises both global and local registrations, the evaluation is performed slice by slice. figure 8 shows the result of the validation.

The maximum score is for abdomen test where it has the a PET volume with the minimum slice thickness of all. As indicated in tableI, the PET slice thickness for test-1 is 0.3875 which is half of the slice thickness of other datasets. The minimum score is for the brain test which is the most difficult of all the other datasets. In this dataset, there is abrupt change or deformation in the MRI scan from one slice to the other which does not exist in the PET slices.

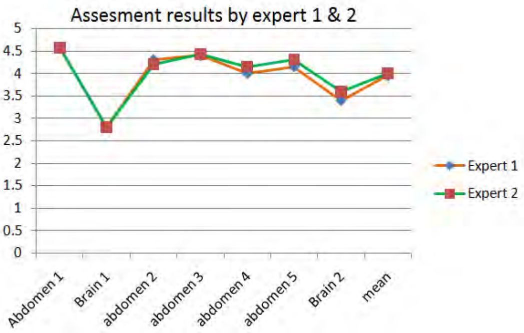


Fig. 8: Visual assessment by experts.

In addition, the MRI slices contains additional information out of the brain including ear and other parts which doesn't exit in the PET. This all affects the performance of the registration.

IV. Discussion

The overall performance of the registration algorithm is very good as indicated by the visual assessment of the experts. In general, the performance of the registration algorithm for abdomen is very promising. The mean of the registration for the abdomens is above 4.3. The mean processing time of the above datasets is 12sec. Moreover, a comparison of with and without intensity transformation is performed and both accuracy and time are improved by 25%.

To conclude, a good semi-automatic registration algorithm is developed. The registration algorithm is supported by dynamic and user friendly visualization and GUI support. The performance of the regurgitation is "Very-Good" as evaluated by experts visually.

As a future work, we recommend to validate the algorithm with additional dataset. In addition, the dataset used above are all axial images and it will be good to test and validate the algorithm for coronal images. Last but not least is to test the algorithm for clinical data with prostate cancer.

References

[1] N. M. Alpert, J. F. Bradshaw, D. Kennedy, and J. A. Correia. The Principal Axes Transformation A Method for Image Registration. *Journal of Nuclear Medicine*, 31(10):1717–1722, 1990.

[2] B. Ardekani, M. Braun, B. Hutton, I. Kanno, and H. Lida. A fully automatic multimodality image registration algorithm. *Journal of Computer Assisted Tomography*, 19(4):615–623, 1995.

[3] N.L. Baisa. MRI-PET Registration With Automated Algorithm in Pre-clinical Studies, VIBOT Thesis, 2013.

[4] M. Bernier, R. Lepage, L. Lecomte, L. Tremblay, L. Dor-Savard, and M. Descoteaux. Free-Form B-spline Deformation Model for Groupwise Registration. *Conference Proceeding International Society of Magnetic Resonance in Medicine(ISMRM)*, page 3255, 2011.

[5] R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[6] A. Collignon, F. Maes, D. Delaere, D Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. *Information Processing in Medical Imaging*, pages 263–274, 1995.

[7] B. Dogdas. Image registration with applications to multimodal small animal imaging, PhD Thesis. *University of Southern California*, 2007.

[8] F. Brunotte H. Haas, B. Collin, PharmD, A. Oudot, S. Bricq, A. Lalande, X. Tizon, Vrigneaud, P.M. Walker. Integrated PET/MRI in preclinical studies State of the art. *tijdschrift voor nucleaire geneeskunde*, 35(4):1144–1152, 2013.

[9] H.J. Johnson, M. McCormick, and L. Ibanez. The ITK Software Guide. Third Edition Updated for ITK version 4.5. 2013.

[10] J.V. Hajnal, D.L.G. Hill, and D. J. Hawkes. *Medical image registration*. CRC Press LLC, 2001.

[11] C. Kagadis, G. Loudos, K. Katsanos, S. Langer, and G. Nikiforidis. In vivo small animal imaging: Current status and future prospects. *Medical Physics*, 37(12):6421, 2010.

[12] J. Kybic and M. Unser. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, (11):1879–2890, January.

[13] B. Likar and F. Pernuš. A hierarchical approach to elastic registration based on mutual information. *Image and Vision Computing*, 19:33–44, 2001.

[14] Z. Lu and W. Chen. Fast and Robust 3-D Image Registration Algorithm Based on Principal Component Analysis. *Bioinformatics and Biomedical Engineering, 2007. The 1st International Conference*, pages 872–875, 2007.

[15] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997.

[16] D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellyn, and W. Eubank. Nonrigid multimodality image registration. *SPIE Medical Imaging*, pages 1609–1620, July 2001.

[17] D. Mattes, D.R. Haynor, H. Vesselle, T. Lewellen, and W. Eubank. PET-CT image registration in the chest using free-form deformations. *IEEE transactions on Medical Imaging*, 22(1):120–128, January 2003.

[18] J.C. Moore. Visualizing with VTK. *Linux Journal*, 20:93–100, 1998.

[19] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[20] B. Pichler, H. Wehrl, A. Kolb, and M.S. Judenhofer. PET / MRI : The Next Generation of Multimodality. *Seminar in Nuclear Medicine*, 38(3):199–208, 2008.

[21] J. P. Pluim, J. B. Maintz, and M. Viergever. *IEEE transactions on Medical Imaging*, (8):809–814, August.

[22] J. P. W. Pluim, J. B. A. Maintz, and M. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE transactions on Medical Imaging*, 22(8):986–1004, August 2003.

[23] M. Preuss, P. Werner, H. Barthel, U. Nestler, F. Wolfgang Hirsch, D. Fritzsch, M. Bernhard, and O. Sabri. A PET-MRI registration technique for PET studies of the rat brain. *Nuclear Medicine and Biology*, 27(2):121 – 125, 2000.

[24] Saruchi. Adaptive Sigmoid Function to Enhance Low Contrast Images. *International Journal of Computer Applications*, 55(4):45–49, 2012.

[25] C. Studholme, D. L. Hill, and D J Hawkes. Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimization of voxel similarity measures. *Medical Physics*, 24(1):25–35, January 1997.

[26] P. Thévenaz, T. Blu, and M. Unser. Handbook of medical imaging. chapter Image Interpolation and Resampling, pages 393–420. Academic Press, Inc., Orlando, FL, USA, 2000.

[27] J. Vaquero and M. Desco. PET, CT, and MR image registration of the rat brain and skull. *IEEE Transactions on Nuclear Science*, 48(4):1440–1445, 2001.

[28] R. Woods, S. Cherry, and J. Mazziotta. *Journal of Computer and Tomography*, 16(4):620–635.

[29] R. P. Woods, J. C. Mazziotta, and S. R. Cherry. MRI-PET registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17(4):536–546, 1993.

[30] R. Xu, Y. Chen, S. Tang, S. Morikawa, and Y. Kurumi. Parzen-window based normalized mutual information for medical image registration. *IEICE Transactions on Information and Systems*, E91.D:132–144, 2010.

[31] R. Yao, R. Lecomte, and E. Crawford. Small-animal PET: what is it, and why do we need it? *Journal of Nuclear Medicine Technology*, 40(3):157–165, September 2012.

Video Mosaicing using Visual Odometry from Stereovision

Yuan Huang Lee
Ocean Systems Lab
Heriot-Watt University
EH14 4AS Edinburgh
Email: yl433@hw.ac.uk

Yvan Petillot
Ocean Systems Lab
Heriot-Watt University
EH14 4AS Edinburgh
Email: y.r.petillot@hw.ac.uk

***Abstract*—Reconstruction of 3D points from a single pair of stereo image is a much researched topic with numerous accurate implementations when well calibrated. The reconstruction of a scene in 3D from a recorded video has more practical usage and is seen to be a greater challenge. Enhancement on the estimated camera pose will lead to a higher accuracy in the scene reconstruction. An open source visual odometry algorithm (LIBVISO)[1] is used as the main framework to obtain an improved estimate of the camera pose. The framework is chosen as it is implemented for stereo setup, it is real time, has practical accuracy, and it is open source thus providing a good skeleton for experimentation.**

In this paper, a range of descriptors and detectors are tested on the original algorithm and analyzed to compare the accuracy and speed. Loop closure methods are then integrated into the original framework and optimized using the g2o open source framework[2]. The improvements are noted using a visual odometry benchmark, and by evaluating the performance on a simulated and a real underwater video sequence.

I. Introduction

Video mosaicing is defined as the procedure to combine the views of the camera at different time instances into a common scene and often involves stitching images from the video frames into a larger visual representation of the area. The use of a two-camera setup, commonly known as a stereo setup provides depth information to the captured view, and thus, a 3D video mosaicing problem can be formulated. The 3D data is then used to obtain a metric measurement for the movement of the camera and this is called visual odometery. Visual odometry can be briefly defined as the use of visual information to estimate the change of position over time. In the paper, the approach for video mosaicing will be to use the obtained odometry to form the 3D scene by projecting the cloud of points from the estimated position. Therefore accuracy of the visual odometry obtained and the resulting reconstructed scene will be the main focus of the paper.

Video mosaicing is applied in the 2D scene for example to create image mosaics of text documents, to create a huge map from aerial photography or to document a coral reef area underwater[3]. The frames of the video are stitched by using mutual features to match the subsequent frames. Mosaicing is performed by using the features to estimate the pose of the image and this can be done incrementally[1] or globally[4] after the whole sequence has been processed. Video mosaicing in 3D is an extrusion of the 2D application mentioned, with the added benefit of representing the surveyed area in 3D. The significance of a proper 3D reconstruction from a video sequence is seen in multiple fields whether above water or for underwater uses. The dimensions or volume of an object of interest such as a coral reef for environmental protection, an underwater wreckage or buildings can be computed if needed and a visualization of a seabed terrain can help in making geological prediction for mining minerals such as diamonds or oil. The many uses and versatility of reconstructing a 3D scene from a stereo recorded video, provide an alternative to the use of 3D laser scans (above water) or sonar (underwater).

Feature detection and descriptors are key elements of video mosaicing since the camera undergoes substantial change in motion. Introducing changes in rotation, scaling or lighting will cause less robust feature descriptors to fail, and such motion are more common underwater compared to the movements of a car along a road. Thus, the use of different features and detectors are compared for both above water and underwater sequences. Visual odometry suffers from the same drawback as the wheel odometry of a robot which is drift in the motion estimation. As time passes, the small errors at each time step of the camera motion will accumulate and to correct such errors, a graph optimization technique from detected loop closures will be used. If the loop closure detection is a part of the front-end algorithm, the back-end which consists of the optimization process should optimally be able to handle possible outliers in the loop closure detection. The work done in the thesis tries to augment an existing visual odometry algorithm for the purpose of obtaining a proper 3D mosaic from video.

II. Related Work

Video mosaicing had been an active area of research used to create a visualization of small to large scenes. Sato[5] implemented a video mosaicing approach with the goal of obtaining high resolution mosaics of documents and photos from a video using a hand-held camera. The approach is tailored for a planar object and image features are tracked from a frame to another using template matching with Harris corner features. Marzotto[6] uses video mosaicing to perform global alignment from the video frames to generate a super resolution mosaic by using a graph based technique. The Kanade-Lucas-Tomasi

(KLT) tracker is used to compute the inter frame homography, and the graph is constructed using vertices as frames and edges to link overlapping frames. Bundle adjustment is then used to find the best homography to minimize the total misalignment in the mosaic by minimizing a cost function which is a common strategy as seen in [5]. Video mosaicing is made more challenging by trying to mosaic a larger area in an underwater environment in 3D as done by Pizarro[7] using a single camera with the aid of onboard navigational sensors. A local-to-global approach is used whereby 3D submaps is obtained separately and then registered in a global frame for bundle adjustment, using Harris corners as features and RANSAC to estimate the essential matrix. On the other hand, a stereo rig is used by a group of researchers in Girona[4] and the approach uses global alignment to minimize the error cost between correspondences and the distance between fiducial points.

A. Visual Odometry

Different setups have been used to obtain the movement of a camera such as the use of a monocular camera[8], a stereo setup[1] or the use of a kinect device[9]. The odometry in the paper by Civera[8] is the result of solving the SLAM problem using the Extended Kalman Filter(EKF). The algorithm is capable of performing loop closure and a rough estimate of the camera motion can be observed however the metric accuracy would be questionable since a single camera setup will have less accuracy compared to the use of a stereo rig. Geiger provided an open source visual odometry framework in his paper which is the LIBVISO library[1]. The author uses a combination of blob and corner detectors and descriptors computed from the Sobel filter response of the frames and the features are matched circularly. The method does not specify any loop closure method but it is capable of processing all the frames in real time and produces visual odometry with minimal drift even for longer distances. The method used by Geiger is further refined by Bellavia et al.[10] with their method named Selective SLAM (SSLAM). A similar loop chain matching method like in LIBVISO is used however the processed frames are selectively chosen to discard frames with similar visual content to reduce the propagation of error in pose estimation. The resulting approach manages to produce results with less drift especially in the reduction of rotational error and the accumulation of error with increasing distance is less pronounced compared to [1]. Another visual odometry algorithm implementation by Badino [11] shows further improvement by using the whole history of tracked features with minimal additional computation. An augmented feature set is created which consist of sample mean from previously measured features transformed to the current frame. The optimization will minimize the usual feature correspondence as seen in the papers [10][1] however improving it with an additional minimization between the feature correspondence for the current features and the augmented feature set. Badino also performed tests using few detectors and descriptors, KLT, SURF and a combination of Harris and FREAK, concluding that Harris and FREAK returns the best result.

Although the use of stereo rig is the main focus, it is interesting to briefly review the paper by Huang et al.[9] which performs visual odometry and mapping using an RGB-D camera. FAST feature detector is used on different Gaussian pyramid levels and the descriptors is a simple normalized 9x9 pixel patch. Instead of using RANSAC like the previous approach, a graph of consistent feature matches is computed and the maximal clique in the graph is approximated and results in reduced error. The implementation of this framework is also available online (known as FOVIS) however instead of using this framework, LIBVISO is chosen based on the findings in the publication by Wirth [12] which compares FOVIS with LIBVISO. Both framework returns comparable results in term of accuracy but the authors mentioned that LIBVISO is preferable in real underwater marine environment which is our targeted scenario. LIBVISO is also very much suited for sequential stereo video frames where the transition from one frame to another follows a trajectory unlike PVMS2[13] where the image sequence are sparse and non sequential.

B. Feature Detectors and Descriptors

The choice of feature detectors and descriptors may increase the robustness and the accuracy of the visual odometry by eliminating false matches in the visual odometry feature correspondence problem. [1]uses features which performs well in cases where the motion between frames is small however the corner and blob features together with the Sobel filter responses from the image is not invariant to changes such as scale and rotation. A few features which are commonly used and relevant to be tested are Harris corner detector[14], FAST detector[15], SURF features[16], SIFT features[17], and newer features such as ORB features[18], BRIEF[19], BRISK[20] and FREAK[21]. These newer features are binary features which benefit from speed up during the matching process and a more compact memory requirement.

FREAK uses a retina inspired keypoint descriptor and is shown[21] to be better performing compared to SURF, SIFT and BRISK. Corner features are shown to be highly invariant to changes in rotation but not to scale[22]. Scale invariance is often introduced on the keypoints by extracting the feature from multiple scale, such as performing FAST on different levels of a Gaussian pyramid. AGAST[23] is noted to be the detector used in both BRISK and FREAK testing by their authors, while ORB uses FAST detector with the orientation computed.

Attributes	Harris	FAST	SURF	SIFT	ORB	BRIEF	BRISK	FREAK
Scale inv.	No	No	Yes	Yes	Partial	No	Yes	Yes
Rotation inv.	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Type	Corner	Corner	Blob	Blob	-	-	-	-

Table I
COMPARING DETECTOR (IN BLUE), DESCRIPTORS(IN RED) AND BOTH(IN BLACK)

C. Loop Closure and Graph Optimization

A simple but expensive way to perform loop closure would be to search for matches between the current frame and all past frames in the sequence. More efficient strategies can be used to identify loop closure based on keyframe selection and selecting a subset of the keyframes (selected frames) to perform matching. Huang et al.[9] uses a simple approach to add keyframe, which is adding keyframes after a set distance or rotation from the previous keyframe has occurred. The possible matches are limited by taking only frames with similar pose to the current frame and further filtered for matching by a bag of visual words model to determine likely candidates.

Another approach is to have new keyframes when the frames can no longer be aligned with the previous keyframe due to low number of features (RANSAC inliers)[24]. An entropy approach has also been used to obtain the subset of keyframes, the idea being that the entropy increases as the frames are further apart[25]. For loop closure detection, a whole image scene descriptor image has been used[26] using the BRIEF descriptor which does not require a training stage unlike vocabulary-based method[27] which can be used for place recognition. The loop closures are detected by getting the distance between the image descriptors.

The use of these methods are sometimes not wholly accurate and there will be false positives due to incorrect loop closure detection. The graph optimization framework should be capable of handling such outliers which can potentially distort the resulting optimization. The g2o framework [2] is augmented with robust methods that ensure that inconsistent loop closure are disregarded by the algorithm. For example [28] integrated dynamic covariance scaling as a robust function to reject outliers while not compromising the convergence speed in the g2o framework. The optimization framework also offers different robust kernel functions such as the Huber and the Cauchy error function. Another approach for robust optimization is the switchable constraints method by Sunderhauf et al.[29] which works by controlling the loop closure constraints with a switch variable. The switch variable is included as part of the optimization problem therefore the optimization now searches too for the optimal graph topology since some edges can be 'disabled'.

III. Methodology

In the LIBVISO framework the motion between frames are computed following the steps mentioned in their paper[1]. Four steps are mentioned in the paper to achieve 3D reconstruction which are feature matching, egomotion estimation, stereo matching and 3D reconstruction. The LIBVISO framework is used for feature matching and egomotion estimation while for stereo matching and 3D reconstruction, we will use the block matching method[30] to obtain the disparity image which is then converted to 3D point clouds. The feature matching method is substituted with different detectors and descriptors for testing while a loop closure method is added after the egomotion estimation step. Stereo matching and 3D reconstruction is performed on the sequence of stereo image and the point clouds are concatenated in the same coordinate frame.

A. Feature Matching and Egomotion Estimation

Feature matching is performed in a circular way on four images (the pair of stereo images from the previous and current time) as in Fig. 2. A match is added when the same point is reached after matching through a circle (P4_1 == P1). To reduce the number of matches, we added a ratio between the cost of the best match and the 2nd best match to reject point matches that do not satisfy the criteria (i.e. $\frac{bestcost}{2ndbestcost} < 0.8$). A range of detectors are integrated to be part of possible choices in the framework. The detectors are readily available in the OpenCV library[31] and in our experimentation, the following detectors are used, Harris corner detector, SURF (Speeded Up Robust Features) detector, SIFT (Scale Invariant Feature Transform), MSER (Maximally Stable Extremal Region), CenSurE detector, FAST detector, ORB features (which are in fact FAST features computed in pyramid with orientation) and the original feature detectors in the LIBVISO framework (corners and blobs). In a few test cases, the blob type and corner type detectors are combined, just as in the LIBVISO framework, therefore having a combination of Harris features (corner) and SURF features(blob). The detectors are then coupled with the following descriptors, BRIEF, ORB, FREAK, and BRISK descriptors. The descriptors can be classified according to their length of 32 byte (ORB and the original Sobel descriptors in LIBVISO) or 64 byte (BRIEF, ORB, FREAK). The framework has been extended to accommodate 64 byte length descriptors while still maintaining the use of the processor optimized SSE instructions. The chart in 1 shows the decomposition of the different combinations tested.

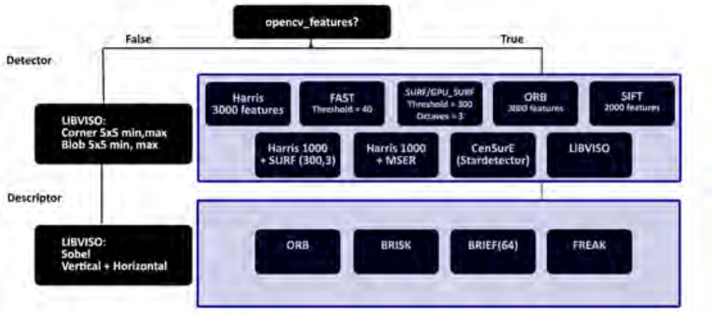


Figure 1. Augmented with OpenCV features for testing

As mentioned in [1] the matched features between the left image in the previous (FeaturesP) and current time frame (FeaturesC) will then be used for the motion estimation step. FeaturesP will be projected from 2D to 3D using the camera calibration parameters and then using 1 the 3D of FeaturesP will be projected to the current frame in 2D.

$$\hat{p}_{t-1} = \pi(X_{t-1}; R, t) = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = [Intrinsic_{3x3}][R \,|\, t] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

The rotation and translation will be estimated by minimizing the difference between the 2D points of FeatureC and the re-projected 2D points of FeaturesP as shown in 2. 3 points are randomly sampled and the optimization is performed iteratively to obtain the highest number of inliers, and finally the estimated frame-to-frame motion parameters is computed from the inliers.

$$\min_{R,t} \sum_{i=1}^{N} \left(\left\| p^i_{t,left} - \pi_{left}(X_{t-1}; R, t) \right\|^2 + \left\| p^i_{t,right} - \pi_{right}(X_{t-1}; R, t) \right\|^2 \right) \quad (2)$$

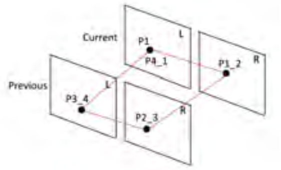


Figure 2. Circular matching of features

B. Augmented Framework - Loop Closure and Graph Optimization

The framework is augmented with loop closure detection and the visual odometry is optimized using the g2o library. The method used for loop closure detection in the augmented framework relies on the pre-existing detected features to minimize recomputation of the features and descriptors, while at the same time, reusing the matching strategy and egomotion estimation for the selection of keyframes. When the motion update is successfully computed between the previous and current frame, the keyframe selection is performed by matching the current frame with the previous keyframe. If the number of inliers and the ratio defined by $number\_of\_inliers/number\_of\_matches$ falls below a certain threshold, the current frame is saved into a vector containing all the other keyframes, and the index is placed in a circular buffer, to ensure that the memory allocated to the frame is not cleared.

When the keyframe is saved, the neighbouring frames from $t-1$ and $t-2$ will also be stored to obtain a better match later in the loop closure. However if the current frame has good matches with the most recent keyframe an edge can be constructed between the two frames thus adding an additional motion constraint even though loop closure does not occur. Loop closure matching and detection is not performed for all frames but only when a keyframe is selected to reduce computational effort. After the keyframe has been added, the loop closure detection starts by sorting and saving all previous keyframes that satisfy a few criteria which are:

- The possible matches are filtered by the search radius. Only keyframes within a specific radius of the newly saved keyframe will be attempted to be matched. The radius is adaptive and incrementally increasing as the distance traveled by the camera increases. This is based on the assumption that the drift increases with distance.
- The previous 10 (user specific) keyframes before the current keyframe are not included in the search radius. Usually loop closure happens between frames that are captured further apart in time.

Matching is performed with the reduced set of frames and a limit is set so that only a maximum number of frames is considered (e.g.. 25 frames). To prevent all frames to be from the same region, the frames are placed into multiple clusters (e.g.. 6) and the clusters are made to be of a certain distance apart. This is useful to increase the probability of loop closure especially when the search radius have increased after a long distance travel. Without the clustering, the frames selected based on their distance are sometimes the nearby frames from the nearest time (e.g. the previous 11th key frame and before). The frame with the highest number of matches is used as the best frame to perform the motion estimation. It will be costly to perform the egomotion estimation on each of the probable keyframes, however simply performing the circular matching and getting the number of matches as a scoring metric is a more efficient method. After the best keyframe is selected, the matching is performed on the neighbours of the keyframe from $t-1$ and $t-2$ and the highest number of matches is taken again as the best frame. This allows the loop closure detection to be performed on a refined scale thus matching with frames that are closer. The motion will be estimated between the loop closure frames and the information matrix added to the graph for optimization.

To perform graph optimization, the projection matrix from the motion estimation will be converted to the Vertex3D format for the g2o optimization. The rotation matrix will be converted to normalized quaternions and the XYZ position is obtained directly from the translation vector. The vertices are constructed using the concatenated poses, while the edges between the vertices are computed from the incremental pose obtained from the egomotion estimation between two frames. The information matrix which is the inverse of the covariance for the edge is constructed by dynamically changing it depending on the distance and the inlier ratio. The assumption made is that, a higher inlier ratio, denotes a smaller uncertainty in the edges.

```
Variance(x,y,z) = 0.02 + sc*0.02+(dist(x,y,z))*0.015;
rot_variance = 0.02 + 0.005*xyz + sc*0.02;
```

The scale is changed by the ratio of the inlier count and total matches. Thus a lower ratio will give rise to a higher scale and the uncertainty will increase.

```
sc = 5 - log(10*ratio+0.0001) - 2*ratio;
```

C. 3D reconstruction on ROS[32]

The augmented framework (changes to the LIBVISO library) is run on ROS using the viso2_ros wrapper and the stereo_image_proc package is used to rectify the images. The augmented framework will run through all the frames in the recorded video and compute the vertices of the graph and the edges, including loop closures. The constructed graph will be optimized using the g2o library and the optimized pose will be used to project the 3D point cloud in the environment. A node pointcloud_odometry will compute the 3D points from the rectified images and publish the point clouds together with the associated transform read from the g2o file. The point clouds will then be concatenated by being transformed to a common coordinate frame and the pcl viewer will be used for the visualization of the point clouds. The computation of the point cloud from the stereo image will be performed by computing the disparity using the block matching function in Open Cv and converting the disparity to 3D points. The process is illustrated in Fig. 3

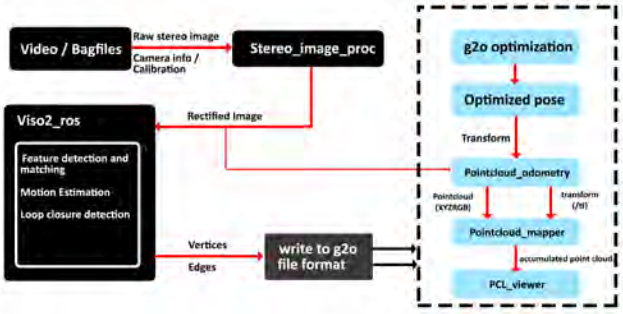


Figure 3. Post-Optimization Visual odometry for 3D reconstruction (mosaicing)

IV. Results

A. Visual Odometry

The improvement or degradation in visual odometry is evaluated using the KITTI Vision Benchmark Suite[33]. Rotation and translation will be evaluated as separate measures. Instead of using the error of the end point of the trajectory, the benchmark utilizes the average of all relative relations at a fixed distance. Therefore the error will be calculated for every specified distance in the trajectory, summed and averaged. The benchmark evaluation returns the percentage of the translation error and the rotation error (deg/m) as a function of path length and velocity. The datasets used in the evaluation is the dataset 00 to 10 which comes with the ground truth for computing the error metric. The table shows the combination of detectors and descriptors used on the LIBVISO framework and the resulting error in translation and rotation. The odometry evaluation returns the rotation error in rad/m and shall be converted to deg/m. The parameters used for the Open Cv features are displayed in 1 and the results are shown in the table below. The results for the error metric after loop closure is also shown. The loop closure optimization is done with 30 iterations using the Gauss-Newton method and for the robust kernel, a Cauchy with kernel width of 0.1 to 0.5 is used.[1].

Detector + Descriptor	No Loop Closure		With Loop Closure	
	Translation Error (%)	Rot. Error (deg/m)	Translation Error (%)	Rot. Error (deg/m)
FAST+FREAK	3.1705	0.015	2.1969	0.0107
Harris + FREAK	3.1212	0.0151	2.2478	0.0113
Harris + BRISK	3.2234	0.0156	2.4268	0.0124
Harris, SURF+ BRIEF64	2.8192	0.0124	2.1311	0.0099
Harris, SURF+ FREAK	2.7190	0.0122	2.0779	0.0097
LIBVISO+ FREAK	2.4476	0.0109	1.7618	0.0081
ORB	3.1705	0.0150	2.2047	0.0108
MSER, Harris + BRIEF64	3.0754	0.0149	2.4475	0.0123
LIBVISO	2.6293	0.0117	1.8177	0.0084
SIFT + FREAK	2.9923	0.0137	2.2768	0.0111
CenSurE + BRIEF64	3.4539	0.0159	2.6705	0.0131
SURF + FREAK	2.5247	0.0104	1.8837	0.0077

Table II
Comparison of visual odometry error among different detectors and descriptors with and without loop closure

After sorting the best performing algorithm we can summarize that the change of descriptors does indeed improve the accuracy of the visual odometry. The use of FREAK descriptors improves upon the accuracy in comparison to the original sobel descriptor, ORB descriptor, BRIEF and BRISK. In terms of feature detectors, we can summarize that the use of LIBVISO features, and the use of SURF features for detection is capable of providing better result compared to using Harris corner points, MSER, SIFT, FAST or CenSurE. The top three combination of detectors and descriptors are listed below according to translation and rotation errors. In the course of the experiment it should be noted that certain features such as MSER is tried independently but due to the lower number of features, the motion estimation tends to fail at times. However, even though the features computed by SURF is usually around 1000, the features are reliable enough to compare with denser feature choices such as 3000 Harris features or the LIBVISO features which usually amounts to around 4000 features.

- **Translation:** LIBVISO+FREAK **>** SURF + FREAK > LIBVISO
- **Rotation:** SURF+FREAK **>** LIBVISO + FREAK > LIBVISO

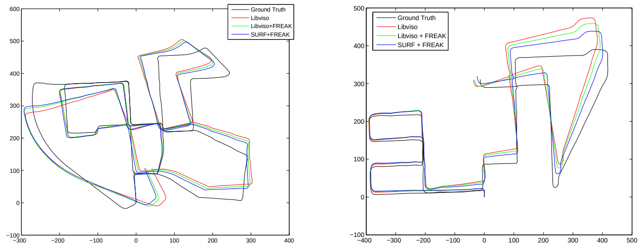


Figure 4. For long distance sequence 00 and sequence 08 in KITTI dataset. Visual comparison of trajectory using MATLAB plot

Fig. 4 shows the trajectory plot of two chosen sequence for visual odometry performance by varying the detector and descriptor. SURF + FREAK provides the best representation

[1] LIBVISO in the table and figures refers to the use of the original features (corner/blobs) and detectors (Sobel), while LIBVISO + FREAK means that the LIBVISO features (corner/blobs) are used but FREAK is used as the descriptor instead of the Sobel descriptors originally.

in the 2D plot and this can be seen in the resulting table where it has the lowest rotational error. As for the translation error although in the trajectory, SURF + FREAK looks to be the best, we can assume that this is not reflected in the result due to the fact that the trajectory plot is a 2D plot while the benchmark utilizes 3D positions. Drift might be greater for SURF + FREAK in the un-plotted axis direction.

As seen in II and Fig. 5, the loop closure detection improves result for all combination of detector and descriptors. Note that not all the sequence in the datasets have loop closure since in some case, the known areas are not revisited. In the tests done, the parameters used are uniform for all combinations however more loop closures can be detected for some detectors if the parameters had been tuned. Conclusively, even with loop closure integrated, good results are obtained using the original LIBVISO, LIBVISO + FREAK and the SURF + FREAK combination just as in the tests without loop closure. LIBVISO + FREAK manages to produce the best result for translation and SURF + FREAK continues to top the comparison in the rotation measure.

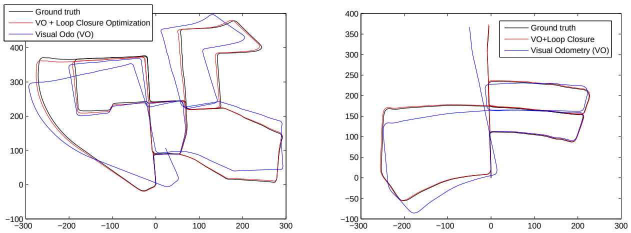


Figure 5. Trajectory before and after loop closure optimization for sequence 00 and sequence 02.

B. 3D Video Mosaicing and Reconstruction

In this section, the result for the process illustrated in 3 will be presented and evaluated using two datasets. The first dataset is a short sequence from the underwater simulator (UWSIM) and the second sequence is a more complex dataset from the TRIDENT project. A naive metric evaluation is used for the simulated sequence and the 3D generated mosaic will be evaluated visually for both dataset. In the first dataset (Dataset1-UWSIM), the simulated underwater vehicle is moving in a boxed trajectory 4m x 3m, from one corner of the square to the other. The stereo camera of the vehicle is facing downwards to the bottom of the pool which is superimposed with a flat image mosaic, a few amphoras and red coloured boxes. The vehicle returns to the original position and the end of the sequence so the ideal case would be 0 shift between the position of the vehicle at frame 0 and the last frame. The naive metric used here would be simply the distance between the two frames. Besides, the resulting 3D mosaic for three different configurations are shown, mainly the best three combinations summarized in the visual odometry benchmarking in the previous section and the resulting 3D mosaic after optimization for loop closure.

Setup	Translation error (m)	Translation error (After LC) (m)
Original Features	0.0575	0.0057
LIBVISO + FREAK	0.0250	0.0036
SURF + FREAK	0.0207	0.0029

Table III
TRANSLATION ERROR BETWEEN DIFFERENT ALGORITHM SETUP WITH AND WITHOUT LOOP CLOSURE (LC)

From the image mosaic in Fig. 6 since the sequence is short, the drift is minimal but accumulative, therefore the difference in the translation can be seen at the end of the sequence. Among the three configurations, SURF+FREAK and LIBVISO + FREAK performs comparably better than the original setup. The reconstruction is further refined after optimizing the graph of poses. The difference in depth can be observed in Fig. 7 between the mosaics before and after optimization and it can be observed that the optimized mosaic fits the environment where the floor of the pool is flat (single depth layer).

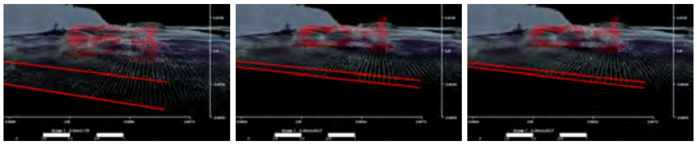


Figure 6. Point clouds at the start point for different setup. From left to right, Original, Libviso + FREAK, and SURF + FREAK. The different reconstructed layers are highlighted in red.

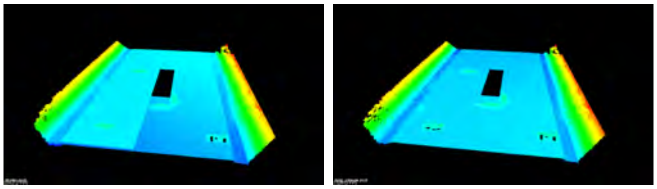


Figure 7. A depth representation of the whole mosaic (Dataset1-UWSIM) before (left) and after (right) optimization.

For the second dataset (Dataset2-TRIDENT), the bag files used for testing are obtained from a real world recording of an underwater scene. The 3D reconstruction of the TRIDENT sequence is shown using the depth representation by the pcl viewer. For the reconstructed scene, when there are sharp edges to the color transition, it can be interpreted as an incorrect re projection of the scene since the bottom of the pool is flat. Sharp edges denotes a strong difference in depths. The reconstructed scenes and the point clouds of the reconstruction after loop closure will be shown in Fig. 8.

After optimization, a comparison between the three methods shows that a proper reconstruction is obtained using the FREAK descriptors due to the larger number of loop closure sections. Loop closure detection and optimization of the graph fuses the different layers of the pool bottom mosaic to a single coherent mosaic. However it should also be noted that the mosaic still has inaccuracies in the re projection.

Video Link: 3D reconstruction from both dataset.

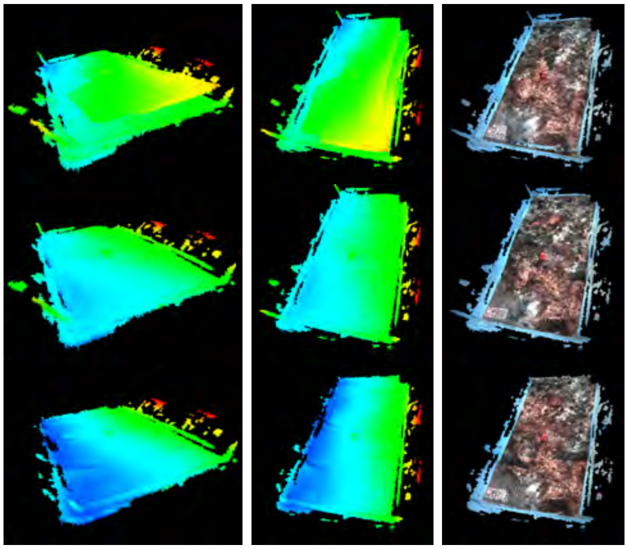


Figure 8. Depth and RGB point cloud view after optimization for LIBVISO, LIBVISO+FREAK, SURF+FREAK from top to bottom row (Dataset2-TRIDENT)

V. Conclusions

From the comparison of the detectors and descriptors, it can be concluded that the FREAK descriptor provides the best association between the frames in the video in comparison to many others that were surveyed, namely the sobel descriptor, ORB, BRIEF, and BRISK. The FREAK descriptor is not only better but also requires less computational time compared to BRISK or other time consuming descriptors such as SIFT. On the other hand, for the choice of feature detector, the features used in LIBVISO are actually one of the best among all that were tested for the KITTI sequences[33]. The SURF feature detector can be considered as another alternative to the corner and blob detectors used in LIBVISO. Although the SURF detector requires higher computation cost, the use of the GPU_SURF function using a compatible graphic card is able to greatly reduce the required time as long as the graphic card has available computational memory and resources.

In the paper, a loop detection method is also formulated, partly inspired by other works but with the addition of several extra features such as clustering and searching among the neighbours. In the KITTI benchmark, there were no noticeable loop detection errors and this is because images in the benchmark which are successfully paired for loop closures are quite similar when the vehicle travels through the same road. However for the underwater video sequence where the motion of the robot has a greater degree of freedom, the loop closure detection will fail when the non-robust features such as the Sobel descriptors are used. Loop closure detections are more reliable with an rotation invariant feature such as FREAK. With the additional optimization afforded by the loop closure detection, the visual odometry especially for sequences where loop closures are evident, will have the accuracy improved greatly both in the visual odometry and in the reconstruction. The sequences with loop closure detection produces optimized visual odometry which closely approximates the ground truth as shown in 3. The 3D reconstruction after optimizing the pose graphs are shown to be good enough in the first simulated dataset and would have been better for the TRIDENT dataset if the robot motion is controlled autonomously in a smoother fashion. However, the resulting reconstruction is still good enough to visualize the whole scene with a few inaccurate re projections which can be corrected with post processing on the point clouds to perform merging. Although the choice of feature detector is less important than the choice of descriptors, it is worth noting that the use of SURF detectors has shown better performance when the matching radius is limited even though the translation between frames are larger (capable of coping with large drifts). The matching radius has to increased for the simulated dataset when the original features are used.

Acknowlegment

The LIBVISO framework is mainly used and the ROS point-cloud tools provided by Systems, Robotics & Vision Group, University of the Balearic Islands has been extremely useful to visualize the point cloud data for the 3D reconstruction.

References

[1] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 963–968.

[2] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization." in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

[3] D. Lirman, N. R. Gracias, B. E. Gintert, A. C. R. Gleason, R. P. Reid, S. Negahdaripour, and P. Kramer, "Development and application of a video-mosaic survey technology to document the status of coral reef communities," *Environmental monitoring and assessment*, vol. 125, no. 1-3, pp. 59–73, 2007.

[4] N. Gracias, P. Ridao, R. Garcia, J. Escartin, M. L'Hour, F. Cibecchini, R. Campos, M. Carreras, D. Ribas, N. Palomeras *et al.*, "Mapping the moon: Using a lightweight auv to survey the site of the 17th century ship 'la lune'," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–8.

[5] T. Sato, S. Ikeda, M. Kanbara, A. Iketani, N. Nakajima, N. Yokoya, and K. Yamada, "High-resolution video mosaicing for documents and photos by estimating camera motion," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 246–253.

[6] R. Marzotto, A. Fusiello, and V. Murino, "High resolution video mosaicing with global alignment," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, pp. I–692.

[7] O. Pizarro, R. M. Eustice, and H. Singh, "Large area 3-d reconstructions from underwater optical surveys," *Oceanic Engineering, IEEE Journal of*, vol. 34, no. 2, pp. 150–169, 2009.

[8] J. Civera, A. J. Davison, and J. Montiel, "Inverse depth parametrization for monocular slam," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 932–945, 2008.

[9] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *International Symposium on Robotics Research (ISRR)*, 2011, pp. 1–16.

[10] F. Bellavia, M. Fanfani, F. Pazzaglia, and C. Colombo, "Robust selective stereo slam without loop closure and bundle adjustment," in *Image Analysis and Processing–ICIAP 2013*. Springer, 2013, pp. 462–471.

[11] H. Badino, A. Yamamoto, and T. Kanade, "Visual odometry by multi-frame feature integration," in *First International Workshop on Computer Vision for Autonomous Driving at ICCV*, December 2013.

[12] S. Wirth, P. L. N. Carrasco, and G. O. Codina, "Visual odometry for autonomous underwater vehicles," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–6.

[13] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 8, pp. 1362–1376, 2010.

[14] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[15] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.

[16] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[17] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[19] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 778–792.

[20] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[21] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 510–517.

[22] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[23] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 183–196.

[24] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments," in *Experimental Robotics*. Springer, 2014, pp. 477–491.

[25] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2100–2106.

[26] N. Sunderhauf and P. Protzel, "Brief-gist-closing the loop by simple means," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1234–1241.

[27] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2161–2168.

[28] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 62–69.

[29] N. Sunderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1879–1884.

[30] K. Konolige, "Small vision systems: Hardware and implementation," in *Robotics Research*. Springer, 1998, pp. 203–212.

[31] G. Bradski, *Dr. Dobb's Journal of Software Tools*, 2000.

[32] M. Quigley, J. Faust, T. Foote, and J. Leibs, "Ros: an open-source robot operating system."

[33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Visual Servoing on the Baxter Research Robot for Collaborative Tasks

José Luis Part

Abstract— **Over the last years, the use of robotic systems in public areas have been steadily increasing. There is a growing desire that robots and people can work side by side without any risks or danger and of course, without the use of safety cages. In order for this to be possible, robots have to be capable of coping with dynamically changing and unstructured environments. This supposes the development of intelligent and reactive systems capable of dealing with human-like situations.**

The work presented in this dissertation aims to offer a reliable solution for performing cooperative tasks between humans and robots by mean of the development of a visual servo control on a human-size humanoid robot. In order to improve the perception capabilities of the robot, an external depth sensor is used. The proposed method incorporates modules for human skeleton tracking, pose detection, object pose estimation and visual servoing. In addition, a method for calibrating the depth sensor coherently with the way the robot perceives the world is proposed.

I. INTRODUCTION

As technology is advancing, the separation between robots and humans in different environments is becoming narrower. There is a common interest from the research community and the industry sector in getting robots to interact with people and among themselves in a seamlessly integrated environment. The availability of new state-of-the-art sensing hardware is improving the quality and accuracy of the collected data and the emphasis put into the design of compliant actuators is allowing the use of robots in more places that, until now, were exclusive for people.

The aim of this thesis is to develop a control scheme based on visual servoing that targets the case study of a person handing over a tool, and implement it on the Baxter Research Robot platform. For this purpose, a position-based visual servoing control is proposed for collaborative manipulation tasks with the Baxter Research Robot using 3D perception.

The Baxter Research Robot has several limitations. Since it was designed to count only with the most essential hardware in order to keep its cost as low as possible, many challenging tasks cannot be performed with the in-built capabilities. In order to improve the perception capabilities, an external sensor for 3D data acquisition was employed.

This paper is divided into 7 sections. Section II contains a brief recollection of the related work in the field. Section III presents the theoretical concepts referred during the rest of the document. Section IV deals with the implementation details related to both, hardware setup and software integration. Section V presents the results obtained by the proposed solution. Section VI concludes the work done and finally, Section VII discusses the future work that can be done in order to improve the proposed solution and further extend it to other situations.

II. RELATED WORK

A lot of work has been done in the field of visual servoing. In [13], a highly redundant robot (HRP-2) uses visual servoing to catch a ball while walking on a pre-planned path. In [14], position-based visual servoing is used on the ARMAR-III humanoid robot for manipulation tasks. In this case, the robot has a physical marker in its hands and uses multi-sensor fusion in order to cope with inaccurate object localization and fuzzy sensor information. In addition, the robot counts with a database of known objects which it uses for matching and model pose estimation. More recently, a position-based visual servoing control was implemented on the REEM humanoid robot [11] in order to control the relative position of the hand with respect to a pre-grasping location with reference to the target object. In this case, visual markers were attached both to the robot hand and to the object in order to compute their relative pose through monocular vision.

III. BACKGROUND

A. Visual Servoing

Visual servoing is a technique for robot control that makes use of visual feedback to generate the control signals that will be used for controlling the robot.

Depending on the physical configuration of the robot, the visual servoing systems can be classified into eye-in-hand systems or eye-to-hand systems. The main difference between these configurations is the location of the vision sensor with respect to the robot manipulator. In the first case, the sensor is mounted on the end-effector of the manipulator, which implies that a motion of the robot will induce the same motion on the sensor. On the contrary, an eye-to-hand system has the characteristic that the sensor is mounted somewhere else with respect to the manipulator and a motion of the latter does not necessarily induce a motion on the sensor.

Another classification of the visual servoing schemes depends on how the visual features are chosen. If the features are directly obtained from a camera feed, *i.e.* image features, the scheme is referred to as image-based visual servoing (IBVS). On the other hand, if the features correspond to 3D parameters estimated from the data acquired by the sensor,

This work was supported by an Erasmus Mundus grant from the European Commission

J. L. Part is with Université de Bourgogne (France), Universitat de Girona (Spain), Heriot-Watt University (UK) and the University of Edinburgh (UK) as a student of the European Master in Vision and Robotics (ViBot). jlp30@hw.ac.uk

the scheme is referred to as position-based visual servoing (PBVS).

The goal of every vision-controlled system is to minimize an error [1] that can be expressed as:

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), a) - \mathbf{s}^* \tag{1}$$

where $\mathbf{m}(t)$ represents a vector of measurements that, along with the parameters a that characterize the acquisition sensor, conform the current visual features, and $\mathbf{s}^*$ corresponds to the desired visual features.

A classic approach requires to find the relationship between the rate of change in the visual features and the velocity of the end-effector. Such relationship can be expressed as:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v} \tag{2}$$

where $\mathbf{v} = (\boldsymbol{v_c}; \boldsymbol{\omega_c})$ represents a vector composed by the linear $\boldsymbol{v_c}$ and angular $\boldsymbol{\omega_c}$ velocities of the end-effector, and $\mathbf{L}_s$ represents the interaction matrix, also known as the feature Jacobian, which relates the first-order partial derivatives of the features with the end-effector velocities.

By combining (1) with (2), we can rewrite (2) as a relationship between the time variation of the error and the end-effector velocity:

$$\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{v} \tag{3}$$

where $\mathbf{L}_e = \mathbf{L}_s$.

In addition, since the control takes place in the joint space, (3) is normally replaced by:

$$\dot{\mathbf{e}} = \mathbf{J}_e \dot{\mathbf{q}} = \mathbf{L}_e \, {}^c\mathbf{V}_n \, {}^n\mathbf{J}_n(\mathbf{q}) \dot{\mathbf{q}} \tag{4}$$

where ${}^n\mathbf{J}_n(\mathbf{q})$ is the robot Jacobian expressed in the end-effector frame and $\mathbf{q}$ is the vector of joint angles. $\mathbf{J}_e = \mathbf{L}_e \, {}^c\mathbf{V}_n \, {}^n\mathbf{J}_n(\mathbf{q})$ is called the task Jacobian since it involves the robot Jacobian and the features Jacobian. ${}^c\mathbf{V}_n$ represents a velocity twist matrix between the camera frame and the end-effector frame.

Since the input to our system will be the variation of the error $\dot{\mathbf{e}}$ and the control command will be the vector of joint velocities $\dot{\mathbf{q}}$, it is interesting to invert the previous equation. By also adding the constraint of an exponential decoupled decrease of the error ($\dot{\mathbf{e}} = -\lambda \mathbf{e}$), (4) can be rewritten as:

$$\dot{\mathbf{q}} = -\lambda \widehat{\mathbf{J}_e^+} \mathbf{e} \tag{5}$$

where $\widehat{\mathbf{J}_e^+}$ is an estimation of the Moore-Penrose pseudo-inverse of the task Jacobian $\mathbf{J}_e$. In general, it is practically impossible to find the exact values of the interaction matrix $\mathbf{L}_e$ because they need to be computed from the measurements which are inherently noisy. Thus, an estimation is used.

B. Joint Limits Avoidance

Considering the visual servoing task as the primary task, many approaches have been proposed for the design of higher order tasks that aim to make the primary task more robust without affecting its performance, *i.e.* the regulation to zero of the errors. Such tasks include but are not limited to joint limits and kinematic singularities avoidance, obstacle and occlusion avoidance, and keeping the visual features within the range of view.

In particular, we are interested in avoiding joint limits, since reaching a joint limit will produce the failure of the primary task. Several approaches have been proposed [5-8] for solving this issue by exploiting the redundant degrees of freedom of the robot. The most classical solution is based on the definition of a cost function that has to be minimized. Such cost function is designed to be minimal at safe configuration and maximal in the vicinity of the joint limits.

In order for the secondary task to not affect the performance of the primary task, a classical projection operator onto the null-space of the primary task Jacobian is used such that the complete task can be expressed by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_e + \mathbf{P}_e \mathbf{g} = -\lambda \mathbf{J}_e^+ \mathbf{e} + (\mathbf{I}_n - \mathbf{J}_e^+ \mathbf{J}_e)\mathbf{g} \tag{6}$$

where $\mathbf{e}$ is a column vector in $\mathbb{R}^k$ that corresponds to the primary task, $\mathbf{g}$ is a column vector in $\mathbb{R}^n$ that represents the motion induced by the secondary task, $\mathbf{J}_e \in \mathbb{R}^{k \times n}$ is the task Jacobian defined such that $\dot{\mathbf{e}} = \mathbf{J}_e \dot{\mathbf{q}}$, $\mathbf{J}_e^+$ is the Moore-Penrose pseudo-inverse of the task Jacobian, n is the number of degrees of freedom of the manipulator, k is the number of components of the task function and $\mathbf{P}_e = \mathbf{I}_n - \mathbf{J}_e^+ \mathbf{J}_e$ is a projection operator onto the null-space of the task Jacobian, which guarantees that the motions induced by the secondary task are compatible with the constraints imposed by the primary task.

One of the main drawbacks of the previous approach is that it is dependent on the redundant DoF of the robot. If there are no redundant DoF, then the null-space of the task Jacobian will be the empty space and no secondary task will be realizable.

In order to overcome this limitation, a new large projection operator was proposed in [9]. The classical projection operator is constrained so it does not perturb the regulation to zero of the errors between the current and desired features. The new large projection operator instead has only the constrain that it should not perturb the regulation to zero of the norm of the error $||\mathbf{e}||$. This new projection operator is given by:

$$\mathbf{P}_{||e||} = \mathbf{I}_n - \frac{1}{\mathbf{e}^T \mathbf{J}_e \mathbf{J}_e^T \mathbf{e}} \mathbf{J}_e^T \mathbf{e} \mathbf{e}^T \mathbf{J}_e \tag{7}$$

The main disadvantage of the large projection operator is that $\mathbf{P}_{||e||} \nrightarrow \mathbf{P}_e$ when $\mathbf{e} \rightarrow \mathbf{0}$. Moreover, $\mathbf{P}_{||e||}$ becomes unstable as soon as $\mathbf{e} \rightarrow \mathbf{0}$ since the denominator in (7) becomes zero as well.

In order to overcome this issue, a switching strategy is proposed that switches from the large operator to the classical

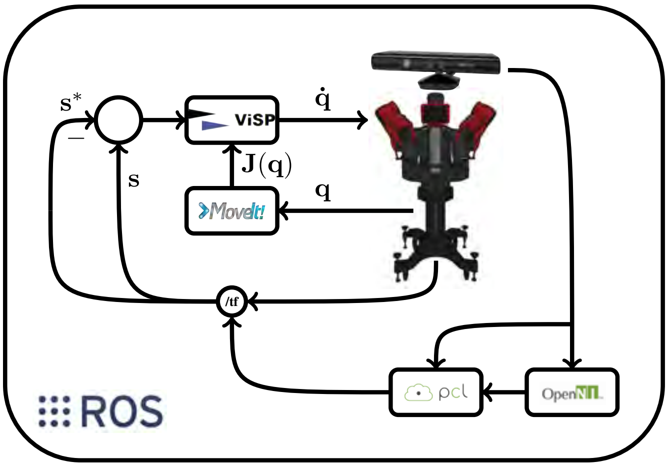


Fig. 1. Software architecture. For simplification purposes, the connexion between OpenNI and PCL was done directly although in reality, OpenNI broadcasts the skeleton joints transformations onto the **/tf** topic, which are then picked up by the pose detection and pose estimation nodes. Also, the calibration stage was not included in this diagram since it does not form part of the normal program flow.

operator when a threshold for the norm of the error is exceeded. This way, $\mathbf{P}_{||e||} \rightarrow \mathbf{P}_e$ is ensured.

The switching based projection operator is then defined as:

$$\mathbf{P}_\lambda = \bar{\lambda}(||\mathbf{e}||)\mathbf{P}_{||e||} + (1 - \bar{\lambda}(||\mathbf{e}||))\mathbf{P}_e \qquad (8)$$

where $\bar{\lambda}(||\mathbf{e}||)$ is a switching function.

IV. IMPLEMENTATION

A. Software Architecture

All the modules implemented for this work have been written completely in C++ under Linux Ubuntu 12.04 LTS. The libraries used are open-source with the exception of the NiTE middleware [21], which is embedded in the functionality of the OpenNI library [20].

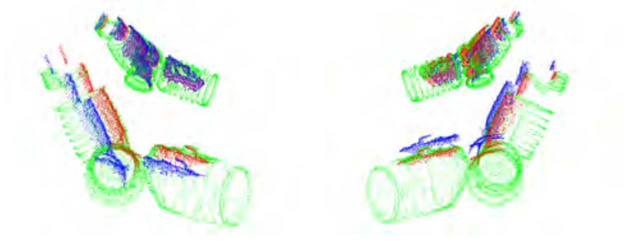
The complete system is integrated within the ROS [16] environment. The distribution adopted was Hydro.

The integration of each component is illustrated in Fig. 1. This simplified diagram shows how each library interacts with the rest of the system. Essentially, there are two main divisions, the 3D perception and the actual control of the robot.

Each constitutive functionality of this application was programmed as an individual ROS node which interacts with other nodes in the system through ROS messages. The description of each one of these integral parts is given in the following sections.

B. System Calibration

The method proposed here for calibrating the system is by mean of point clouds registration. By mean of registering the point cloud acquired by the sensor and the robot model, the rigid transformation between the coordinate frame of the sensor and the world coordinate attached to the robot can be found. The main advantage of this method is that it copes with calibration errors, both in the vision sensor and also in the joint position sensors.



(a) (b)

Fig. 2. Calibration results. Robot model (green), rough alignment of the point cloud acquired by the sensor (blue) and aligned point cloud acquired by the sensor (red).

The method used for registration is Iterative Closest Point (ICP). Prior to apply this method, the point clouds are roughly aligned. The results of the registration process can be observed in Fig. 2.

C. Skeleton Tracking and Pose Detection

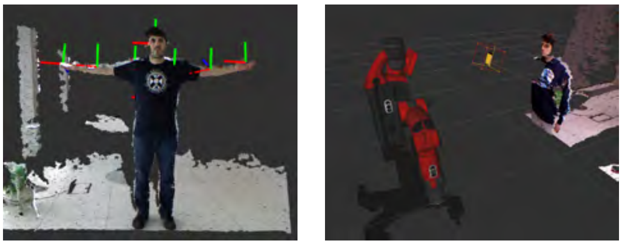
In order to perform the skeleton tracking, an existing ROS package [24] was adapted. This package consists of a node that is, in essence, a ROS wrapper for an OpenNI application that uses the NiTE middleware to deliver the joint positions in space of the detected skeleton. Although the orientation of the joints are also provided, they are not used given the unreliability inherent to the limited resolution of the sensor.

After a new skeleton has been detected in front of the sensor, the node converts the joints information into a set of pose transforms which are then broadcast onto the **/tf** topic as shown in Fig. 3(a).

After the skeleton is available, the information of the location in space of the joints are used to detect the pose of the person by mean of simple heuristics that determine if the person is facing to the front and if his hand is pointing towards the robot. Finally, the location of the hand is used to define a Region of Interest for the forthcoming steps as shown in Fig. 3(b).

D. Object Pose Estimation

In order to perform object pose estimation, several methods were proposed although the results obtained are not



(a) (b)

Fig. 3. a) Skeleton frames. b) Region of interest generated around the object after the pose of the person have been detected.

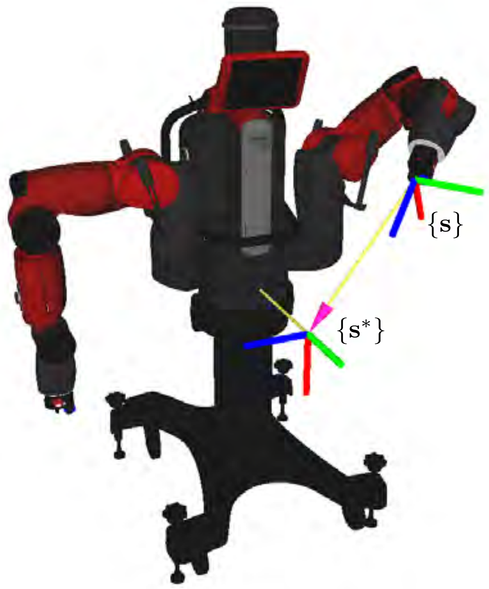


Fig. 4. Coordinate Frames used for the Visual Servoing Task. $\{\mathbf{s}^*\}$ corresponds to the desired pose reference frame and $\{\mathbf{s}\}$ to the current pose reference frame of the end-effector. Note that the arrow symbolizes that the current pose reference frame is with respect to the desired pose reference frame.

suitable for the application. Such module should be robust and computationally inexpensive because this application requires real-time response. Further research should be done in this direction.

E. Visual Servoing

1) Control Law: In order to implement the visual servoing task, the ViSP software package [3] was used. In addition, the MoveIt! library [18] was used for handling the robot model and getting the Jacobian for the arm that executes the motion.

The system was configured as an eye-to-hand system where the eye corresponds to the Kinect sensor and the hand is the end-effector that is used for the grasping task. Given that all the transformations between coordinate frames are handled by the **/tf** package [16], it is easy to reduce the necessary coordinate frames to the ones corresponding to the current and desired poses for the end-effector as shown in Fig. 4.

In order for the trajectory of the end-effector to be a straight line, a selection for the current and the desired features [1] is $\mathbf{s} = ({}^{s^*}\mathbf{t}_s, \theta\mathbf{u})$ and $\mathbf{s}^* = \mathbf{0}$, giving an interaction matrix of the form:

$$\mathbf{L_s} = \begin{bmatrix} {}^{s^*}\mathbf{R}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix} \tag{9}$$

where ${}^{s^*}\mathbf{R}_s$ is the rotation matrix between the current and desired frames, and $\mathbf{L}_{\theta\mathbf{u}}$ is the angle-axis parametrization of the previous rotation given by:

$$\mathbf{L}_{\theta\mathbf{u}} = \mathbf{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{sinc\theta}{sinc^2\frac{\theta}{2}}\right)[\mathbf{u}]_\times^2 \tag{10}$$

where θ corresponds to the angle and $\mathbf{u}$ to the axis around which the rotation is performed.

The task Jacobian can be found by combining the interaction matrix with the robot Jacobian in the end-effector reference frame:

$$\mathbf{J_s} = \mathbf{L_s}\,{}^{s}\mathbf{J}_s(\mathbf{q}) \tag{11}$$

where the twist velocity matrix is not taken into account because it is the identity matrix.

Finally, the control law is computed as:

$$\dot{\mathbf{q}} = -\lambda\widehat{\mathbf{J}_s^+}\mathbf{e} \tag{12}$$

In order to preserve the integrity of the robot and to avoid any sudden unpredicted motions, the joint velocities are limited under a defined threshold. When a velocity is above this threshold, all the velocities are scaled down with respect to the highest velocity.

2) Joint Limits Avoidance: When dealing with visual servoing, there is no implicit way to determine when a joint is near its limit. Hence, it is necessary to include a secondary task which goal is to keep the joints at a safe distance from their limits without affecting the performance of the primary task (visual servoing task).

A method for designing a secondary task for joint limits avoidance is proposed in [10] which makes use of the large projection operator defined in [9].

Writing the joint velocities as a composition of the velocities generated by the primary task and the velocities generated by the secondary task:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_2 = \dot{\mathbf{q}}_1 + \sum_{i=1}^{n}\dot{\mathbf{q}}_2^i \tag{13}$$

a suitable way for generating the vector of joint velocities $\dot{\mathbf{q}}_2^i$ for avoiding the joint limit of the i_{th} joint is:

$$\dot{\mathbf{q}}_2^i = -\lambda_{sec_i}\lambda_{li}\mathbf{P}_\lambda\mathbf{g}_i^l \tag{14}$$

where $\mathbf{g}_i^l$ is a vector indexing function that controls the activation and sign of the avoidance task, λ_{li} is a tuning function that ensures the smoothness of injecting the avoidance task into the primary task, λ_{sec_i} is an adaptive gain function used to control the magnitude of the avoidance task and $\mathbf{P}_\lambda$ is a projection operator defined by:

$$\mathbf{P}_\lambda = \bar{\lambda}(||\mathbf{e}||)\mathbf{P}_{||e||} + (1 - \bar{\lambda}(||\mathbf{e}||))\mathbf{P}_e \tag{15}$$

where $\mathbf{P}_e$ is the classical projection operator, $\mathbf{P}_{||e||}$ is a large projection operator and $\bar{\lambda}(||\mathbf{e}||)$ is a switching function.

Fig. 5 shows the evolution of the tuning function where emphasis is put into the three main regions. The first one corresponds to the safe configuration where the secondary task is not active, the second one to a warning zone where

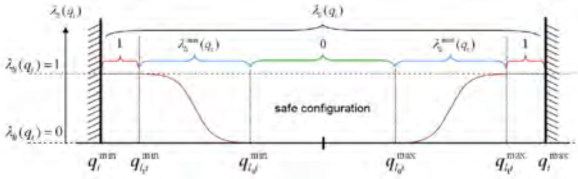
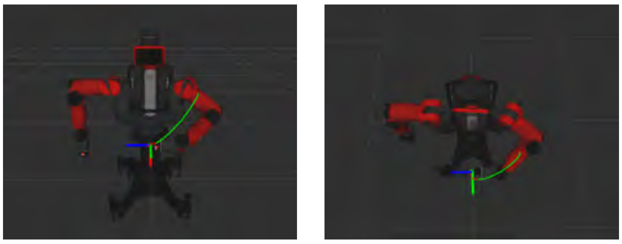


Fig. 5. Tuning function (plot taken from [10]).



(a)

(b)

Fig. 6. End-effector trajectory for $\lambda = 0.5$ while in safe configuration (green) and with joint limits avoidance active (red), seen from two different perspectives.

the secondary task is being injected progressively depending on the distance to the joint limit, and a third one to a danger zone where the secondary task is fully injected.

V. RESULTS

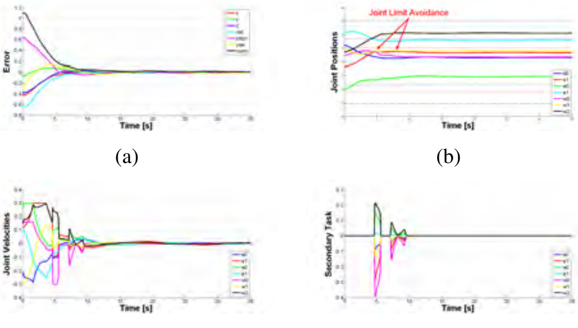
In Fig. 6, two perspectives of the trajectory realized by the end-effector are shown. From these views, it is evident that the end-effector does not follow a straight line. This is due to the fact that the errors in ”y” and ”z”, shown in Fig. 7(a), do not decrease with a pure exponential evolution but rather present some oscillations. This effect is also observed in the evolution of the error in ”yaw” although its physical effect is not perceived in the trajectory. Given that the interaction matrix used for performing the visual servoing task guarantees a decoupling between the translational and rotational velocities, the result obtained is not as expected. Further research should be carried out in order to determine the reason why this is occurring.

In Fig. 7, the evolution of the errors, the joint positions and the joint velocities are shown. In addition, the joint velocities generated by the secondary task are displayed. In Fig. 7(b) the joint limits are added in order to show how the secondary task prevents the joint ”s1” to reach its limit. If the secondary task is not added and the joint limit is reached before the primary task has converged, the latter will fail. Hence the need for the secondary task.

VI. CONCLUSIONS

The system developed during this thesis involves a great deal of software and hardware integration. This supposes also to sort out the inherent difficulties of linking different libraries together and find the correct controllers for the hardware.

Despite the issues found during the experiments, this approach shows the applicabilities of a visual servo control for robot-human interaction in unstructured environments. It also reflects the importance of defining a secondary task for joint limits avoidance.



(a)

(b)

(c)

(d)

Fig. 7. Plots for $\lambda = 0.5$. a) Regulation to zero of the feature errors. b) Joint positions with their joint limits. c) Joint velocities. d) Velocities corresponding to the secondary task.

In addition to the visual servo control, a method for system calibration that relies on point cloud registration was introduced.

VII. FURTHER WORK

During the development of this thesis, several issues arose. As a direct consequence, much of the work that was planned could not take place. In the following, a summary of the further work that could be realised in order to improve and further extend the horizons of this thesis is proposed:

- Investigate the source of the fact that the end-effector is not performing a straight line.
- Propose a real-time method for object pose estimation.
- Use of machine learning for pose detection and object recognition.
- Define a tertiary task for collision avoidance with the environment.
- Use of a filtering technique like the Kalman filter for object tracking.
- Use the panning capabilities of the Baxter Research Robot head for increasing the range of view. Another possibility would be to build a pan-tilt unit and place it on top of the robot to reduce the near field distance limitation.

REFERENCES

[1] F. Chaumette and S. Hutchinson, *”Visual Servo Control Part I: Basic Approaches”*, IEEE Robotics and Automation Magazine, Vol. 13, N° 4, pp. 82-90, December 2006.

[2] F. Chaumette and S. Hutchinson, *”Visual Servo Control Part II: Advanced Approaches”*, IEEE Robotics and Automation Magazine, Vol. 14, N° 1, pp. 109-118, March 2007.

[3] E. Marchand, F. Spindler and F. Chaumette, *”ViSP for Visual Servoing: A Generic Software Platform with a Wide Class of Robot Control Skills”*, IEEE Robotics and Automation Magazine, Vol. 12, N° 4, pp. 40-52, December 2005.

[4] F. Chaumette, E. Marchand, F. Novotny, A. Saunier, F. Spindler and R. Tallonneau, *”Building a Visual Servoing Task”*, Lagadic Group Tutorial, INRIA, October 2011.

[5] E. Marchand, F. Chaumette and A. Rizzo, *"Using the Task Function Approach to Avoid Robot Joint Limits and Kinematic Singularities in Visual Servoing"*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, pp. 1083-1090, November 1996.

[6] F. Chaumette and E. Marchand, *"A Redundancy-Based Iterative Approach for Avoiding Joint Limits: Application to Visual Servoing"*, IEEE Transactions on Robotics and Automation, Vol. 17, N° 5, pp. 719-730, October 2001.

[7] T. F. Chang and R. V. Dubey, *"A Weighted Least-Norm Solution based Scheme for Avoiding Joint Limits for Redundant Manipulators"*, IEEE Transaction on Robotics and Automation, Vol. 11, N° 2, pp. 286-292, April 1995.

[8] B. Nelson and P. K. Khosla, *"Strategies for Increasing the Tracking Region of an Eye-in-Hand System by Singularity and Joint Limits Avoidance"*, International Journal of Robotics Research, Vol. 14, N° 3, pp. 255-269, June 1995.

[9] M. Marey and F. Chaumette, *"A New Large Projection Operator for the Redundancy Framework"*, IEEE International Conference on Robotics and Automation, pp. 3727-3732, May 2010.

[10] M. Marey and F. Chaumette, *"New Strategies for Avoiding Robot Joint Limits: Application to Visual Servoing using a Large Projection Operator"*, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 6222-6227, October 2010.

[11] D. J. Agravante, J. Pagès and F. Chaumette, *"Visual Servoing for the REEM Humanoid Robot's Upper Body"*, IEEE International Conference on Robotics and Automation, pp. 5233-5238, May 2013.

[12] R. B. Rusu and S. Cousins, *"3D is here: Point Cloud Library (PCL)"*, IEEE International Conference on Robotics and Automation, May 2011.

[13] N. Mansard, O. Stasse, F. Chaumette and K. Yokoi, "Visually-guided Grasping while Walking on a Humanoid Robot", IEEE International Conference on Robotics and Automation, pp. 3041-3047, April 2007.

[14] N. Vahrenkamp, S. Wieland, P. Azad, D. Gonzalez, T. Asfour and R. Dillmann, *"Visual Servoing for Humanoid Grasping and Manipulation Tasks"*, IEEE-RAS International Conference on Humanoid Robots, pp. 406-412, December 2008.

[15] Rethink Robotics, *http://www.rethinkrobotics.com*

[16] Robot Operating System, *http://www.ros.org*

[17] Visual Servoing Platform, *http://www.irisa.fr/lagadic/visp/visp.html*

[18] Ioan A. Sucan and Sachin Chitta, *"MoveIt!"*, *http://moveit.ros.org*

[19] R. B. Rusu, *"Point Cloud Library"*, *http://pointclouds.org*

[20] OpenNI, *"OpenNI 1.5.2 Doxygen Documentation"*

[21] PrimeSense, *"PrimeSense NiTE Algorithms 1.5"*, 2011

[22] P. Mihelich, S. Gedikli, R. B. Rusu, *"Openni Camera Package"*, *http://wiki.ros.org/openni_camera*

[23] P. Mihelich, *"Openni Launch Package"*, *http://wiki.ros.org/openni_launch*

[24] Tim Field, *"Openni Tracker Package"*, *http://wiki.ros.org/openni_tracker*

Automatic Discrimination of Color Retinal Images using the Bag of Words Approach

Ibrahim Sadek*, Désiré Sidibé*, and Fabrice Meriaudeau*

*University of Burgundy, Le2i Laboratory, 12 rue de la fonderie, 71200 Le Creusot, France

Abstract—**Diabetic retinopathy (DR) and age related macular degeneration (ARMD) are among the major causes of visual impairment worldwide. DR is mainly characterized by red spots, namely microaneurysms and bright lesions, specifically exudates whereas ARMD is mainly identified by tiny yellow or white deposits called drusen. Since exudates might be the only manifestation of the early diabetic retinopathy, there is an increase demand for automatic retinopathy diagnosis. Exudates and drusen may share similar appearances, thus discriminating between them is of interest to enhance screening performance. In this research, we investigative the role of bag of words approach in the automatic diagnosis of retinopathy diabetes. We proposed to use a single based and multiple based methods for the construction of the visual dictionary by combining the histogram of word occurrences from each dictionary and building a single histogram. The introduced approach is evaluated for automatic diagnosis of normal and abnormal color fundus images with bright lesions. This approach has been implemented on 430 fundus images, including six publicly available datasets, in addition to one local dataset. The mean accuracies reported are 97.2% and 99.77% for a single based and multiple based dictionaries respectively.**

I. Introduction

According to the world health organization (WHO) diabetes mellitus (DM) is a lifelong disorder which takes place either when the pancreas doesn't produce sufficient insulin (type 1 diabetes) or when the body cannot effectively benefit the insulin it produces (type 2 diabetes). Insulin is a hormone produced in the pancreas by beta cells that regulates the level of blood sugar. Hyperglycemia, or increased blood sugar level causes serious damage to body's system, including diabetic retinopathy. The most important reasons of diabetes are increasing age, overweight, and sedentary lifestyle. During the first two decades of disease, approximately all patients with type 1 diabetes and more than 60% of patients with type 2 diabetes have retinopathy [1]. The prevalence of diabetes is estimated to increase from 2.8% to 4.4% in the time span of $2000 - 2030$. The total number of people is projected to increase from 171 million in 2000 to 360 million in 2030 [2]. Diabetic patients can prevent severe visual loss by attending regular diabetic eye screening programs and receiving optimal treatments [3]. Diabetic retinopathy (DR) and age related macular degeneration (ARMD) are among the leading causes of visual impairment worldwide. DR occurs most frequently in adult aged $(20 - 74)$ years, and it is characterized by the presence of red lesions (microaneurysms) and bright lesions (exudates) which appear as small white or yellowish white deposits with sharp margins and variable shapes located in the outer layer of the retina, their detection is essential for diabetic retinopathy screening systems. ARMD usually affects people over 50 years of age. It is caused by a damage to the macula, the small sensitive area of the retina that gives central vision (seeing fine details and colors), and categorized by drusen, tiny yellow or white deposits in a retina layer called Bruch's membrane. The Severity of ARMD can be categorized into three classes: early, intermediate, and advanced. In some patients bright lesions such as retinal exudates can be the only manifestations of early diabetic retinopathy. Thus, computer aided detection (CAD) systems have been proposed in order to detect exudates. However, these bright lesions must be identified from drusen because they share common characteristics [4]. This represents a challenge for readers or CAD based screening systems designed for DR diagnosis. Consequently, developing a CAD system for reading and analyzing retinal images decreases observational unintentional failure and the false negative rates of ophthalmologists interpreting these images. The aim of this research work is to design a system that will be able to identify normal, drusen, and exudates in color retinal images using the bag of words approach (BOW), because there is a few approaches in the literature designed for this purpose.

II. Related Work

In literature, a wide variety of CAD systems to detect retinal features and lesions involve three main steps. The first step is the preprocessing in order to compensate for great variability between and within retinal images. Green channel is considered the most preferable choice, because it provides a maximum contrast between different retinal lesions and structures. The second step is to extract candidate lesions, in some approaches feature selection may be performed in order to remove redundant features. The last step is to classify candidate lesions into normal or abnormal. Grinsven et al. [5] have proposed to use the BOW approach to retrieve and classify images with bright lesions, namely drusen and exudates. However, this approach needs a prior knowledge about the location of the optic disk and macula. Pires et al. [6] have also used the BOW in order to identify images with bright lesions such as hard exudates, cotton wool spots, and drusen, in addition to images with red lesions like hemorrhages and microaneurysms. Nevertheless, the proposed strategy requires manual annotation of unhealthy regions. Deepak et al. [7] have developed a strategy for bright lesion detection using a visual saliency based framework. This method relies on accurate detection of drusen and exudates which is considered as a challenge to any CAD system. We present a novel approach by adapting a single based and multiple based dictionaries for identifying normal images and abnormal images with bright lesions.

III. Proposed Method

The proposed method is based on the bag of words (BOW) approach to automatically discriminate between normal, drusen, and exudates in color retinal fundus images. In this approach, the images are preprocessed. Subsequently, SURF as well as HOG and LBP features are extracted from local regions of retinal images. Then, a visual codebook is constructed by adapting a K-means clustering algorithm. The cluster's centers are considered as visual words within the codebook. Each individual feature in the image is quantized to the nearest word in the codebook, and an entire image is substituted by a global histogram counting the number of occurrences of each word in the codebook. The size of the resultant histogram is the same as the number of words in the codebook and also the number of clusters obtained from the clustering algorithm. The Final histogram representation is fed into a linear kernel SVM for classification.

A. Preprocessing

The main purpose of the preprocessing step is to reduce the inter and intra patient variability. According to the paper introduced by Cree et al. [8] the background-less fundus image has normally distributed colors. Thus, the image can be represented by the scalar mean μ and standard deviation σ throughout the entire image. If these two parameters are calculated for a reference image, it is possible to equalize the colors of the new image to the reference one in a more effective manner than simple histogram equalization [9]. In this work, the mean μ and std σ are empirically chosen for all datasets instead of computing them from a reference image. Furthermore, the preprocessing is applied only to the green channel rather than the three planes of the RGB color space. All images are resized to a height of 512 pixels, while maintaining the aspect ratio because of their large sizes. The description of the process for a single color plane is explained as follows:

$$\begin{aligned}
\mu_{ref} &= 0.5 \\
\sigma_{ref} &= 0.1 \\
I_{out} &= I_{in} - \text{medianFilter}(I_{in}) \\
\mu_{out} &= mean(I_{out}) \\
\sigma_{out} &= std(I_{out}) \\
I^1_{out} &= (I_{out} - \mu_{out}) \div \sigma_{out} \\
I^2_{out} &= (I^1_{out} \times \sigma_{ref}) + \mu_{ref}
\end{aligned} \tag{1}$$

The background image is estimated by a median filter, whose size is approximately $\frac{1}{30}$ the height of the fundus image. I_{in} is the image to be equalized, I_{out} is the background-less image, and I^2_{out} is the equalized image. Fig. 1 shows an example for normal image equalization as well as drusen image equalization. Although the two images (Fig. 1 (a) and (b)) have different ethic backgrounds and quality level, the resultant (Fig. 1 (c) and (d)) images have very similar colors.

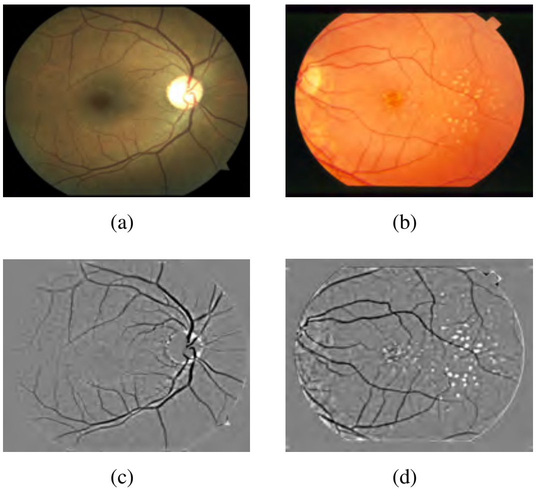

(a) (b) (c) (d)

Fig. 1 (a) Normal image, (b) drusen image, (c) equalized normal image, and (d) equalized drusen image.

B. Feature Extraction

In this approach, SURF, HOG, and LBP features are extracted from the three channels of the RGB color space. Typically, The dimension of the SURF descriptors per image are 64 × number of interest points. Two strategies are adapted such as ordinary SURF and dense SURF (DSURF). In the first, SURF descriptors are extracted from all RGB color channels, then they are horizontally concatenated to get a feature matrix of a size 64 × total number of interest points extracted from the three channels. In the second, SURF descriptors are extracted from a dense grid uniformly distributed throughout the image i.e. SURF descriptors are computed on 16 × 16 pixel patches (non overlapping) with a spacing of 16 pixels. As opposite to ordinary SURF, for each patch we get a feature vector of a dimension 64, then for each image channel we get a feature matrix of a size 64 × number of patches. Finally, each image constitutes a feature matrix of a size 192 × number of patches by vertically concatenating each feature matrix. The implementation of SURF is done using mat-lab built in function. The HOG descriptors are obtained as similar to [10], due to its lower dimension and discriminative power. Each image channel is divided into fixed number of blocks with a size of 32 × 32 pixels, then each block is subdivided into 4 cells (each cell is 16 × 16 pixels), as a result each block contributes to a feature histogram of a dimension 31. For each channel, the histograms are vertically concatenated forming a feature matrix of a size 31 × number of blocks. In total, the three channels will constitute a feature matrix of a size 93 × number of blocks. The null descriptors originating from the black area surrounding the fundus image are not are not taken into account. The LBP descriptors are extracted from local patches of a size 32 × 32 pixels similar to the HOG. However, for each patch LBP features are computed using a 3

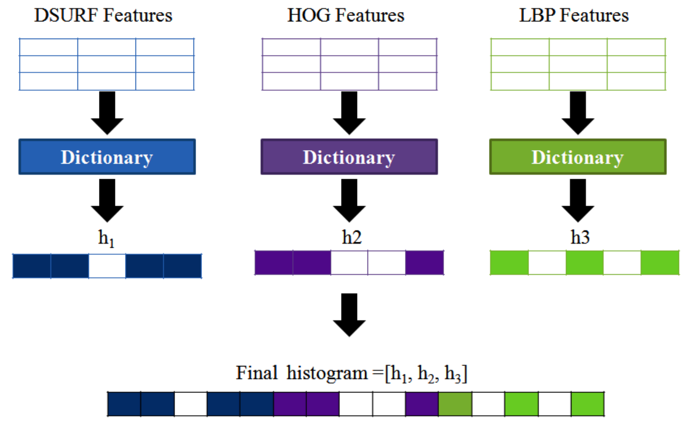


Fig. 2 Multiple based dictionary example. The set of features represent a single image in the training dataset.

× 3 moving window centered at each pixel within the patch. The uniform local binary patterns are selected because of its lower dimension and reducing the number of codes inflicted by high frequency noise. The size of the feature matrix per image is 58 × number of patches, so in total we get a feature matrix of a size 174 × number of patches. The HOG and LBP are implemented using the VLFeat open source library [11].

C. Codebook Generation

The visual dictionary is generated using two different criteria such as; a single based criterion and a multiple based criterion. In the former, a single dictionary D_i is independently constructed from a a pool of features, i.e. DSURF, SURF, HOG, or LBP using K-means clustering algorithm. Then, each image feature is quantized to its nearest visual word in every dictionary D_i. These visual words are combined into individual histograms h_i for each dictionary and the system performance is assessed accordingly. In the latter, similar steps are followed. However, the individual histograms are concatenated into a single histogram based on [12] i.e. $h = [h_1, h_2, \ldots, h_N]$. Fig. 2 shows an example of the multiple based dictionary.

D. Classification

The classification's problem has been carried out using LIBSVM [13]. The data is separated into training and testing sets, where each example in the training set contains a class label and a unique histogram counting the number of occurrences of each visual word. Based on the training data, the objective of the SVM is to produce a model which is able to estimate the target values of the test data given only the test data histograms. Assume a training set of instance label pairs (x_i, y_i), $i = 1, 2, \ldots, l$ where $x_i \in \mathbb{R}^n$ and $y \in \{1, -1\}^l$ such that $y = +1$ for positive samples and $y = -1$ for negative samples, the SVM requires solution of the following lagrange optimization problem:

$$\min_{w,b,\xi} \ \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$
$$\text{subject to} \quad y_i \left(w^T \phi(x) + b \right) \geq 1 - \xi_i \qquad (2)$$

The training vectors x_i are mapped into a higher dimensional space by a kernel function $\phi(x)$. The SVM finds a linear hyperplane which maximizes the margin $(\frac{1}{2} w^T w)$ in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. This parameter is referred to as $bestc$ which should be tuned carefully during the training phase since it significantly affects the classifier performance. There are different kernel functions available i.e. linear, polynomial, radial basis function, and sigmoid. However, in our case we consider the linear one since it provides us with the best classification results. The linear kernel function is defined as $K(x_i, x_j) = x_i^T x_j$.

IV. Dataset

In this research, we have used 430 images from six publicly available datasets as follows: STARE[1], DRIVE[2], DRIDB[3], HEI-MED[4], MESSIDOR[5], and HRF[6], in addition to one private dataset obtained from the Oak Ridge National Laboratory, USA (ORNL). For the MESSIDOR dataset, images are taken at three different clinical sites. The distribution of these datasets is described as shown in Table. I.

We have employed 81 normal images, 85 drusen images, and 264 exudate images obtained from (ORNL, HRF, DRIDB, and DRIVE), (ORNL and STARE), and (ORNL, HEI-MED, and MESSIDOR) respectively. The images are divided into two sets; Set A and Set B. Set A contains 220 images acquired from ORNL, HEI-MED, HRF, DRIVE, DRIDB, and only one clinical site of MESSIDOR named MES1 whereas Set B constitutes 210 images obtained from ORNL, HEI-MED, HRF, DRIVE, DRIDB, and two clinical sites of MESSIDOR named MES1 and MES1. The idea is to use Set A as a training set, then measure the system performance based on Set B and vice-versa. In this way, we can assess how well the system behaves when the test set contains different images than the ones included in the training set. This is usually called cross dataset testing. That means the proposed system (selected features, dictionary: single or multiple, and number of visual words) should be discriminating enough to classify the date present in the Set A based on Set B, and also the data present in Set B based on Set A. The system performance is assessed using the accuracy measurement which is calculated as follows:

$$\text{Accuracy} = \frac{\text{Total \# of correctly classified images}}{\text{Total \# of images}} \% \qquad (3)$$

[1] see (http://www.ces.clemson.edu/~ahoover/stare/)
[2] see (http://www.isi.uu.nl/Research/Databases/DRIVE/)
[3] see (http://www.fer.unizg.hr/ipg/resources/image_database)
[4] see (http://vibot.u-bourgogne.fr/luca/heimed.php)
[5] kindly provided by the Messidor program partners (see http://messidor.crihan.fr)
[6] see (http://www5.cs.fau.de/research/data/fundus-images/)

Table. I Data distribution of Set A and Set B. MES1: MESSIDOR site 1, MES2: MESSIDOR site 2, and MES3: MESSIDOR site 3.

SETA			
	Normal	**Drusen**	**Exudates**
ORNL	18	30	10
HEI-MED	...	...	13
STARE	...	12	...
HRF	7	...	...
DRIDB	5	...	...
DRIVE	10	...	...
MSE1	...	...	115
MSE2	...	...	...
MSE3	...	...	...
# of images	40	42	138

SETB			
	Normal	**Drusen**	**Exudates**
ORNL	18	31	10
HEI-MED	...	...	13
STARE	...	12	...
HRF	8	...	...
DRIDB	5	...	...
DRIVE	10	...	...
MSE1	...	...	...
MSE2	...	...	63
MSE3	...	...	40
# of images	41	43	126

V. Results and Discussion

The classifier's parameter i.e. the value of C which is refered to as $bestc$ is computed by carrying out a class classification with 10 fold cross validation. Since K-means clustering algorithm (hard assignment) is employed, different values of K are used such as $\text{K} = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$ in order to achieve satisfactory classification results. Two experiments are performed. The first experiment is to use Set B as a training set and Set A as a test set, while the second experiment is to use Set A as a training set and Set B as a test set.

A. Experiment 1

Regarding the single based dictionary, the highest accuracy 98.63% is obtained using DSURF descriptors at K=70, subsequently HOG, SURF, and LBP achieve accuracies of 97.27% at K=100, 85.91% at K=90, and 91.36% at K=80 respectively. Neither SURF nor LBP descriptors provide satisfactory results as expected. On the contrary, HOG gives approximately similar results to DSURF 97.27% at K=100. Since there is no preprocessing step to remove the optic disk, it might be confusing for the SURF or LBP descriptors to discriminate between normal and exudate images as the intensity characteristics of the optic disk is very similar to the exudate lesions. On the other hand, multiple based dictionary approach overcomes the single based dictionary. At K=100, an accuracy of 99.54% is obtained. In fact for all values of K, multiple based dictionary approach achieves higher results than the single based one, except at K=70 DSURF descriptors result 98.63% is slightly better than multiple based 97.72%. Fig. 3 shows the resultant accuracy for all descriptors versus different values of visual words.

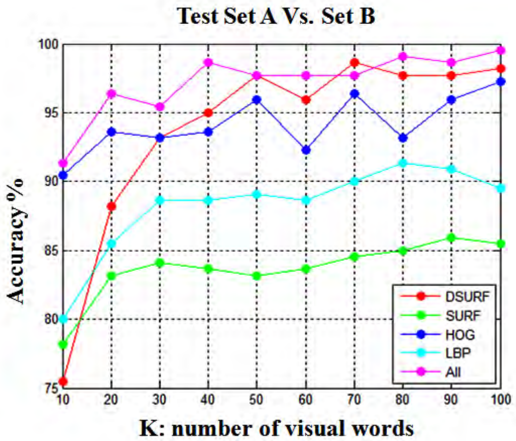


Fig. 3 Accuracy Vs. visual words K for a single and multiple based dictionary (test Set A Vs. Set B) . All: multiple based dictionary approach, DSURF, HOG, and LBP descriptors.

B. Experiment 2

With respect to the single based dictionary, the HOG descriptors achieve the highest accuracy 97.14% at K=100, after that DSURF, SURF, and HOG achieve accuracies of 90.95% at K=50, 85.23% at K=80, and 84.76% at K=70 respectively. SURF and LBP descriptors attain relatively similar results as before. We can notice that the DSURF descriptors don't attain similar performances in both experiments owing to the sharp decrease in accuracy from 98.63% to 90.95%. However, the HOG descriptors achieve satisfactory results with the two experiments which implies the discriminative power of these descriptors. Once more, the multiple based dictionary approach overcome the single based dictionary as at K=100 a 100% accuracy is achieved. Furthermore, for all visual words, it achieve higher results than the single based method as shown in Fig. 4. So far, we can conclude that the multiple based approach achieves significant results in both conditions, which indicates the importance of integrating several descriptors in the task of diabetic retinopathy diagnosis. As we discussed in section IV the proposed approach should be able to discriminate the data present in Set A based on Set B and vice-versa, the multiple based approach managed to accomplish this task with satisfying results such as 99.54%, 100% for the first and second experiment respectively and a mean accuracy of 99.77%.

VI. Conclusion and Future Work

In this work, a bag of words approach was employed in order to discriminate between normal fundus images and abnormal fundus images with bright lesions, specifically drusen and exudates. We have proposed to use a single based and multiple based dictionaries. In the first, a single dictionary is constructed from DSURF, SURF, HOG, or LBP descriptors, after that a histogram of word occurrences is generated for each image and the system performance is assessed accordingly. In the second, the image gets a histogram from each dictionary which are horizontally concatenated to form a single

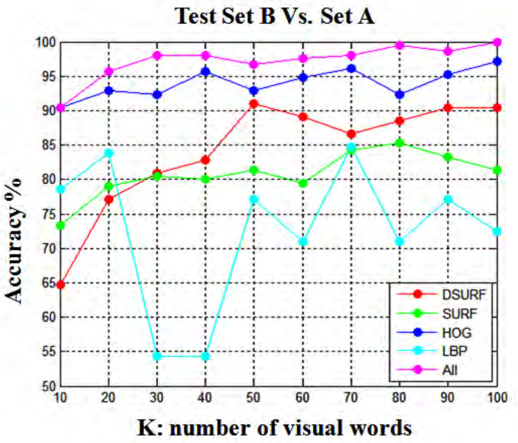


Fig. 4 Accuracy Vs. visual words K for a single and multiple based dictionary (test Set B Vs. Set A). All: multiple based dictionary approach, DSURF, HOG, and LBP descriptors.

histogram, where each feature gets N entries in the histogram, one from each dictionary. The two schemes are evaluated on different datasets. We achieved a mean accuracy of 97.2% with respect to the single based dictionary, while our best accuracy is obtained using the multiple based dictionary with a mean accuracy of 99.77% which reflects the discriminative power of this approach. To conclude, the bag of words approach can play a significant role in the classification of normal fundus images and abnormal fundus images with bright lesions, also it helps physicians in the early diagnosis of diabetic retinopathy as exudates might be the only sign of diabetic retinopathy. In the future, we would increase the size of the datasets and perform more experiments. Introduce a preprocessing step to localize and segment the optic disk, because of the confusion between normal and exudates images, add color features to SURF, HOG, and LBP descriptors, in addition to extending the proposed approach to deal with more challenging spot lesions, namely microaneurysms.

References

[1] Fong, D., Aiello, L., Gardner, T., King, G., Blankenship, G., Cavallerano, J., Ferris, F. and Klein, R. (2004). Retinopathy in diabetes. *Diabetes care*, 27(suppl 1), pp.84--87.

[2] Wild, S., Roglic, G., Green, A., Sicree, R. and King, H. (2004). Global prevalence of diabetes estimates for the year 2000 and projections for 2030. *Diabetes care*, 27(5), pp.1047--1053.

[3] Yen, G. and Leong, W. (2008). A sorting system for hierarchical grading of diabetic fundus images: a preliminary study. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1), pp.118--130.

[4] Niemeijer, M., van Ginneken, B., Russell, S., Suttorp-Schulten, M. and Abramoff, M. (2007). Automated detection and differentiation of drusen, exudates, and cotton-wool spots in digital color fundus photographs for diabetic retinopathy diagnosis. *Investigative ophthalmology & visual science*, 48(5), pp.2260--2267.

[5] van Grinsven, M.J.J.P.; Chakravarty, A.; Sivaswamy, J.; Theelen, T.; van Ginneken, B.; Sanchez, C.I., A Bag of Words approach for discriminating between retinal images containing exudates or drusen, *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on* , vol., no., pp.1444,1447, 7-11 April 2013.

[6] Pires, R.; Jelinek, H.F.; Wainer, J.; Goldenstein, S.; Valle, E.; Rocha, A., Assessing the Need for Referral in Automatic Diabetic Retinopathy Detection, *Biomedical Engineering, IEEE Transactions on* , vol.60, no.12, pp.3391,3398, Dec. 2013.

[7] Ujjwal; Deepak, K.S.; Chakravarty, A.; Sivaswamy, J., Visual saliency based bright lesion detection and discrimination in retinal images, *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on* , vol., no., pp.1436,1439, 7-11 April 2013.

[8] M. J. Cree, E. Gamble, and D. J. Cornforth, Colour normalisation to reduce inter-patient and intra-patient variability in microaneurysm detection in colour retinal images, in *Workshop on Digital Image Computing*, 2005, pp. 163-169.

[9] Giancardo, L., Meriaudeau, F., Karnowski, T., Li, Y., Garg, S., Tobin Jr, K. and Chaum, E. (2012). Exudate-based diabetic macular edema detection in fundus images using publicly available datasets. *Medical image analysis*, 16(1), pp.216--226.

[10] Felzenszwalb, P., Girshick, R., McAllester, D. and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9), pp.1627--1645.

[11] VLFeat: *An Open and Portable Library of Computer Vision Algorithms* (2008) by A. Vedaldi, B. Fulkerson.

[12] Mohamed Aly and Mario Munich and Pietro Perona. (2011). Using More Visual Words in Bag of Words Large Scale Image Search. *Caltech, USA.*

[13] Chang, C. and Lin, C. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), p.27.

Fast vision-based relocalization for MAVs

Quim Sànchez

Abstract— **Visual odometry techniques are becoming increasingly popular in robotic vehicles as a means to provide navigation information. This is further relevant in the context of Micro Aerial Vehicles (MAVs) where the payload constrains impose strong limitations on the choice of navigation sensors. Modern Visual Odometry algorithms, although quite sophisticated, are prone to failures when the tracking is lost due to image blur or sudden large changes in the image content. The loss of the tracking requires finding the location of the vehicle with respect to a previously built map (kidnapped robot problem). In this work we study and propose some relocalization solutions for Visual Odometry algorithms. This work is intended to work with SVO (Semi-direct Visual Odometry) but the proposed framework can be used with other methods. The relocalization method from PTAM is used as a base line and new alternatives based on space geometry and machine learning are proposed.**

I. INTRODUCTION

Micro Aerial Vehicles (MAVs) are about to play a major role in tasks like search and rescue, environment monitoring, security surveillance, inspection and goods delivery (Amazon). However, for such operations, navigating based on GPS information only is not sufficient. Fully autonomous operation in cities or other dense environments requires MAVs to fly at low altitudes where GPS signals are often shadowed or indoors, and to actively explore unknown environments while avoiding collisions and creating maps. Precisely autonomous operations requires MAVs to rely on alternative localization systems. For minimal weight, power consumption and budget a single camera can be used for this propose.

Real-time monocular Visual Odometry (VO) algorithms can be used to estimate the 6 DoF pose of a camera relative to its surroundings. This is attractive for many applications such as mobile robotics (and not only aerial) and Augmented Reality (AR) because cameras are small and self-contained and therefore easy to attach to autonomous robots or AR displays. Further, they are cheap, and are now often pre-integrated into mobile computing devices such as PDAs, phones and laptops.

SVO (Semi-direct Visual Odometry) [4] is a very fast VO algorithm able to run at more than 300 frames per second on a consumer laptop. It builds a map based on keyframes and salient points. Most monocular VO are feature-based where scale and rotation invariant descriptors (SIFT, SURF…) are extracted and matched in order to recover the motion from frame to frame while finally refining the pose with reprojection error minimization with the map. SVO uses a different approach by using direct methods. Instead of matching descriptors, it uses intensity gradients to minimize the error between patches around detected salient points to estimate the frame to frame transformation. Finally, it uses Bundle Adjustment to align with the map and avoid or minimize drift.

The main problem with most existing monocular VO implementations (including SVO) is a lack of robustness. Rapid camera motions, occlusion, and motion blur (phenomena which are common in all but the most constrained experimental settings) can often cause the tracking to fail. While this is inconvenient with any tracking system, tracking failure is particularly problematic for VO systems: not only is the camera pose lost, but the estimated map could become corrupted as well.

This problem is accentuated during a fast agile maneuver (e.g., a flip) and so a good relocalization is important when these are intended to be performed.

II. RELATED WORK

A. Place Recognition

Klein and Murray present in [7] the relocalization method used in PTAM [6]. PTAM is a VO algorithm based on keyframes that are used during the relocalization. The relocalization method consists of two steps. First, given the current frame, the most similar keyframe is retrieved, and its know pose is used as a baseline. As a measure of similarity the cross correlation, being the difference between subsampled, blurred and zero-mean images is used. The small blurry images are stored every time there is a new keyframe and the small blurry image of a new frame is computed during the relocalization to be compared with the keyframes.

Other methods can be used for image retrieval, for example using bag of words [14]. Nistér and Stewenius [11] propose to use a tree structure to store words in order to handle much larger vocabulary or have a much faster retrieval. Every node of the tree would have k child nodes which are the clustering results of k-means. The tree is build by recursive k-means. This structure is expensive to build because k-means is very resource consuming. During the online process, new words can be appended to the final leaves.

Özuysal et al. [12] proposes a simplified random forest classifier which relates image patches to objects. It is simplified because instead of using a tree structure, they use a linear structure applying all the binary tests to the patch. The result of the tests is a binary descriptor, the list of binary tests is called Fern. Every object is trained with multiple random warps of the known view to introduce information from possible different views of the object. In the end every object can be represented with many binary descriptors and every descriptor should output a probability distribution of possible objects represented. Evaluating multiple Ferns and joining the produced distributions, the final classification is achieved.

B. Pose Estimation

Geometric methods are typically used to find the transformation from the found keyframe using the classic pipeline of salient points detection, feature extraction and matching. The five-point algorithm can then be used to find the scaled 6 DoF transformation or the full 6 DoF with the three-point algorithm if depth is known [5].

During the second step of the relocalization, the transformation from the retrieved frame to the query frame must be calculated. This transformation will be finally appended to the know keyframe pose. In PTAM, an image alignment algorithm, Efficient Second-Order Minimization method (ESM) [2], is employed. ESM is a Gauss-Newton gradient descent algorithm, which can be used with different image warp functions. It is similar and based on the Lucas-Kanade [1] algorithm but using Second-order functions. Therefore, it results in a faster convergence.

C. Joint Place and Pose estimation

One approach to solve the relocalization problem was proposed by Williams [15]. In their implementation, they use Random Forest classifiers to characterize a salient object in space. To do so, the classifier needs to be trained with as many as possible representations of the object (multiple views). Therefore, the first time an object is found, multiple warps of the patch are used to initialize its presence in the classifier. On later encounters with the object, the classifier is incrementally trained with additional data. During the relocalization phase salient points are classified using the trained classifier and the three-point algorithm is used to recover the 6 DoF position. This method is memory expensive and requires a GPU to generate the patch warps.

Shotton et al. [13] also propose a method to solve the place recognition and pose estimation problem simultaneously using random forests. RGB-D data is used to train the classifier. In this case, all the information is encoded in the classifier so no previous data storing or computing (salient point detection, descriptor extraction, etc...) is needed. The classifier is trained to an individual RGB-D pixel, and an RGB-D pixel query will output a probability distribution over the position in $\mathbb{R}^3$. This can be applied to all pixels of a frame or to a sparse subset selection of them. Ideally, the camera pose can be inferred from only three pixels, but as the output of the classifier can be very noisy, a second step is applied. From the output from many pixels an energy function is minimized using preemptive RANSAC in order to find a pose that agrees with most of the distributions.

To train this method, a very complete dataset of RGB-D images with 6 DoF poses from the environment associated to them is needed. That makes it difficult to be used with SLAM problems where the map get populated incrementally. An online training method should be developed.

III. Approach

We propose two different approaches to address the relocalization proble. First, a local approache is based on the PTAM implementation, where two steps are performed. The first step has been named *Place Finder* and the second *Real Pose Recognition*. Multiple methods will be proposed to solve the second step. Then, on the other side, a global approach will be proposed. In this case machine learning methods (*ferns*) will be used to recognize points in space.

A. PTAM method

PTAM [6] is a VO algorithm based on keyframes and so the relocalization method proposed is based on keyframes as well. Every keyframe is associated with a camera pose that will be used to relocalize. During the relocalization there are two steps involved. We called the first step *Place Finder* and the second *Real Pose Finder*.

1) Place Finder: During this step, the algorithm tries to find the keyframe image most similar to the last acquired image. The pose associated with the most similar keyframe is used as an initial rough estimation of the current pose. The similarity score should be invariant to small view point changes because the new acquired image will, most probably, never be taken from the same pose as any of the keyframes. Also it should be fast to compute.

The used similarity score is the Cross Correlation between images meaning the sum of the squared error between two zero-mean images as in 1. To make to computation faster both images are resized become 40×30. Then, to make the images more resistant to view point changes it is blurred with a 3×3 Gaussian kernel with $\sigma = 2.5$. The resulting image is a resized, blurred and zero-mean image called *small-blurry-image*.

$$d_{CC} = \sum_{x,y}[(I(x,y) - \bar{I}) - (G(x,y) - \bar{G})]^2 \qquad (1)$$

2) ESM Real Pose Finder: The second step of the PTAM relocalization algorithm found, in order to refine the pose of the most similar keyframe to explain the current pose of the camera. During this step, in the implementation from PTAM, only rotations are corrected. An image alignment through optimization algorithm (ESM [9]) is used to find the $SE(2)$ transformation between the two, followed by a minimization to find a transformation in the world frame.

The Extended Second order Minimization algorithm (ESM) [9] is based on the algorithm proposed by Lucas and Kanade [1] in 1981. The goal of both algorithms is to align one template image T to a different input image I through a parametrised warping function.

With this goal an error function is defined on which the Gauss-Newton or Levenberg-Marquardt schemes can be applied. The error is the squared difference between the template image and the warped input image

$$e = \sum_{x} [I(W(x;p)) - T(x)]^2 \qquad (2)$$

The warping function is going to be in $SE(2)$, that is, translation and rotation of the image plane. The algorithm assumes that a current estimate of p exists and iteratively tries improves it by increments Δp. On every iteration equation 3 is solved on Δp and then it is used to update p as in eq. 4.

$$\min_{\Delta p} \sum_{x} [I(W(x;p + \Delta p)) - T(x)]^2 \qquad (3)$$

$$p \leftarrow p + \Delta p \qquad (4)$$

The solution that minimizes eq. 3 on Δp can be found in a least squares sense. The equation needs to be derived and then set it equal to zero.

The ESM algorithm used in PTAM is very similar to the derived above with the difference that while the Lucas-Kanade takes the gradient from the input image, ESM used both the gradient of the input image and the gradient of the template image and averages them.

$$\nabla I = \frac{1}{2} [\nabla I_t + \nabla I_q] \qquad (5)$$

In Figure 1 the error of the overlapped images is visualized. It can be seen that translation and rotation are well corrected but there still is a misalignment caused mostly by a change on scale which is not taken into account during the alignment.

3) Alternative Real Pose Finder: During the mapping of an area, the VO algorithm finds landmarks in the world frame which are associated with detected featured points in keyframes (i.e. every featured point in an image is related to a $3D$ position in the world frame). Given a new image, some extracted featured points can be related to a keyframe using descriptor-matching which at the same time are related to world positions. From this information the full 6 DoF translation $SE(3)$ can be computed using the prospective three-point algorithm (P3P).

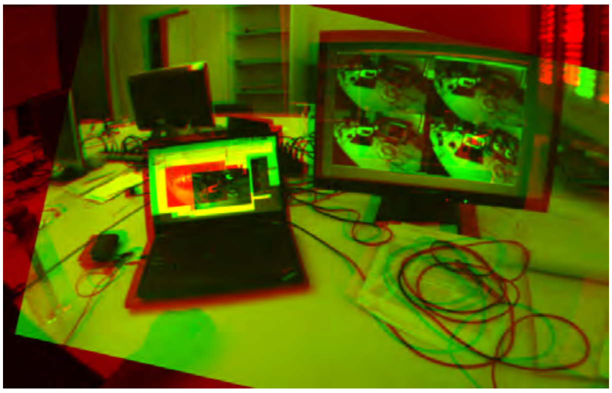


Fig. 1: $SE(2)$ transformation error visualization

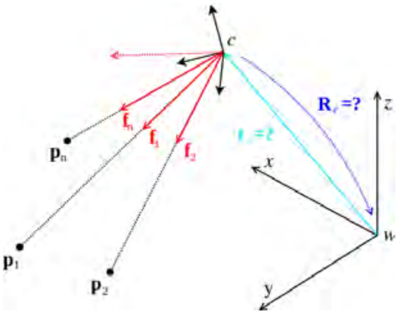


Fig. 2: From camera frame vectors f (or image pixels if the camera calibration is available) and fix points p the transformation between the two coordinate frames can be computed using the P3P algorithm. Image taken from [8]

First, descriptors from every featured point are extracted. Second, a brute force KNN matching is performed from all the extracted descriptors from on image and all the descriptors of the second image. The first, and second most similar descriptor are retrieved. Then, only good matches are kept, that is, using the matching technique described by Lowe [10], only matches with a descriptor ratio between the first and the second closest match of 0.8 or less are kept.

Finally, the three-point algorithm is fed with the pixel positions from the query image and the landmarks from the other. Because there are still outliers after the described simple filtering, this process is run in RANSAC [3] framework.

Figures 3 is an example of the described above.

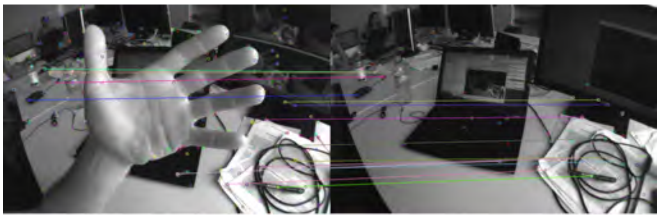


Fig. 3: Accepted matches using SIFT

IV. Using ferns

As said previously with the three-point algorithm it is possible to recover the 6 DoF of the camera pose from the relation from pixel coordinates and points in space. In this case machine learning techniques are used to model this relationship. In the classifier scheme, an object in space is a class and multiple views seen from the camera should all be classified as this class.

A *fern* [12] is a descriptor made from a set of binary tests such as in equation 6. When used as a classifier, every possible evaluation of a *fern* will contain a posterior probability distribution for every class.

$$f_j = \begin{cases} 1, & \text{if} \quad I(d_j, 1) < I(d_j, 2) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

One *fern* is usually not descriptive enough to correctly classify. In [12] it is claimed that with 50 *ferns* and $S = 11$ a problem with 200 different classes is tractable. Regarding storage requirements, it would involve $50 \times 2^{11} \times 200 = 20480000$ elements to be stored in memory. If these are stored as *float* then 78 MB are needed, which is tractable.

V. Results

The descrived methods have been tested in a desktop scene covering an area of 7x3 meters. The path taken to acquire training and testing data covering the mentioned area can be seen in 4. The dataset contains 84 frames for training and 69 for testing.

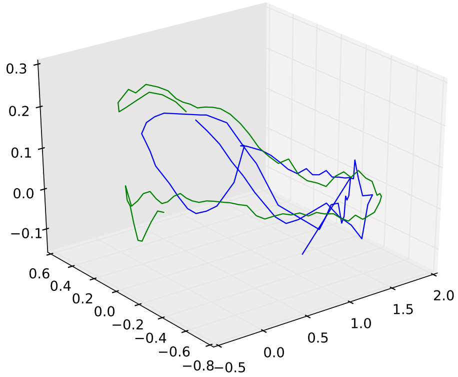


Fig. 4: In green, the path used for training. In blue, the path used for testing

A. Multi Relocalizer

1) Extended Second-order Minimization Real Pose Finder: The performance of ESM Real Pose Finder is not great in this dataset as seen in 6 although it was able to correctly relocalize in other datasets.

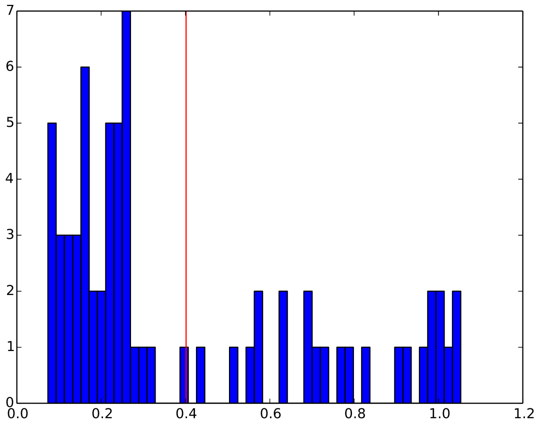


Fig. 5: Translation error histogram using CC *Place Finder* and ESM *Real Pose Finder*

2) Three-point Real Pose Finder: On the other side, the Three-point *Real Pose Finder* works very well as seen in 6. It can correctly retrieving the pose of 42 of the 69 frames. This method is very dependent on the results of the *Place Finder*, and on this dataset, the cross correlation method was not very effective. Better results on it would help iprove the results of this method and the ESM method.

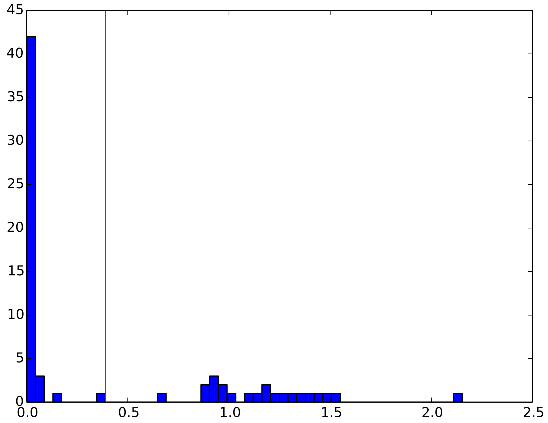


Fig. 6: Translation error histogram using CC and P3P, with the mean marked as a red line

B. Ferns

Finally, as an alternative to the PTAM relocalization method, it was proposed to use machine learning techniques, more concretely *ferns*. In [12] it is shown that *ferns* can be used to distinguish up to around 200 different classes. We applied the method to this dataset where there are 1730 classes. In figure 7 it can be seen that almost 50 of the 69 frames where correctly relocalized. Which is more than using Multi Relocalizer with the three-point *Real Pose Finder*. The classifier was trained with 100 *ferns* of 12 tests each.

It should be noticed that not all points need to be well classified, as long as more than 3 points are correctly identified then the posterior RANSAC will find and use the once that agree.

The three-point with CC method and this one, can correctly relocalize the same number of frames. Probably there is one part of the dataset that is more ambiguous and difficult to recognize and both algorithms struggle with it.

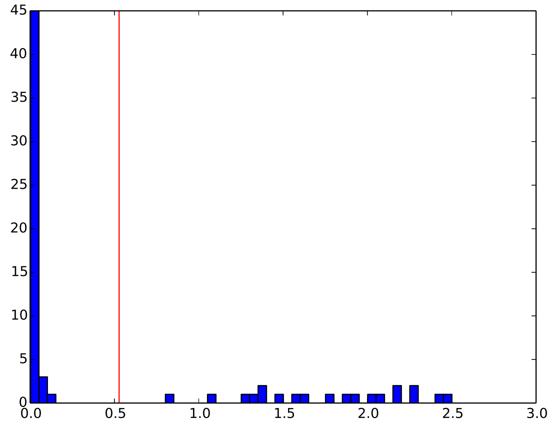


Fig. 7: Translation error histogram using *ferns* with 12 tests, with the mean marked as a red line

VI. Execution time

In figure 8 the mean execution time of relocalization is shown. There, it can be seen that the ESM and the P3P methods are the faster while the method using *ferns* classifier is slower and is sensitive to the number of classes. Also, while ESM and P3P use optimized third party libraries, the *ferns* based classifier has been integrally implemented by us, maybe not achieving the best performance. The used training time for the classifier can be seen in figure 9.

VII. Conclusions

This work has addressed an important part was missing in SVO, a good relocalization method which should recover the 6 DoF pose from only a map and a new frame. Different methods have been studied and implemented. First, as a starting point, the method from PTAM has been implemented which is based on image alignment.

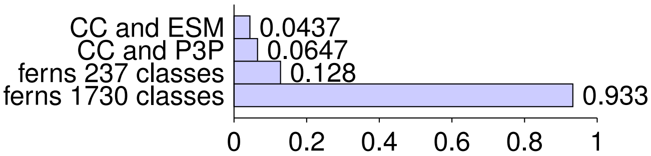


Fig. 8: Single relocalization execution time

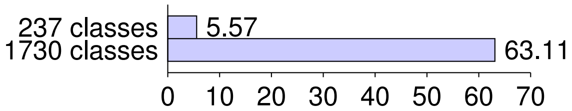


Fig. 9: *ferns* classifier training time

Then, an alternative method based on the previous has been proposed. This method is based on space geometry and on the descriptor extraction and matching framework and uses the three-point algorithm to find the camera pose by relating some world points and pixel coordinates. This method produces very accurate results when matches are found between images and in general is a great improvement over the base line, being more robust and accurate.

Finally a new approach is proposed. The central idea is to use machine learning techniques to characterise the appearance of some known points in space, to later be able to retrieve their position from the pixels of an image. *Ferns* are very similar to Random Forests but simpler and easier to implement, while being able to encode the same of information. A classifier based on *ferns* have been implemented and the same time as integrated in a relocalizer.

This method has been found to give good results even in larger areas where more than 1700 classes need to be classified even though a simplified version of the training was used.

References

[1] Simon Baker and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255, February 2004.

[2] S Benhimane and E Malis. Homography-based 2D visual servoing. *Robotics and Automation, 2006. ICRA*, 2006.

[3] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[4] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. *Proc. IEEE Intl. Conf. on Robotics*, 2014.

[5] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[6] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. *IEEE and ACM International Symposium on Mixed and Augmented Reality*, November 2007.

[7] Georg Klein and David Murray. Improving the agility of keyframe-based SLAM. *Computer VisionECCV*, 2008.

[8] Laurent Kneip and Paul Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision.

[9] Steven Lovegrove. *Parametric dense visual SLAM*. PhD thesis, Imperial College London, 2012.

[10] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[11] D Nister and H Stewenius. Scalable recognition with a vocabulary tree. *Vision and Pattern Recognition*, 2006.

[12] M Ozuysal and M Calonder. Fast keypoint recognition using random ferns. *Pattern Analysis*, 2010.

[13] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2930–2937, 2013.

[14] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. *Computer Vision, 2003. Proceedings.*, (Iccv):1470–1477 vol.2, 2003.

[15] Brian Williams, Georg Klein, and Ian Reid. Real-time SLAM relocalisation. *Vision, ICCV. IEEE 11th*, pages 1–8, 2007.

A Novel Approach to Image Calibration in Super-Resolution Microscopy

Isabel Schlangen
Heriot-Watt University
Edinburgh, UK
Email: is117@hw.ac.uk

Jérémie Houssineau
Heriot-Watt University
Edinburgh, UK
Email: jh207@hw.ac.uk

Daniel E. Clark
Heriot-Watt University
Edinburgh, UK
Email: d.e.clark@hw.ac.uk

Abstract— **For many disciplines in natural sciences like biology, chemistry or medicine, the invention of optical microscopy in the late 1800's provided groundbreaking insight into biomedical mechanisms that were not observable before with the unaided eye. However, the diffraction limit of the microscope gives a natural constraint on the image resolution since objects which are smaller than the wavelength of the illuminating light – such as proteins or ions – cannot be recognised in classical microscopy.**

Recently, different techniques have been developed to partly overcome this restriction using fluorescent molecules as markers. Like this, it is possible to monitor a vast diversity of intracellular processes on a molecular level which are of interest for biomedical research.

Since these developments in superresolution microscopy are quite recent, suitable data analysis techniques are still to be advanced. This document aims to deploy the potential of the so-called Hypothesised filter for Independent Stochastic Populations (HISP) for multi-object estimation in a biomedical context by extending its framework to a novel joint object state and sensor drift estimator.

I. Introduction

During the past decade, super-resolution microscopy has evolved extensively due to its impact on biomedical research. Novel techniques were developed to overcome the diffraction barrier of optical microscopes, either by true sub-wavelength imaging using passive illumination like near-field scanning optical microscopy (NSOM, [1]), Spatially Modulated Illumination (SMI, [2]) and 4Pi microscopy ([3]), or by so-called functional methods, for example PALM (PhotoActivated Localization Microscopy, [4]) and STORM (STochastic Optical Reconstruction Microscopy, [5]). The latter two techniques are based on the idea of active illumination by labeling the molecules of interest with a fluorescent marker and sensing the light that these emit after excitation.

Functional super-resolution imaging techniques in particular have great potential to advance the research of cell-related diseases such as diabetes or cancer since they can be performed on living cells. However, since these methods are quite recent, suitable data analysis techniques have to be developed which can cope with the following imaging artifacts inherent to super-resolution microscopy:

1) *Low SNR*
In order to capture phenomena which last only for nanoseconds, the frame rate has to be as high as possible. However, the amount of photons illuminating the sensor decrease for higher frame rates such that the molecules lose contrast against the background noise.
2) *Bleaching*
After excitation through an external light source, the molecules begin to fluoresce. Over time, the intensity of photon emission decreases which again affects the visibility of the structures of interest.
3) *Drift*
Due to the small size of the monitored objects being only a few nanometers wide, even the smallest movements of the microscope induced motor vibrations or thermal expansion could have a visible effect on the acquired images. Modern microscopes are equipped with a drift corrector already but they depend on the incorporation of reliable beads which is not always easily provided.

The first two issues can be solved on the image preprocessing level using suitable denoising and image enhancement techniques that will be described below. The correction of sensor drift, however, is usually performed separately from the actual task. Many image registration methods have been developed in the past many of which heavily rely on the extraction of reliable feature points [6]. In some cases, however, the moving objects are the only possible features and stable markers are not always available. For that reason, it is desirable to integrate the estimation of the sensor drift in the multi-object estimation process.

Previous research proved that it is possible to use particle filtering to estimate the non-linear motion of the sensor and formulate the multi-object estimation as a dependent process. In [7], a Simultaneous Localisation and Mapping (SLAM) technique is used to jointly estimate both the trajectory of the robotic vehicle and the motion of its surroundings. Another important task is joint object tracking and camera calibration [8] which is based on the same idea. Moreover, [9] resp. [10] extend this concept to introduce a first joint molecule and sensor motion estimator for biomedical applications, using the Probability Hypothesis Density (PHD) filter.

The aim of this document is to formulate an alternative to the method of [10] and to show the potential of simultaneous estimation techniques for a diversity of different applications. The new approach will incorporate the Hypothesised filter for Independent Stochastic Populations (HISP) which was formulated by Houssineau, Del Moral and Clark in 2013.

The introduction of the HISP filter gives a whole new perspective on the task of multi-object target tracking. In [11], a novel framework for treating multi-object estimation is introduced which involves the concept of distinguishability of objects which has not been considered in previous approaches like Multi-Hypothesis Tracking (MHT) or PHD filtering. Since first experiments have proved superior performance to the PHD filter especially in cases of low detection rates [12], it is desirable to explore its potential for calibration purposes. Thus, a novel joint multi-object and sensor drift estimator will be introduced on the following, extending the formulation of the general HISP filter.

II. Image Pre-Processing

In [13], different image enhancement techniques for fluorescence microscopy imaging are presented and compared. One of the more lightweight solutions in terms of computational effort is a method based on the so called à trous wavelet decomposition. Similar to the Difference-of-Gaussians method, the input image is convolved with a filter[1] several times and the differences between the outputs are computed.

In particular, let us define the one-dimensional kernel $H = \left[\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right]$ and denote the grayscale input image by $\mathscr{I} = \{I(i,j) | i \in \{1 \dots m\} \wedge j \in \{1 \dots n\}\} \in \mathbb{R}^{m \times n}$. By sequential row- and column-wise convolution with H, a sequence $\{\mathscr{I}_k\}_{k \in \{1 \dots K\}}$ is obtained where K denotes the number of applied convolution operations. Calculating the differences $\mathscr{W}_k = \mathscr{I}_{k-1} - \mathscr{I}_k$ where $W_k(i,j) = I_{k-1}(i,j) - I_k(i,j)$ for $i \in \{1 \dots m\}$ and $j \in \{1 \dots n\}$, we find the à trous wavelet decomposition

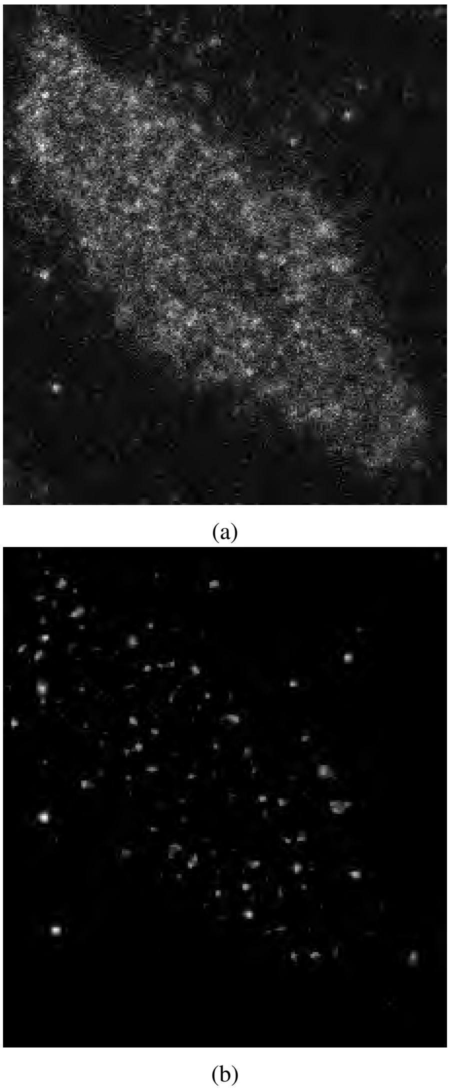
$$\mathscr{I} = \mathscr{I}_K + \sum_{k=1}^{K} \mathscr{W}_k. \tag{1}$$

Each of the layers $\mathscr{W}_k$ contains features of different sizes, so the image noise can easily be filtered out by thresholding each of the layers before reconstructing the image via (1), replacing $\mathscr{W}_k$ by the thresholded images $\hat{\mathscr{W}}_k$.

Fig. 1a shows the performance of the algorithm on the example of an input image which was acquired using the STORM technique. The resulting image is almost free of noise and shows only the relevant structures in the image. Since the focus of this method is on the size of the structures rather than the absolute intensities, possible bleaching effects are removed automatically.

III. A PHD/HISP Calibration Method

In the following, a filter for the estimation of the sensor drift is superimposed onto the PHD/HISP filtering techniques. We choose a Sequential Monte Carlo approach for the estimation of the drift since the underlying models are not necessarily linear. The PHD based calibration has been described in [10], the details are omitted here. Instead, a novel sensor drift estimator will be introduced based on the HISP technique. Thus, let us introduce some useful notation.


(a)

(b)

Fig. 1: Output of the à trous wavelet method on an example image acquired using STORM. Fig. 1a shows the noisy input image and Fig. 1b demonstrates the respective performance of the filter.

Let Ξ_t denote the sensor state space and assume a fixed number of N particles of the form $\{(\xi^n, w_n)\}_{n \in [1 \dots N]}$ where ξ_t^n represents the estimated state at time t and w_n the corresponding weight. Moreover, write $p_t^{\Xi} \in \mathscr{P}(\Xi_t)$ for the sensor state probability measure, where $\mathscr{P}(\Xi_t)$ is the collection of all probability measures on the state space. The set of possible sensor states at time t is denoted by S_t.

Each particle can either be combined with a PHD or an HISP filter for the underlying object tracking, this paper focuses on the latter.

Denote the finite set of proper observations z_t at time t by Z_t, and write $Z_t^s = Z_t \cup \phi_s'$ for the set of all observations with ϕ_s' being the empty observation. Define the set $Y_t^s = \{y_t = (z_0, \dots, z_t)\}$ to be the set of all possible sequences $z_0, \dots, z_t$ of measurements, where each $z_{t'} \in Z_{t'}$ was observed at time t'. We call the sequences y_t the observation paths

[1] Note that the word *filter* is ambiguous here. In the field of image processing and thus in the current context, filters are masks that introduce a certain effect like smoothing or sharpening on an image when they are convolved. Another class of filters are the Bayesian filters which are used for probabilistic target tracking.

up to time t. Furthermore, define $y_t^s \in Y_t^s$ to be the observation path where nothing has been observed up to time t, i.e. $y_t^s = (\phi_s', \dots, \phi_s')$. We call an object with observation path $y_s := y_t^s$ indistinguishable. Thus, the set of proper observation paths is denoted by $Y_t = Y_t^s \backslash \{y_t^s\}$. Based on this, set $p_t^{(y)}$ to be the probability density of the observation path y at time t.

Let $M_{t-1,t}(\cdot)$ be the Markov transition kernel from time $t-1$ to time t, and define the likelihood function $g_t(\cdot,\cdot)$, where $g_t(z,y)$ determines how likely the measurement z belongs to observation path y. Moreover, let $\Psi_{g_t(z,\cdot)}$ the function that performs Bayes' rule on the pdf $p_t^{(y)}$ given the likelihood $g_t(z,\cdot)$.

Finally, we introduce a function $w(y,z,\bar{n},\xi)$ that basically describes how the assignment of z to y affects the total likelihood, i.e. w determines how well the remaining paths y' and observations z' fit if the pair (y,z) would be removed.

The estimation of the posterior density $p_{t+1}^{(y)}(\cdot)$ in the formulation of the classical HISP filter will be exchanged by the joint probability

$$p_t^{(y)}(\cdot,\xi) = p_t^{(y)}(\cdot|\xi)p_t^{\Xi}(\xi) \tag{2}$$

for a given sensor state $\xi \in \Xi_t$ at time t. The resulting sensor drift estimator recursion is defined by the following three steps:

A. Initialisation

Since the first time step can be regarded as a reference frame, the sensor states of all particles will be initialised on the origin without loss of generality. Furthermore, no observations have been made in the past, so all HISP filters are initialised by setting the only existing observation path to the empty one $y_0 = () \in Y_0^s$ for all particles; the corresponding law is the initial distribution $p_0^{(y_0)}$.

B. Prediction

In the prediction, the particle filter is propagated by sampling from the previous states according to the state transition model of the system, i.e. $\xi_t^n \sim M_{t-1}^{\Xi}(\xi|\xi_{t-1}^n)$ for all $n \in \{1,\dots,N\}$. Since the movement of the sensor and the target movements are independent, the prediction of the dependent HISP filters can be performed independently by

$$p_{t+1}^{(y)}(f) = \hat{p}_t^{(y)}(M_{t,t+1}(f)) \tag{3}$$

for any test function f on the observation space and a suitable Markov kernel $M_{t,t+1}$ which models the transition from time t to time $t+1$.

C. Update

The update of the particle filter is performed using Importance Sampling with Rejection Correction (see [8]), which will be discussed in the following subsection. After that, the HISP filters are updated in dependence on the resampled sensor states.

1) Resampling: In particle filtering, a crucial element is to resample the particles according to their respective weights; in the present case, the likelihood function of the individual HISP filters provides a suitable weighting via

$$\hat{w}_n = L(Z_t^s; Y_t^n, \xi_t^n),$$

Here, $L(Z_t^s; Y_t^n, \xi_n)$ denotes the likelihood of the set Z_t^s dependent the hypotheses Y_t^n carried by the nth particle with state ξ_n. A possible specific formulation will be defined later on (cf. (12)).

While the prediction is done by sampling from the old particles according to the underlying motion model under the assumption equal weighting, the resampling in the update is performed according to the importance of each particle given by their respective likelihood. In other words, particles with high weights w_n have higher probability to create offspring than the low-weighted estimations. Hence, we can write the particle distribution for the resampling as a weighted sum of Dirac functions

$$\alpha = \sum_{n=1}^{N} w_n \delta_{\xi_n},$$

where δ_{ξ_n} equals ∞ at ξ_n and 0 everywhere else. However, it is not recommended to resample from a Dirac function since such a system can diverge quite quickly. In contrast, the distribution is made continuous by convolution with a Gaussian kernel which will be called regularisation kernel in the following[2]. In particular, let d be the dimension of the state space Ξ and define K_h to be a Gaussian rescaled by $\frac{1}{h^d}$ with zero mean and the initial particle covariance given by the sensor motion model, d denoting the dimension of the state space. Thus, the resampling distribution becomes

$$\pi(\xi) := \sum_{n=1}^{N} w_n K_h(\xi - \xi_n).$$

The usage of a regularisation kernel spreads the new particles in the proximity of the respective particle instead of just reproducing it which helps to keep up the diversity of the particle set.

Still, if the weights of the particles are very different from each other, a single resampling from π could still lead to a loss of diversity by only generating new samples from the components with high weights. In [14], a progressive resampling process is proposed which iteratively "flattens" the distribution by potentiation with a factor $c_l < 1$ and resampling therefrom; the series (c_l) slowly converges to 1 over a certain number L of iterations until the original density is regained in the Lth step. This method effects in a less aggressive selection of suitable particle positions.

Furthermore, rejection correction is used to prevent being stuck in local maxima, inspired by the approach of Markov Chain Monte Carlo sampling [14]. Thus, a newly sampled

[2] In fact, other symmetric probability density functions like the Epanechnikov kernel can be used, see [14].

particle $\hat{\xi}$ is rejected and set to its previous state ξ with probability

$$min\left\{1, k\frac{L(Z_t^s; Y_t^s, \hat{\xi})}{L(Z_t^s; Y_t^s, \xi)}\right\}$$

for a positive constant k which is a tuning parameter of the algorithm.

2) The Update of the Daughter Process: In the correction, the sensor estimator is dependent on the multi-object estimator since the measurements are only observed indirectly through the daughter process (see [7]). Thus, let us reformulate the results of [11] resp. [12] involving the sensor state estimate. The following notation will help us to recall the data association theorem which was first stated in [11] and which will be used for the final update.

Assume that the state of y is in a subset B_y of the state space $E_{s,t}$. Furthermore, let $\hat{Y} = \{y_1, \dots, y_{n_{as}}\} \subseteq Y_{t-1}$ resp. $\hat{Z} = \{z_1, \dots, z_{n_{as}}\} \subseteq Z_t$ be such that their cardinalities n_{as} are equal, i.e. $n_{as} = |\hat{Y}| = |\hat{Z}|$. Let us describe the set of possible matches between $\hat{Y}$ and $\hat{Z}$ by the set $\mathscr{S}(\hat{Y}, \hat{Z}) = \{(y_i, z_{\pi(i)}) \in \hat{Y} \times \hat{Z} | \pi \in \mathbb{S}_{n_{as}}\}$ where $\mathbb{S}_{n_{as}}$ is the symmetric group of sets with cardinality n_{as}. Furthermore, let us define four sets representing the valid associations (as), false alarms (fa), missed detections (md) resp. the remaining cases (r) as follows:

$$F_{\text{as}}(\hat{Y}, \hat{Z}) = \bigcup_{\nu \in \mathscr{S}(\hat{Y}, \hat{Z})} \left(\bigtimes_{y \in \hat{Y}} (B_y \times \nu(y)) \right), \tag{4}$$

$$F_{\text{fa}}(\hat{Z}) = \bigtimes_{z \in \hat{Z}} (E_{s,t} \times z), \tag{5}$$

$$F_{\text{md}}(\hat{Y}) = \bigtimes_{y \in \hat{Y}} (B_y \times \phi_s'), \tag{6}$$

$$F_{\text{r}}(n) = (E_{s,t} \times \phi_s')^n, \qquad n > 0 \tag{7}$$

where × denotes the Cartesian product. Theorem III.1 gives a formulation for the data association between the set of hypotheses Y_{t-1} and the observation set Z_t at time step t.

Theorem III.1 (Data association, [11])**.** *The measurable set which describes the problem of the association of the measurement Z_t and the set of hypotheses Y_{t-1} at time t can be described by the map*

$$F_\diamond(Y_{t-1}, Z_t, \bar{n}) = \bigcup_{\substack{\hat{Y} \subseteq Y_{t-1}, \hat{Z} \subseteq Z_t \\ |\hat{Y}| = |\hat{Z}|}} F_{as}(\hat{Y}, \hat{Z}) \times F_{fa}(\hat{Z}^c) \times F_{md}(\hat{Y}^c) \times F_r(\bar{n} - |\hat{Z}^c|),$$

where $\hat{Y}^c$ and $\hat{Z}^c$ denote the complements of the set $\hat{Y}$ resp. $\hat{Z}$ in Y_{t-1} resp. Z_t.

$F_\diamond(Y_{t-1}, Z_t, \bar{n})$ describes all possible associations with the resulting false alarms and missed detections for different choices of $\hat{Y}$ resp. $\hat{Z}$, where $\bar{n}$ denotes the number of indistinguishable individuals. Choosing an appropriate measure $p_\diamond$ for each part, it is possible to measure the associated probability. This result will be used in the following.

Theorem III.2 (HISP filter update in a calibration process)**.** *Let $\bar{n}$ denote the number of indistinguishable objects and assume a likelihood function $g_t(\cdot|\cdot)$. The association measure of a fixed $z \in Z_t$ with any $y \in Y_{t-1}$ dependent on the sensor state ξ is given by*

$$p_{t,z}^{(y,z)} = \sum_{\xi \in S_t} \frac{\hat{\eta}(y, z, \bar{n}, \xi)}{\sum_{\xi' \in S_t} \sum_{y' \in Y_{t-1}^s} \eta(y', z, \bar{n}, \xi')} \Psi_{\hat{g}_t(z,\cdot)}\left(p_t^{(y)}(\cdot|\xi)\right) \tag{8}$$

where $\hat{\eta}(y, z, \bar{n}, \xi)$ resp. $\eta(y, z, \bar{n}, \xi)$ are defined as

$$\hat{\eta}(y, z, \bar{n}, \xi) = p_t^{(y)}(\hat{g}_t(z, \cdot)|\xi) p_t^{\Xi}(\xi) w(y, z, \bar{n}, \xi).$$
$$\eta(y, z, \bar{n}, \xi) = p_t^{(y)}(g_t(z, \cdot)|\xi) p_t^{\Xi}(\xi) w(y, z, \bar{n}, \xi),$$

and $\Psi_{\hat{g}_t(z,\cdot)}\left(p_t^{(y)}(\cdot|\xi)\right)$ is the so-called Boltzmann-Gibbs transformation which performs the update on the underlying filters. Furthermore, the function $\hat{g}_t(z, \cdot) = \mathbb{1}_{X_t} g_t(z, \cdot)$ discards false alarms. Analogously to (8)*, define the association measure between a given $y \in Y_{t-1}$ and any $z \in Z_t$ via*

$$p_{t,y}^{(y,z)} = \sum_{\xi \in S_t} \frac{\eta(y, z, \bar{n}, \xi)}{\sum_{\xi' \in S_t} \sum_{z' \in Z_t^s} \eta(y, z', \bar{n}, \xi')} \Psi_{g_t(z,\cdot)}\left(p_t^{(y)}(\cdot|\xi)\right). \tag{9}$$

The expressions $w(y, z, \bar{n}, \xi)$ describe the joint probability of all hypotheses and measurements w.r.t. particle ξ without the pair (y, z), where

$$w(y, z, \bar{n}, \xi) = p_\diamond^{(\cdot,\xi)}(F_\diamond(Y_{t-1} \backslash y, Z_t \backslash z, \bar{n})),$$
$$w(y_s, z, \bar{n}, \xi) = p_\diamond^{(\cdot,\xi)}(F_\diamond(Y_{t-1}, Z_t \backslash z, \bar{n} - 1)),$$
$$w(y, \phi_s', \bar{n}, \xi) = p_\diamond^{(\cdot,\xi)}(F_\diamond(Y_{t-1} \backslash y, Z_t, \bar{n})).$$

With (8)/(9) *and indicator function δ_{ϕ_s}, the updated law is calculated via one of the following equations:*

$$\hat{p}_t^{(y,z)} = \left(1 - p_{t,z}^{(y,z)}(1)\right) \delta_{\phi_s} + p_{t,z}^{(y,z)}(y \in Y_{t-1}^s, z \in Z_t) \tag{10}$$
$$= \left(1 - p_{t,y}^{(y,z)}(1)\right) \delta_{\phi_s} + p_{t,y}^{(y,z)}(y \in Y_{t-1}, z \in Z_t^s). \tag{11}$$

Proof. The equality in (8) can be shown by writing down the definition of the Boltzmann-Gibbs transformation as in [11] and insert the joint probability $p_t^{(y)}(\cdot|\xi) p_t^{\Xi}(\xi)$:

$$\begin{aligned} \Psi_{\hat{g}_t(z,\cdot)}\left(p_t^{(y)}(\cdot|\xi) p_t^{\Xi}(\xi)\right)(dx) &= \frac{\hat{g}_t(z, x)\ p_t^{(y)}(dx|\xi)\ p_t^{\Xi}(\xi)}{\int p_t^{(y)}(\hat{g}_t(z, dx)|\xi)\ p_t^{\Xi}(\xi)} \\ &= \frac{\hat{g}_t(z, x)\ p_t^{(y)}(dx|\xi)}{\int p_t^{(y)}(\hat{g}_t(z, dx)|\xi)} \\ &= \Psi_{\hat{g}_t(z,\cdot)}\left(p_t^{(y)}(\cdot|\xi)\right)(dx). \end{aligned}$$

Equation (9) follows analogously. For the proof of the general HISP update and details on the involved functions, we refer to [12]. □

Note that the update of the HISP as described in III.2 is a rescaled version of the original HISP update since the additional term in the Bayes rule vanishes. As a consequence of the theorem, we choose the likelihood of a measurement set Z_t^s given the particle ξ to be

$$\begin{aligned} L(Z_t^s; Y_t^s, \xi) &= \sum_{\xi' \in S_t} \sum_{y' \in Y_{t-1}^s} \eta(y', z, \bar{n}, \xi') \\ &= \sum_{\xi' \in S_t} \sum_{z' \in Z_t^s} \eta(y, z', \bar{n}, \xi') \end{aligned} \qquad (12)$$

which comes directly from the denominator in the HISP calibration update (8) resp. (9). Note that the second equality in (12) holds since the set $\mathscr{S}(\hat{Y}, \hat{Z})$ described above can be written in each of the following ways:

$$\mathscr{S}(\hat{Y}, \hat{Z}) = \bigcup_{y_i \in \hat{Y}} \{(y_i, z_j) | z_j \in \hat{Z}\} = \bigcup_{z_j \in \hat{Z}} \{(y_i, z_j) | y_i \in \hat{Y}\}.$$

We also have to include the dependency on the sensor state in the definition of the values $w(y, z, \bar{n}, \xi)$ since these joint probabilities are also affected by the newly introduced sensor state.

Theorem III.3. *Let $p_t^{(y,z,\xi)} = p_t^{(y)}(g(z, \cdot)|\xi) p_t^{\Xi}(\xi)$. Then*

$$w(y, z, \bar{n}, \xi) = C_{as}(y, z, \xi)\ C_{fa}(z, \xi)\ C_{md}(z, \xi)\ C_r(z, \xi)$$

where the sets $C_\diamond$ are defined as

$$C_{as}(y, z, \xi) = \sum_{(\hat{Y}, \hat{Z}) \in \mathrm{A}_t(y,z)} \left[\sum_{\nu \in \mathscr{S}(\hat{Y}, \hat{Z})} \prod_{\bar{y} \in \hat{Y}} \frac{p_t^{(\bar{y}, \nu(\bar{y}), \xi)} \quad p_t^{(y_s, \phi_s', \xi)}}{p_t^{(\bar{y}, \phi_s', \xi)} \quad p_t^{(y_s, \nu(\bar{y}), \xi)}} \right],$$

$$C_{fa}(z, \xi) = \prod_{z \in Z_t \backslash z} \frac{p_t^{(y_s, z, \xi)}}{p_t^{(y_s, \phi_s', \xi)}},$$

$$C_{md}(y, \xi) = \prod_{\bar{y} \in Y_t \backslash y} p_t(\hat{y}, \phi_s', \xi),$$

$$C_r(\bar{n}, \xi) = (p_t^{(y_s, \phi_s', \bar{n})})^{\bar{n}}.$$

and $\mathrm{A}_t(y, z)$ is the set of admissible sets of the form

$$\mathrm{A}_t(y, z) = \left\{ (\hat{Y}, \hat{Z}) \mid \hat{Y} \subseteq Y_{t-1} \backslash y,\ \hat{Z} \subseteq Z_{t-1} \backslash z,\ |\hat{Y}| = |\hat{Z}| \right\}.$$

Proof. Evaluate the projection of $p_\diamond$ on F_{as}, F_{fa}, F_{md} and F_r separately:

$$p_\diamond(F_{\mathrm{as}}(\hat{Y}, \hat{Z})) = \sum_{\nu \in \mathscr{S}(\hat{Y}, \hat{Z})} \prod_{y \in \hat{Y}} p_\diamond^{(y, \nu(y), \xi)}(B_y \times \nu(y)),$$

$$p_\diamond(F_{\mathrm{fa}}(\hat{Z})) = \prod_{z \in \hat{Z}} p_\diamond^{(y_s, z, \xi)}(E_{s,t} \times z),$$

$$p_\diamond(F_{\mathrm{md}}(\hat{Y})) = \prod_{y \in \hat{Y}} p_\diamond^{(y, \phi_s', \xi)}(B_y \times \phi_s'),$$

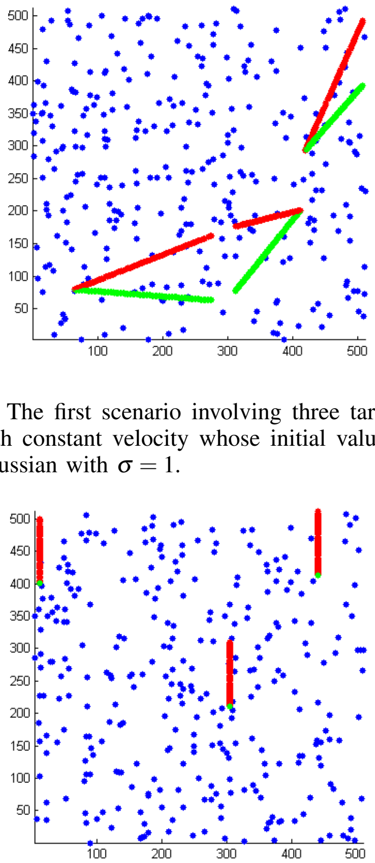
$$\begin{aligned} p_\diamond(F_{\mathrm{r}}(\bar{n})) &= p_\diamond^{(y_s, \phi_s', \xi)}((E_{s,t} \times \phi_s')^{\bar{n} - |\hat{Z}^c|}) \\ &= p_\diamond^{(y_s, \phi_s', \xi)}(E_{s,t} \times \phi_s')^{\bar{n} - |\hat{Z}^c|}. \end{aligned}$$

The result is obtained via factorisation of the above. □

IV. Experiments

In order to evaluate the performance of the proposed approach, it will be contrasted against the PHD based calibration introduced in [10].

Two scenarios are simulated to highlight different aspects of the calibration. The drift which will be estimated is set to have a constant velocity of $(0, 1)$ throughout all runs, where 1000 particles are used. For the sake of computational complexity, the number of simulated molecules which are present in each frame is set to 3. In both cases, we generate 100 square images of size 512px × 512px, and independent and identically distributed (i.i.d.) background noise is introduced whose cardinality is Poisson with mean $\lambda = 3$. The probability of detection is set to 0.9.



(a) The first scenario involving three targets with constant velocity whose initial value is Gaussian with $\sigma = 1$.

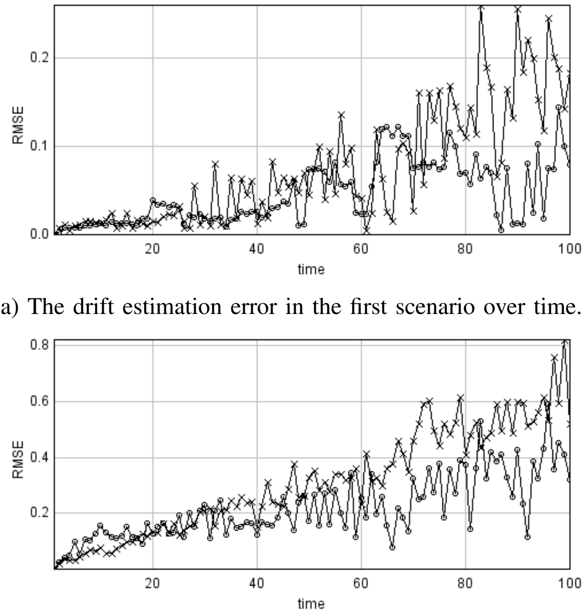
(b) The second scenario showing three static targets having small white noise with $\sigma = 0.1$.

Fig. 2: Examples of the two test scenarios for the algorithm evaluation, estimating a linear drift with constant velocity $(0, 1)$ over 100 frames. The ground truth is shown in green, whereas the simulated measurements are plotted in red. The blue marks indicate the total background clutter.

To cope with the randomness of the particle filter, the results are averaged over 20 Monte Carlo runs.

A. First Simulation

The first scenario assumes a linear movement on the three objects on in random, but fixed directions up to a small white noise. Their initial positions are set to random, i.i.d. positions in the field of view. An example of the total plot of one Monte Carlo run is shown in Fig. 2a, showing the true positions over all frames in green and the simulated, drifting measurements in red.



(a) The drift estimation error in the first scenario over time.

(b) The drift estimation error in the second scenario over time.

Fig. 3: Error plot showing the Root Mean Square Error (RMSE) of the drift position vector over time, averaged over 20 Monte Carlo runs. The crosses mark the PHD filter results, the circles stand for the HISP calibration performance.

In this simulation, the system will be supported by initialising the velocity vector to the true state $(0,1)$. Like this, it is possible to determine to which extent the movement of the targets affects the drift estimation. Resampling is performed with rate $c = 0.8$.

The Root Mean Square Error (RMSE) of the estimated sensor drift over time gives a useful visualisation of the filter's behaviour. Fig. 3a illustrates that gradually, the uncertainty in the movement of the targets results in growing uncertainty in the drift estimation, yet the error is still reasonably low. However, it can be noted that the HISP filter calibration (marked with $\circ$) shows an overall better performance, arriving at a mean RMSE value of under 0.1, whereas the PHD calibration error (marked with $\times$) rises to an average value around 0.2.

B. Second Simulation

In the second case, the three targets are initialised as static objects on random, i.i.d. positions with a small white noise (cf. Fig. 2b). The measurement and motion models for the PHD/HISP filtering are intialised accordingly to the parametrisation of the simulated data, and the resampling rate c is set to 0.2.

In contrast to the first case, the particle positions are initialised at zero position with random velocity, i.e. now, the filter has to estimate the sensor drift without prior knowledge about its true value. Consequently, we set the initial uncertainty of the particle position to 1.1 and the standard deviation of the velocity to 0.1 such that the particle filter has the possibility to explore the state space.

If we plot the RMSE between drift estimate and ground truth against time again (Fig. 3b), it can be seen that the HISP filter outperforms the PHD approach by a factor $2:3$.

V. Conclusion

A novel sensor calibration method has been derived based on the Hypothesised filter for Independent Stochastic Populations (HISP). First experiments showed that it outperforms the existing PHD filter approach already for simple scenarios with high detection rate; even better performance can be expected in scenarios with low detection probability since the classic HISP filter was previously proven to handle low detection rates better than the PHD filter. Thus, the HISP calibration could be of invaluable importance for the analysis of super-resolution images in biomedical applications since it makes the usage of reliable markers obsolete.

VI. Acknowledgments

Isabel Schlangen has an Erasmus Mundus Scholarship sponsored by the European Union. The authors thank the Life Science Interface Laboratory (LSI) at Heriot-Watt for providing the PALM/STORM data.

References

[1] D. W. Pohl, "Optical near field scanning microscope," 1982, uS Patent No. 4604520.

[2] M. Hausmann, C. Cremer, J. Bradl, and B. Schneider, "Wave field microscope with detection point spread function," Mar. 11 2008, uS Patent 7,342,717. [Online]. Available: http://www.google.com/patents/US7342717

[3] S. Hell, "Double-confocal scanning microscope (Doppelkonfokales Rastermikroskop)," 1991, european Patent No. 0491289.

[4] E. Betzig, G. Patterson, R. Sougrat, O. Lindwasser, S. Olenych, J. Bonifacino, M. Davidson, J. Lippincott-Schwartz, and H. Hess, "Imaging Intracellular Fluorescent Proteins at Nanometer Resolution," *Science*, vol. 313, No. 5793, pp. 1642–1645, 2006.

[5] M. Rust, M. Bates, and X. Zhuang, "Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM)," *Nature Methods*, vol. 3, pp. 793–796, 2006.

[6] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, pp. 977–1000, 2003.

[7] C. S. Lee, D. E. Clark, and J. Salvi, "SLAM With Dynamic Targets via Single-Cluster PHD Filtering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7,3, pp. 543–552, 2013.

[8] B. Ristic, D. Clark, and N. Gordon, "Calibration of Multi-Target Tracking Algorithms Using Non-Cooperative Targets," *13th Conference on Information Fusion (FUSION)*, pp. 1–8, 2010.

[9] J. Franco Monsalve, "Simultaneous Tracking of Multiple Particles and Sensor Position Estimation in Fluorescence Microscopy Images," Master Thesis, Heriot-Watt University, Edinburgh, 2013.

[10] J. Franco, J. Houssineau, D. Clark, and C. Rickman, "Simultaneous Tracking of Multiple Particles and Sensor Position Estimation in Fluorescence Microscopy Images," *IEEE International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 122–127, 2013.

[11] J. Houssineau, P. Del Moral, and D. E. Clark, "General Multi-Object Filtering and Association Measure," *5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2013.

[12] J. Houssineau and D. E. Clark, "Hypothesised filter for independent stochastic populations," 2014, arXiv: 1404.7408.

[13] I. Smal, M. Loog, W. Niessen, and E. Meijering, "Quantitative Comparison of Spot Detection Methods in Fluorescence Microscopy," *IEEE Transactions on Medical Imaging*, vol. 29, 2, pp. 282–301, 2010.

[14] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science. Springer, 2001.

Semi-supervised Learning for Medical Image Segmentation

Viktor Stefanovski

***Abstract*— Semi-supervised learning represents novel machine learning concept incorporating supervised and unsupervised learning procedures. In this paper image segmentation model based on semi-supervised learning with the help of decision forest is presented. The model incorporates supervised learning and furthermore semi-supervised learning procedures that can be used for medical image analysis. Through variation of numerous synthetic data as well as real data scans the validity and importance of the developed model is demonstrated.**

I. Introduction

The art of Image Segmentation consists in successful separation of given image into separate regions of interest or even more specifically separate segments containing same or at least similar features. It represent important part of the Computer Vision science and it serves as analytical tool of spatial content of pictures.

One of the hottest topics as of late in the field of machine learning is the semi-supervised method of learning. With its help a set of previously unlabelled and even unseen data can be segmented to a high level of certainty with the help of previously labelled data.

Starting from the idea of self-learning which appeared in 1965, going through supervised learning we come to semi-supervised learning as a popular approach in most recent times. What this basically represents is a combination of the supervised and unsupervised learning, or using unlabeled data (or in some cases previously unseen data) as well as labelled data with the goal of improving the accuracy of the learning procedure. Semi-supervised learning is based upon the following necessary assumptions:

- Smoothness assumption - the label function is smoother in high-density regions as opposed to low-density regions
- Cluster assumption - points of same cluster are very likely to be in same class, and
- Manifold assumption - high-dimensional data lies on a low-dimensional manifold

Semi-supervised methods have transductive nature, which means that they have the task of assigning class labels to already available unlabelled data points. On the other side of the spectrum is the inductive principle, characteristic of the supervised methods, where previously unseen test data is labelled with the corresponding class.

Majority of applications of the semi-supervised learning come from the Medical Image Analysis branch, such as [11], [12], [13], [16] and [7]. Nevertheless, the semi-supervised approach finds broad occupancy in other branches like biology - [3] and [14], video segmentation [2], activity recognition [1] and text classification [15]. It can even be used in conjunction with boosting technique, as stated in [9]. Although semi-supervised learning's classes of algorithms may not be explicitly derived from all of the previously mentioned assumptions, in general they implement one if not more of them. So according to the specific assumption behind them, algorithms are separated into the following three classes:

- Generative Models
- Low-Density Separation, and
- Graph-Based Methods

Alternatively, in most recent times the co-training technique has made its way into the semi-supervised learning world. The idea of co-training lies behind the consideration of two or even multiple views of objects to be classified, which in reality can be considered to represent a special case of Bayesian inference using conditional priors.[8]

This paper is organized as follows: Section II gives the problem description and methods used in performing the semi-supervised segmentation. The implementation of the methods used is explained in details in Chapter III. Section IV presents the results of the supervised and semi-supervised segmentation experiments for both the synthetic and real data and gives their evaluation. In section V the obtained results are discussed and lastly in Section VI conclusions are drawn and future work is formulated.

II. Methodology

A. Decision (Random) Forest

Randomly trained forests (or Random Forest) on high level is two-phase process, consisting of: ”off-line” phase - training and ”on-line” phase - testing.

1) Training phase: Essentially, the training phase is taking care of finding the type and parameters that given a training set optimize a chosen objective function at each node j - maximizing the information gain by searching over a discrete set S_j of possible parameter settings Θ:

$$\Theta_j = argmaxI\left(S_j, \Theta\right) \tag{1}$$

The formula used above, and all of the remaining formulas in this paper originate from Criminisi's book.[5]

Here we are still considering a fully supervised decision so the Information Gain has the following structure:

$$I\left(S_j, \Theta\right) = H\left(\tilde{S}_j\right) - \sum_{i \in [L,R]} \frac{\left|\tilde{S}_j^i\right|}{\left|\tilde{S}_j\right|} H\left(\tilde{S}_j^i\right) \tag{2}$$

This work was supported by TMVS

V. Stefanovski is with Toshiba Medical Visualization Systems (TMVS), EH6 5NP Edinburgh, Scotland, UK Vstefanovski@tmvse.com

where the entropy of the subset $\tilde{S} \in S \cap L$ of training points is: $H(\tilde{S}) = \sum_c p(c)$ log p(c)

2) Test Phase: Tree testing, or the "on-line" phase's task is to label (segment) the data (input test images). This process lasts until the point has reached a leaf node. Here, in the leaf nodes, are contained the classifiers also known as predictors or estimators which associate output with the input *v*.
The output can be *hard* or *soft* probabilistic output.

B. Semi-Supervised Segmentation Forest

In the name of the algorithm there is the notion of semi-supervised, referring to semi-supervised learning - subdiscipline of machine learning. This implies the learning method used when training the segmentation forest. More precisely, semi-supervised learning, same as supervised learning, is twofold process, consisting of training phase and test phase. Firstly, during the training phase classifier is trained, and then used during the test phase for assigning class labels to the different tissues (organs, air, bones etc.) which are present in the test images.
During training every tree is trained independently to each other. For the purpose of training a feature vector representing the feature space and feature vector representing the Ground Truth labels are extracted from the input (training) images. By feature space is implied either the composition of intensity values of either each individual pixel/voxel only or the composition of every pixel/voxel together with their immediate, or further neighbours.
During the test phase, the extracted classifier (model) is used in conjunction with the feature vector obtained from the test images to assign the corresponding class statistics (histograms). The results comprise of *hard* probabilistic histogram-based probability vectors which determine the label to be assigned, as well as *soft* probability clouds.
In order for the result to be meaningful, the training labels should be of discrete, categorized type, so that the output result classes can be of the same type.
The main ingredient making the process semi-supervised is encompassing information for both the labelled and unlabelled pixels/voxels in the images. What this means is that both information from unlabelled and labelled samples are used for the optimization of a training objective function. Effectively, this implies that the training phase will be achieved by optimizing parameters in each node of the forest. The function used represents decision evaluation function whose purpose is maximizing a mixed information gain. By mixed information gain we refer to the following:

$$I\left(S_j, \Theta\right) = I^U(S_j, \Theta) + \alpha I^S(S_j, \Theta) \tag{3}$$

where I^U is the unsupervised term depending on all the unlabelled data used, and I^S is the supervised term depending naturally on all the labelled data used. α is the factor that is used to weight the two terms.
I^S is given in equation (2), whereas I^U is defined via the differential entropy of a multivariate Gaussian density:

$$I^U(S_j, \Theta) = log|\Lambda(S_j)| \sum_{i \in [L,R]} \frac{[S_j^i]}{[S_j]} log|\Lambda(S_j^i))| \tag{4}$$

where Λ is the determinant of the corresponding covariance matrices S.
All of this information is used for training and correspondingly testing weak learners. There are multiple types of weak learners such as: Conic section learner, distance learner, 2D linear decision learner etc. but in the current implementation only the decision stump weak learner is used. It looks along a random subset of dimensions of the presented data, and based on a certain criterion selects threshold that maximize the previously described Information Gain in class labels.
The tree training procedure is where the class statistics. i.e. where the propagation of data takes place while training weak classifiers at each node. This class statistics actually represent the leaf distribution in the tree and are represented by accordingly normalized histograms.
The class labeling employs the Maximum Posterior Probability approach and not the Likelihood approach. This is done by incorporating equal priors to each class which discourages overwhelming the learners from a dominant class in an image.
On the top level of the algorithm we can find the discrimination of the leafs which contain only unlabelled data, effectively called unlabelled leafs or both labelled and unlabelled data, referred to as mixed leafs. And accordingly, when performing the semi-supervised learning procedure, the respective label propagation.

C. Label Propagation

Using only a certain partition of labelled data underlines the need of this so-called label propagation function which has the task of propagating the annotated labels to the available un-annotated samples. Most effective way of performing this propagation procedure is by minimizing a geodesic distance function which actually employs minimization of the generic geodesic distance formula by finding the shortest geodesic path. More elaborate explanation of this geodesic function minimization is given in [5].
Alternatively, there exists a very useful approximation which is implemented in the developed algorithm. It stats that if leafs belonging to same cluster can be associated with the same Gaussian, then the label propagation can be implemented throughout this approximation stating that we can act upon each leaf cluster, as opposed to each individual point. Algorithm 1 shows short outline of the performed tasks.

The local distances d(-,-) are defined by the symmetric squared Mahalanobis distance formula:

$$d(s_i, s_j) = \frac{1}{2}(d_{ij}{}^T \Lambda^{-1}_{l(v_i)} d_{ij} + d_{ij}{}^T \Lambda^{-1}_{l(v_j)} d_{ij}) \tag{5}$$

where $d_{ij} = s_i - s_j$ and $\Lambda_{l(v_i)}$ the covariance associated with the training data at the leaf reached by the point v_i in the *l*th tree.

```
Data: Label Propagation Algorithm
Result: Assigns labels to unlabelled leafs
initialization;
while All the trees have been trained do
    for Every tree do
        for Every leaf do
            Find unlabelled leafs;
            Find mixed leafs;
            Calculate local distance between unlabelled
            and mixed leafs;
            for Every unlabelled leaf do
                Assign nearest mixed leaf class label to
                all points
            end
        end
    end
end
```

Algorithm 1: Label Propagation Algorithm

D. Synthetic Data

Initially, synthetic three-class problem was developed. The objective of this was facilitating the process of understanding and grasping the concepts used when working with decision forest (also known as random forest). The idea behind this is starting with simple environment and then gradually moving towards more sophisticated and complex problems, by constantly upgrading the algorithms.

E. Real Data

There could be up to hundreds or even thousands of virtual slices in one CT volume. To simplify the experiment, we reduced the problem to two dimensions by selecting a single slice from each CT volume. Selection of the ground truth slices was made by an anatomist at Toshiba, where using the specific company tools secondary captures with meaningful information were selected and extracted across dozen datasets.

Consequently, these secondary captures were used for manually labeling Ground Truth of the organs that were visible across each individual image. The end product was Ground Truth containing 5 classes. Those classes are respectively: Left Kidney, Right Kidney, Aorta, Vertebra and Background. The background encompasses the rest of the organs and other entities in the bounds of the scanned body as well as the area outside the body.

First step in this dicom image transformation is transforming the pixel intensities in the raw images to their HU (Hounsfield Units) value by using the dicom tags RescaleSlope and RescaleIntercept. Effectively this is done by the following formula:

$$Im(HU) = inIm * ReSlope(inIm) + ReInter(inIm) \quad (6)$$

Ultimately, because of the size of the input images (512x512) we choose to rescale (downscale) them by a factor of 4 to reduce training and test time. The actual (final) rescaling factor used represents the ration between the values of the PixelSpacing tag and the downscaling factor used.

III. Implementation

All the code was developed under the Matlab R2014a platform. During initial familiarization with the concept of decision forests, open source implementation of the fully supervised decision forest was discovered.[1]. Semi-supervised segmentation encourages labelled training data separation as well as high-density regions separation. Practically this is done by maximizing a mixed Information Gain. By mixed Information Gain is implied that there is unsupervised term which is defined in congruence with the weighted by the parameter *alpha* supervised term. This equation is given by the formula in (3) where effectively the unsupervised Information Gain (4) has the value of the difference between the logarithm of the determinant of the covariance matrix of the descriptor and the logarithms of the determinants of the covariance matrices in the right and left children nodes, all normalized by the n-th root of the dimension of the size of the descriptor in terms of number of features.

A. Label Propagation

Practically, this is implemented by going through all the trees in the forest and all of their leafs and finding the pseudo-inverse covariance matrices while getting rid of the first dimension of the covariance matrices which indicates the number of unlabelled/mixed leafs. In [6] we can find the detailed explanation of the geodesic distance transform algorithm.

IV. Results

Key questions to be answered by evaluating the results of this section are: How does semi-supervised results compare to supervised results and are they viable option in performing the segmentation of medical images? What are the optimal forest parameters values to be used for training the forest models, and why have they been chosen? How does the number of supervised (labelled) samples, and additionally how does the number of unsupervised (unlabelled) samples affect the segmentation process?

A. Synthetic Data Supervised Results

In Fig. 1 we can observe the test results of supervised segmentation using optimized forest parameters. The result obtained for the supervised segmentation is mean error of 0.21%.

B. Synthetic Data Semi-Supervised Results

The result error rate for the semi-supervised segmentation is very close to the error rate for the supervised experiments making it theoretically viable option for segmenting large datasets which do not have to be densely labelled.

In the next table - (Tab. I) we can observe the results of semi-supervised segmentation on synthetic data when only half labelled samples are taken into consideration while using 100

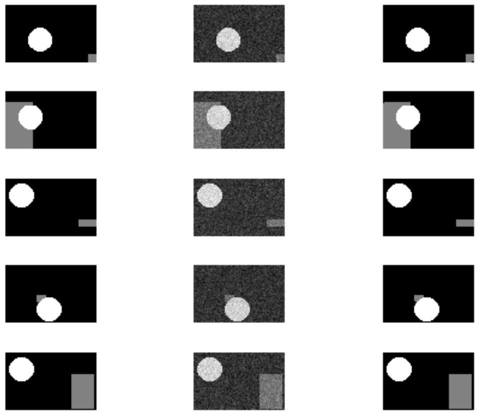


Fig. 1: Supervised segmentation test phase. Ground Truths (left), test images(center) and result(right)

TABLE I: Different percent of labelled data used for Semi-supervised segmentation

Semi-supervised learning using various percentage of labeled samples				
Labelled samples	Propagation t	Train t	Test t	Mean Error
100%	N/A	69 sec	13 sec	0.035%
80%	216 sec	82 sec	14 sec	0.045%
50%	1477 sec	93 sec	14 sec	0.06%
25%	4066 sec	91 sec	13 sec	0.08%
5%	8122 sec	91 sec	13 sec	0.135%
1%	5951 sec	91 sec	14 sec	0.39%
0%	N/A	0.4 sec	1 sec	100%

trees and the process of label propagation. The 100% is the supervised case. Further along, when decreasing the percent of sampled data used the label propagation time increases as well as the error rate. The training and test time on the other hand remain practically the same. Nevertheless we can see that for 1 percent data (only a handful of pixels) still the error rate is considerably inside the limits of reasonable error. In the 0 % case there will be 100% error because if there is no labelled data then there are no samples. In the following figures - (Fig. 2 and Fig. 3) we can see the results when taking only 10 percent of labelled data. When introducing the notion of semi supervised learning and consequentially the mixed information gain one of the most important parameters is the α parameter (3.4). The appropriate tuning of this parameter is important for result smoothing. Having said that we should bear in mind that the above-mentioned point made that results from the synthetic data experiments are very robust. But still taking non-random or non-constant value for α is better alternative than not doing so. After a long series of experiments the following graph (shown on Fig. 4) describing alpha experiments was used for creating the appropriate alpha weighting function.

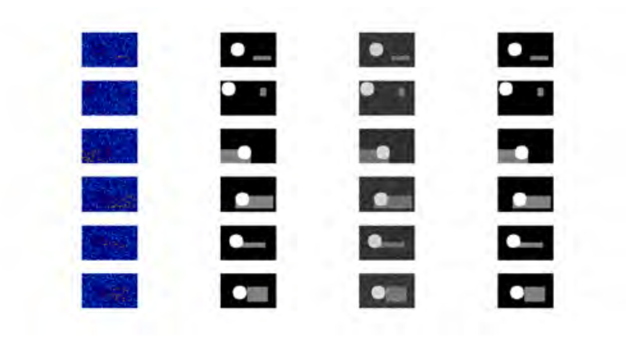


Fig. 2: Semi-supervised segmentation training phase. Samples used (left), Ground Truth (center left) test images (center right) and results (right)

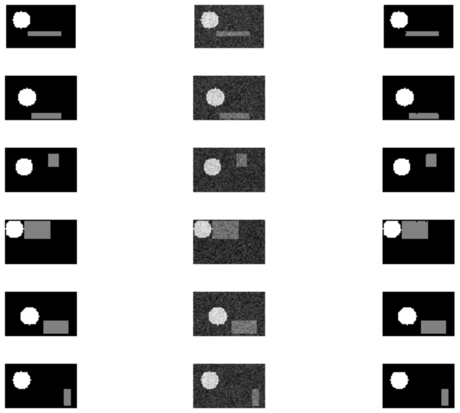


Fig. 3: Semi-supervised segmentation test phase. Ground Truth (left) test images (center) and results (right)

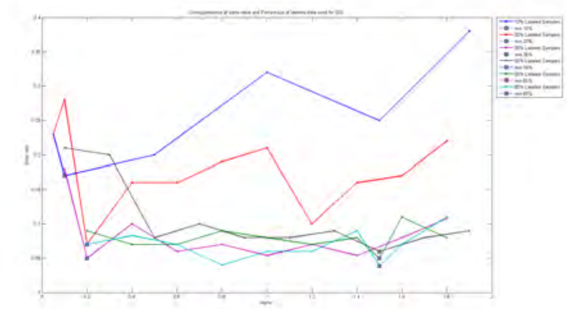


Fig. 4: Experimental results from the alpha tuning procedure

C. Real Data Supervised Results

For the real data experiments we are working in 2D environment, where given a image slice of the abdominal area the objective is segmentation of the kidneys, vertebra and aorta, which are visually distinguishable. Leave-one-out experiments are performed by taking all the images of the dataset except one for training and the remaining image for

[1] available at https://github.com/karpathy/Random-Forest-Matlab

testing. Note that all the images used in this section will have the same structure when it comes to arrangement of the sub-figures. For the training phase the 4 consequent images when going from left to right are: labelled samples used, Ground Truth, test images and results. For test phase there are only 3 sub-images: Ground Truths in the left, test images in the center and results in the right hand side.

In Fig. 5 we can observe the training and test phases when considering fully supervised segmentation, meaning 100% of points used are labelled. The result of the supervised segmentation on the test image in the figure above is mean error of 2.53%.

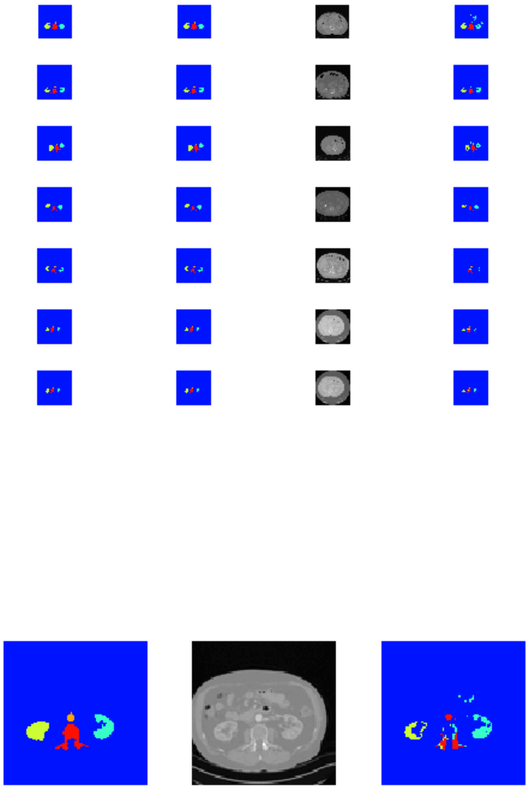


Fig. 5: Train results (top) and training results (bottom) of the supervised segmentation on real data

D. Real Data Semi-Supervised Results

We can talk of performing semi-supervised segmentation when only a percent of labelled data has been used for segmenting the images of interest. In the following experiments, if not stated otherwise, 80% of labelled samples have been used. Ideally, only very small percent should be used, e.g. 5% or even 1% of labelled data and ultimately this points should be no more than a dozen, i.e. handful of samples manually set by a radiologist or a surgeon.

There are multiple parameters whose role is pivotal in the best functioning of the semi-supervised learning algorithm designed for segmentation of the Computed Tomography (CT) images. These parameters and respectively their optimized values used across the following experiments are:

- alpha - 3000 - could be simply over-fitting but clearly alpha need to have value of a forth rank number
- tree depth - 15 was chosen. The deeper the trees the better the results are but also the depth underlines considerably longer training time.
- number of splits - 2000 which means effectively that 40 features from the feature vector are considered, and
- window size - 7x7 window was considered. Gives better result than when considering larger windows

After all of these parameters have been optimized there is only one more parameter to be examined which is the most important. That is the number of trees that the algorithm needs to produce as good as possible results. In Fig. 6 we can see the results when using different number of trees for training. One of the main points of using the semi-supervised approach of learning for segmentation of medical images is the effect of result improvement when increasing the number of labelled (supervised) samples. On the following figure (Fig. 7) we can observe the described effect. Starting from 10% and going through 50% and 90% and arriving to 100% which is the supervised case the error rate constantly improves. Results obtained from semi-supervised learning are comparative with high reliability to results produced by supervised learning, and represent more viable option for segmentation because semi-supervised segmentation can be performed on sparse Ground Truth images.

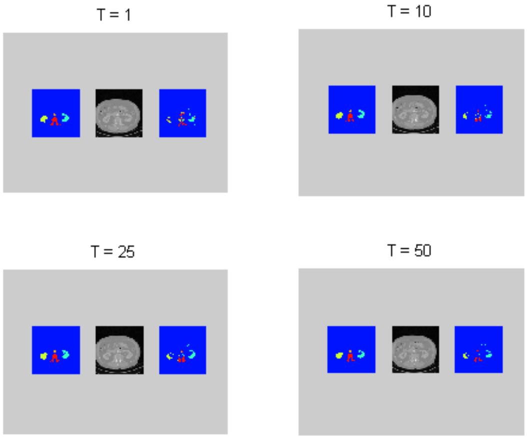


Fig. 6: Test results when using different number of trees T = number of trees

Another important property of using the semi supervised approach is the effect produced by varying the number of unlabelled (unsupervised) samples while using a small percent of supervised (labelled) data. The inspiration to

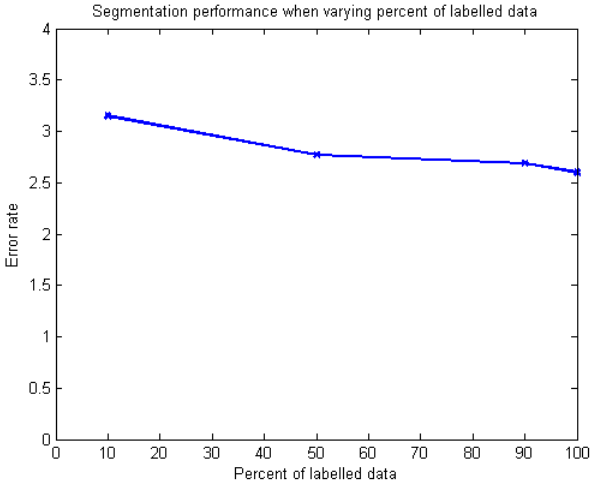


Fig. 7: Labelled data percent VS Segmentation Error

explore this option comes also from Criminisi's book [5], although there are no experiments directed towards proving this in the book.

In theory, by introducing new points (previously unseen), which is equivalent to unsupervised or unlabelled points the results should get better. In practice, this is not the case in statistical sense because of the sampling procedure which will be elaborated in broader manner when evaluating the results, but on the other hand the visual effect of the process of introducing unlabelled points provides the desired proof of the improvement in segmentation.

Fig. 8 contains the results of the experiment of this sort. Namely, only 10% of labelled pixels are used and from the remaining unlabelled pixels a varying portion from 0% to 100% is used. If this percent is less that 100 the unused unlabelled pixels are removed from the descriptor. By looking at the curve it is clear that the error fluctuates.

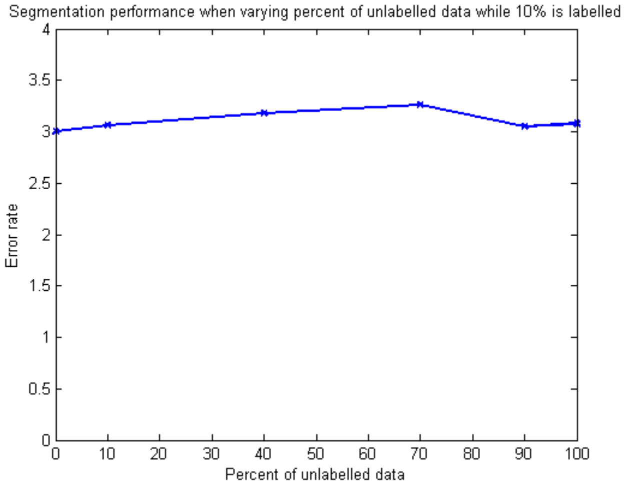


Fig. 8: Unlabelled data percent VS Segmentation Error

Ideally it should be a steady slope of error decrease. This is due to the randomness in sampling. For example, when taking 10% and 40% they sample different portions of the pixels, meaning the 40% do not contain the 10% sampled in the previous experiment plus other 30% but new 40% of pixels, most of which are probably not correlated with the previously sampled 10%. As proof of this is the sudden dip in the segmentation error rate when 90% of unlabelled points are used. But main evidence that using unlabelled points rather than not using them at all is obvious when looking at the terminal cases: 0% and 100% outputs.

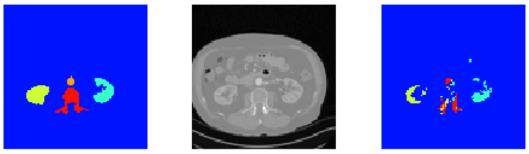


Fig. 9: Test result with 0% unlabelled points when 10% are labelled

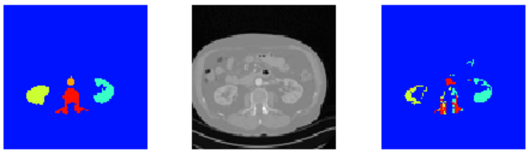


Fig. 10: Test result with 100% unlabelled points when 10% are labelled

Figures 9 and 10 contain the test results for the terminal cases: Although the error rate of the 0% case is smaller the result is far worse. Clearly by looking at the two images we can conclude that in the latter (100%) case all the organs are better segmented, i.e. more pixels corresponding to them are segmented. The organ contours are almost connected. Reasons for the greater error rate in the 100% case is the power of the *cheat feature* which forces greater number of the pixels of the vertebra to be segmented as pixels of the corresponding kidneys, as well as the increased phantom pixels segmentation. Phantom pixels refers to the pixels atop the left kidney (right in the image) belonging to another organ, probably the small bowel. The results obtained in the latter image can be more easily solved by introducing more sophisticated geometrical-based features or some post-processing techniques, such as morphological technique e.g. hole filling (dilation).

V. Discussion

With the help of the machine learning concepts, segmentation of real data has been achieved. By doing this proofs are given that not only the supervised or labelled data plays a role in the segmentation process but that the unsupervised or unlabelled data is also responsible for improvements when performing the segmentation task.
Semi-supervised segmentation can produce results comparable to fully supervised segmentation, while requiring sparsely annotated Ground Truth and considerably longer time, primarily because of performing label propagation. When increasing the amount of labelled data points used the confidence of the results increases. Same goes when increasing the number of unlabelled data points. This corroborates with other research on the same topic. Variations of the forest parameters affects results, both in terms of accuracy and time. There is always reciprocity between the accuracy and time aspects. Increase in accuracy underlines increase in time of execution, and vice versa.

VI. Conclusion and Future Work

A. Conclusion

The study presented in this paper was set up to explore the semi-supervised segmentation, as opposed to the supervised or self-learning techniques from which the notation of semi-supervised learning originated. After doing literature review on the existing approaches a suitable semi-supervised learning algorithm was selected - semi-supervised segmentation forest.
Difficulties in pinpointing appropriate slices and their extraction from CT datasets are one of the main limitations to obtaining more credible results. According to the justified anticipation synthetically created data is highly robust in term of results underlining majority of presented output to be based on real data. Here, it should be noted that big pixel intensity range and similar values additionally impede segmentation outputs. Optimization of forest parameters is instrumental in refining resultant figures.

B. Future Work

By accomplishing this study an evaluative perspective has been offered in increasingly important field in the Medical Image Analysis area. This has every ingredient needed to mature into reliable segmentation procedure on a larger scale, segmenting full 3D data scans instead of the 2D extracted slices. Concerning the aforementioned results improvements a post-processing techniques, such as morphological techniques can be used for enhancing the obtained results.
On a deeper level, improvements for semi-supervised learning may be achieved by optimizing of the existing weak learners such as the 2D linear learner, conic section learner or the distance learner. Another aspect where improvement can be tangible is the propagation of labels. Here instead of the implemented approximation, the actual minimization of the Geodesic distance function taking into consideration the shortest Geodesic path should be incorporated.
The noisy results of optimization of the forest parameters such as the tree number and the effect of increase of unlabelled data used when fixed number of sampled data is considered can be taken care of by refining the sampling procedure, or more specifically by taking seeds and growing trees additively taking percentages iteratively by preserving the previously sampled data across experiments. Finally, testing the whole procedure on larger training set would be ideal for obtaining more reliable results.

VII. Acknowledgments

I want to thank my supervisor Dr. I. Poole for giving me good guidelines in all aspects of the thesis preparation process. Also, I would like to express the deepest appreciation to A. O'Neil for her guidance and persistent help. Lastly, but not less importantly I would like to thank all the members of the Image Analysis Research Team in Toshiba Medical Visualization Systems Europe for their advice and comments given have been of great help through the whole time frame of my allocation.

References

[1] Aziah Ali, Rachel C. King, and Guang Zhong Yang, *Semi-supervised segmentation for activity recognition with multiple eigenspaces*, Proc. 5th Int. Workshop on Wearable and Implantable Body Sensor Networks, BSN2008, in conjunction with the 5th Int. Summer School and Symp. on Medical Devices and Biosensors, ISSS-MDBS 2008, 2008, pp. 314–317.

[2] Vijay Badrinarayanan, Ignas Budvytis, and Roberto Cipolla, *Semi-supervised video segmentation using tree structured graphical models*, IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013), 2751–2764.

[3] Jocelyne Bruand, Theodore Alexandrov, Srinivas Sistla, Maxence Wisztorski, C??line Meriaux, Michael Becker, Michel Salzet, Isabelle Fournier, Eduardo MacAgno, and Vineet Bafna, *AMASS: Algorithm for MSI analysis by semi-supervised segmentation*, Journal of Proteome Research **10** (2011), 4734–4743.

[4] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien, *Semi-Supervised Learning*.

[5] A Criminisi, J Shotton, and E Konukoglu, *Decision Forests for Classification , Regression , Density Estimation , Manifold Learning and Semi-Supervised Learning*, Learning **7** (2011), no. 2-3, 81–227.

[6] Antonio Criminisi, Toby Sharp, and Andrew Blake, *GeoS : Geodesic Image Segmentation*, (2008), 99–112.

[7] Feng Jiao, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans, *Semi-supervised conditional random fields for improved sequence segmentation and labeling*, Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL ACL 06 **44** (2006), 209–216.

[8] Kunlun Li, Juan Zhang, Hongyu Xu, Shangzong Luo, and Hexin Li, *A Semi-supervised Extreme Learning Machine Method Based on Co-training Review of Co-training*, **1** (2013), 207–214.

[9] Pavan Kumar Mallapragada, Rong Jin, Anil K. Jain, and Yi Liu, *SemiBoost: Boosting for semi-supervised learning*, IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009), 2000–2014.

[10] J. R. Quinlan, *Induction of decision trees*, Machine Learning **1** (1986), 81–106.

[11] P.K. Singh and P. Compton, *Evolution oriented semi-supervised approach for segmentation of medical images*, International Conference on Intelligent Sensing and Information Processing. (2004).

[12] Dirk Smeets, Dirk Loeckx, Bert Stijnen, Bart De Dobbelaer, Dirk Vandermeulen, and Paul Suetens, *Semi-automatic level set segmentation of liver tumors combining a spiral-scanning technique with supervised fuzzy pixel classification*, Medical Image Analysis **14** (2010), 13–20.

[13] Yangqiu Song, Changshui Zhang, Jianguo Lee, Fei Wang, Shiming Xiang, and Dan Zhang, *Semi-supervised discriminative classification with application to tumorous tissues segmentation of MR brain images*, Pattern Analysis and Applications **12** (2009), 99–115.

[14] Hang Su, Zhaozheng Yin, Seungil Huh, and Takeo Kanade, *Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features*, Medical Image Analysis **17** (2013), 746–765.

[15] Jun Suzuki, Jun Suzuki, Hideki Isozaki, and Hideki Isozaki, *Semi-Supervised Sequential Labeling and Segmentation Using Giga-Word Scale Unlabeled Data*, Proceedings of ACL-08: HLT, 2008, pp. 665–673.

[16] K. Tingelhoff, A. I. Moral, M. E. Kunkel, M. Rilk, I. Wagner, K. W G Eichhorn, F. M. Wahl, and F. Bootz, *Comparison between manual and semi-automatic segmentation of nasal cavity and paranasal sinuses from CT images*, Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings, 2007, pp. 5505–5508.

Shape from Template with a Dynamical Multi-Object Database

Luis Tobías
Image Science for Interventional Techniques lab, ISIT - UMR 6284 CNRS.
Université d'Auvergne, France
Supervised by: Adrien Bartoli

Abstract— **Current approaches to solve deformable 3D reconstruction from monocular views can be classified into *template-free* that explore the use of a set of images containing distinctive deformations to model at the same time both the 3D position of the object and camera motion, and *template-based* techniques that depend on a supporting image with a known 3D model (so called template) to carry out the reconstruction from an additional image. Both of the aforesaid methods suffer from limitations inherent to their nature. Moreover, by using an object database it is possible to mitigate some of the drawbacks of both template-free and template-based methods. In this paper we explore the possibility of dynamically expand a database with a collaborative model that allows user interaction. In other hand we use feature-based object detection and explore a color-based approach to bootstrap the performance and alleviate the computational burden of the matching process.**

I. INTRODUCTION

Monocular non-rigid 3D reconstruction has been extensively studied in the recent years [6], [7], [25]. Despite this fact, it still lies behind rigid 3D reconstruction, an area where several techniques have already reached a maturity that make them robust and reliable. Well-developed techniques such as Structure-from-Motion (*SfM*), Shape-from-Shading (*SfS*), Shape-from-Focus (*SfF*), and other Shape-from-X techniques are already available in commercial and scientific domains. It is now possible to make 3D reconstruction of a rigid object using a mobile phone [27].

Current approaches to solve deformable 3D reconstruction from monocular views can be classified into *template-free* [10], [30] that explore the use of a set of images containing distinctive deformations to model at the same time both the 3D position of the object and camera motion, and *template-based* techniques that depend on a supporting image with a known 3D model (so called template) to carry out the reconstruction from an additional image.

The mentioned earlier methods suffer from limitations inherent to their nature. Moreover, it has been prove that by using an object database it is possible to mitigate some of the drawbacks of both template-free and template-based methods [2].

We propose to complement the idea presented in the paper "Deformable 3D Reconstruction with an Object Database" [2], where a framework for object recognition and 3D reconstruction was proposed.

The main objectives for this project are: the implementation of a module that allows the template creation at runtime, a collaborative model where templates can be fetched from the internet, and finally to enhance the object detection by exploring the color distribution of the objects in the database.

With functionalities such as the addition and retrieval of new image templates, the database dynamically expands. This expansion cannot be controlled and naturally, at a certain point the database could become massive. Using a wide baseline image matching approach when a database is considerably large is not feasible for a real-time application. Several problems can be encountered in such a model, being one of the most critical the computational resources needed to perform matching against a high number of images. Hence, a set of solutions to reduce the computational burden of performing this operation are needed.

We propose to address this issue by creating a color signature for each template in the database. This signature is a Probability Density Function (*PDF*) modelled by a Gaussian Mixture Model (*GMM*). The *GMM's* are later used to evaluate pixels in the images from a video stream to obtain a probability map. The map is processed taking image blocks of different sizes to compute a score τ that is later evaluated against a threshold t. We conclude that the object is present in the image region if $\tau > t$.

In the later stages our object detection method relies on detecting and matching salient features between the images, using a light weight descriptor. We propose to speed up the matching process by reducing the search area to particular image locations where the objects were probably found with the aforementioned color based detector. The matches are then refined using a modified version of RANSAC [28] adequate to work with deformable models. The set of inliers is then used to estimate a Radial Basis Function (*RBF*) image warp by solving a linear system for the Thin-Plate Spline (*TPS*). Finally, given the estimated warps we perform deformable 3D reconstruction for the detected objects and display the 3D model embedded in the original image.

The rest of this document is structured as follows. §II provides an overview of the background. §III covers the design of the implemented system. In §IV the outputs of this project are presented and a discussion about the system and its drawbacks is stated. In §V a summary of contributions is presented while comparing the original objectives and the actual status of the project. In this section future work is proposed.

II. BACKGROUND

Template-free 3D deformable reconstruction methods infer the 3D model of a deformable object from a sequence of 2D projections across time in a sequence of images. The principal requirement for this methods is to have feature tracks with sufficient baseline between the images. These methods make use of spatio-temporal smoothness priors to limit the problem.

The first approach was proposed by Bregler [10], whose influential work is being followed in multiple publications such as [11] that takes the assumption that some of the features are deforming throughout the sequences while others remain rigid. In [21] an incremental approach is proposed to estimate deformable models. The trajectory space approach is used in [1] where the evolving 3D structure is represented in a trajectory space by means of a linear combination of basis trajectories. In [19] is proposed a Bayesian Finite Element Method (*FEM*) modelling of deformations integrated within an Extended Kalman Filter (*EKF*) framework.

Template-based deformable 3D reconstruction methods require one single image, since they rely on a prior template where the 3D shape of the object at rest is known. These methods have two main steps: (i) registration of the input image to the template and (ii) 3D deformable reconstruction from reprojection and deformation constraints. Some works [22], [24] proposed convex formulations considering inextensible deformation constraints maximizing the surfaces depth. In [7] analytical solutions for the isometric and conformal deformation cases were proposed. They showed that template-based isometric surface reconstruction from a single view registered to a template, generally has a single solution for both developable and non-developable surfaces.

In [2] an new concept is introduced, performing deformable 3D reconstruction from monocular view in a novel approach. Using object detection and multiple templates the authors are able to reconstruct different objects from a single image. This proposal breaks with the main limitations of both template-free and template-based methods: there is no need to have multiple images encoding the deformation of the object as required in template-free methods and multiple objects can be reconstructed at the same time contrary to the template based methods.

The key process is the object recognition framework that allows the selection of the right templates to obtain deformable 3D reconstruction using Shape from Template (*SfT*). Multiple templates are stored in a database that is built offline. It contains not only appearance descriptors as traditionally encoded recognition databases, but also specific data about the material properties in order to ease deformable 3D reconstruction.

III. METHODOLOGY

We keep a simplified version of the system proposed in [2], contrary to this work we do not use a hierarchical three vocabulary tree, but instead we simply perform wide-baseline image matching. The new model is distributed in three main blocks with specific tasks. The first module is in charge of loading the database, computing descriptors and generating the *GMM's* used in the object detection stage. In the second module the object detection takes place, in addition the 3D reconstruction by means of SfT is conducted and the display of the 2D and 3D representations of the object is presented. In the last module the interactive template creation and addition of new objects to the database is executed. To provide an overview about how these blocks are connected a state machine diagram of the system is presented in figure 1. In this diagram the four transitions between components that can be can be executed are:

- The initialization module, linked directly to the recognition mode.
- From recognition mode to template creation mode.
- From template creation mode to recognition mode.
- The exit signal that can be called from both recognition and creation modules modes.

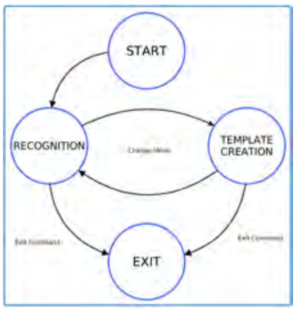


Fig. 1: State machine of the system: four principal transitions connect the three modules of the system.

A. Initialization

In the initialization module all the images in the database are loaded with their respective 3D meshes. For each image in the database we detect and extract features using A-KAZE [4] and the centres for the warp estimation are computed. The camera is initialized and calibrated.

Parametric models of the probability density function of each image are extracted. These *PDF's* are represented by *GMM's* with K clusters, characterized by their means μ_i and covariance matrices Σ_i, and weights ϕ_i:

$$p(\theta) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\mu_\mathbf{i}, \mathbf{\Sigma_i}) \tag{1}$$

Where $\mathcal{N}$ is a normal Gaussian probability density function.

We set the covariance matrices to be diagonal $\Sigma = \sigma^2 I$, assuming independence between the image color components, which is not exactly the case but allows us to avoid a costly matrix inversion with a small loss in accuracy. The *GMM's* are obtained using the expectation-maximization algorithm *EM* as described in [9].

As a final step important variables such as the parameters for the object detection, outlier rejection and 3D reconstruction are set. A 3D rendering context is initialized.

B. Object detection and 3D reconstruction

In the application workflow a sequential process is established, working in such a way that the immediate next step is performed only if the current stage has been successfully accomplished. Otherwise the further steps are not executed.

1) Color based object detection: The *GMM's* obtained in section III-A are used to evaluate patches from the query image, using a sliding window approach, a common technique in object detection [12], [14], [31] that exhaustively scans the image at multiple locations and scales. Our sliding window process works with different window sizes within the range of 30 to 80% of the total image size, and a fixed step equivalent to 10%.

First we evaluate the *GMM* model for all pixels in the image obtaining a probability map. Then the sliding window process takes place. We analyse each subimage window, first converting the image patch w in a row-vector R_w, then sorting it in ascending order. Let n the length of R_w, $z = round(0.7 \times n)$ and $\tau = R_w(z)$ If the probability value τ is greater than certain threshold t we use all the values that are greater than τ to compute a probability score:

$P_{R_w} = R_w(z) + R_w(z+1) \cdots + R_w(n)$.

If $\tau < t$ the patch has a score of zero. The probability score P_{R_w} serve as measurement of how likely is the object to be present in the image, we store the coordinates and the window size of all the evaluated windows.

2) Feature based object detection: In the second stage of the object detection a wide-baseline feature matching approach is used. As mentioned before, in the initialization process features are detected and extracted from all the images in the database. At runtime we obtain descriptors from the query image that we proceed to match with the descriptors from each of the images in the database.

Since the matching method follows the *Brute Force* and the nearest neighbour distance ratio matching strategy, the process requires a big amount of computational resources. In addition the complexity of this process increases as the database grows. Moreover, taking advantage of the previously computed probability scores we can reduce the search area and try to match descriptors only in certain regions of interest. In some cases we could discard images in the database that do not contain the object.

If the probability of the object being in the image is greater than a threshold T_c we proceed to match the descriptors, otherwise we abort the process and check for the next object in the database. The number of matches is then analysed, and the outlier rejection algorithm is performed only if the number of putative matches *nMatches* is greater than a threshold T_m. It is worth mentioning that without a proper matching refinement the warp cannot be correctly estimated dor correctly interpolated. Due to this we must avoid outliers in order to generate a good warp. A variant of the RANSAC method [28] that aims to fit a 2D affine subspace onto an observation set is used. This is due to the fact that the projection of the first-3 principal components of the correct matches falls in a 2D affine hyperplane *relative* to the thick error of the outliers.

An additional step to enable the opportunity of reconstructing multiple instances of the same object is to do not mark the correspondent template as already found, but remove the keypoints that have been successfully identified as an object.

3) 3D reconstruction and rendering: Following the same fashion, the 3D reconstruction process is attempted only if the number of inliers is beyond a threshold T_i. First is needed to find an image warp. This is achieved using two sets of true matches *inliersT* and *inliersQ*. The warp encodes the particular deformation of an object in the image [7]

We proceed to interpolate the mesh points and find the mesh point derivatives δ_x, δ_{xx}, δ_y, δ_{yy} and δ_{xy}. These derivatives are later used to find the solutions ϕ to the *PDE's* in *SfT*. The solutions provided are the 3D point coordinates of the object.

The next step is to render the obtained 3D model. We obtain the surface normals and the model is displayed using the OpenGL library in a 3D drawing context while a 2D projection of the grid is drawn in the 2D query image. The flow diagram that summarizes the pipeline of this component of the software application can be observed in figure **??**. The obtained model can be used to perform augmented reality. Moreover, in our case we simply embed the resultant 3D model over the original image.

C. Interactive template creation

The application provides a simple user interface that is designed in such a way that by pressing a key the user can change the mode, going from the Recognition mode to the Template creation mode, see figure 1. This allows the interaction with the application to create new templates.

The user must flatten the object and position it in an horizontal direction, ten by using the cursor selects four points correspondent to the object's corners.

The corner points are needed to find the homography that maps the template into a rectangular image with known size. A perspective transformation is then applied and the template is rectified. A 3D mesh is created by placing a regular grid with a size equivalent to 10 % of the image size. Since the object is being flattened the depth coordinate is fixed to a constant value.

The result is displayed and the user is asked if the template should be saved or discarded. In case the user agrees to save the template, the image and its corresponding 2D-3D template are stored in the disk.

1) Online database construction: Once the new template is created and saved the user can add it to the database on the fly. Being able to obtain the 3D reconstruction of the new object right after the creation, without completely stopping the application. This step is one of the main differences with the original system where the database was built offline.

To add the template to the database online, the same set of actions that have been done in the initialization process must be performed, for example, obtain the corresponding *GMM*, find the image keypoints, extract descriptors and finally load in memory both the image and mesh.

2) Collaborative data model: We propose to have collaboration among application users. When a new template is created the user can upload it to a server where it will be stored and will remain available to other users. Similarly the application provides the option for connection to the server and download templates from other users.

IV. RESULTS

The implementation of the application is written in *c++*, using the *OpenCV, OpenGL* and *OMP* libraries. The application runs under *Linux* in a laptop computer with a core *i5* 2.3 GHz processor and 4 Gb of RAM memory.

A. Application performance

Two databases are used in our experiments, both initially comprises 3 different objects. We used typical objects made of paper (developable surfaces). The images in the database have a resolution of 640 x 480 pixels. In this experiment the computational times of each step of the pipeline are measured. First we measure the performance using the dataset shown in figure 2, then the same process is repeated with the second dataset in figure 3. To obtain significant data each measurement have been repeated 50 times and the results displayed in tables I and II are the average times.

The time employed in each step can vary depending on several factors such as the number of descriptors detected in the query image and the number of objects in the database. Thus, a highly textured object will increase the computational costs but will lead to a more accurate reconstruction. In addition, the size of the object in the image change according to the depth, having an object of bigger size in the image will generate a bigger number of descriptors. Due to this more (or less) features need to be matched. What directly modifies the time invested time for the matching and further stages.

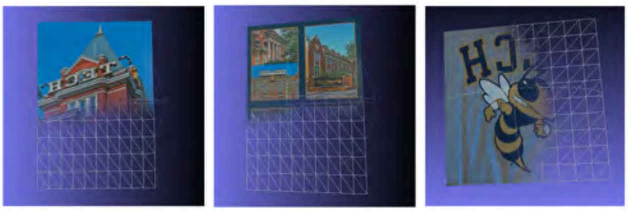


Fig. 2: Object dataset 1: templates used in experiments whose results are shown in table I. In this image the 3D representation of the templates in a flat position is displayed. The images are arranged to show at the same time both the 3D mesh gird and a portion of the texture mapping.

B. Outlier rejection

As mentioned in III-B we use a RANSAC variant that considers deformable surfaces to detect and prune spurious correspondences. The input of the algorithm is a set of matches obtained between the target image and the 2D parameterization of the 3D template. The output is a set of true matches, inliers. For some images the output of the algorithm is presented in figure 4.

Process	1 Object	2 Objects	3 Objects
Undistortion	0.00043	0.00043	0.00043
AKAZE Query	73.04191	84.97628	85.91019
Matching	152.4091	218.52901	234.46101
RANSAC	7.94681	6.96366	5.82059
TPS Warp	0.14787	0.77599	0.45229
TPS Warp Derivatives	5.49202	0.34131	0.49661
Deformable 3D	0.02006	0.03936	0.05811
Normals	0.00808	0.01521	0.02302
Total	**239.07**	**311.64**	**327.22**
AKAZE Template *	*379.501*		

TABLE I: Average computational time required to perform the entire reconstruction pipeline different numbers of objects present in the image. Using a database consisting of 3 templates at resolution of 640 x 480 pixels. Time expressed in milliseconds.

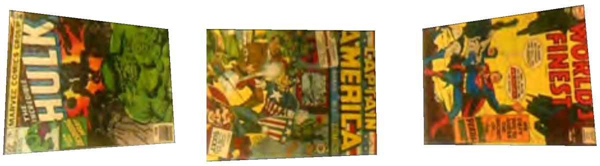
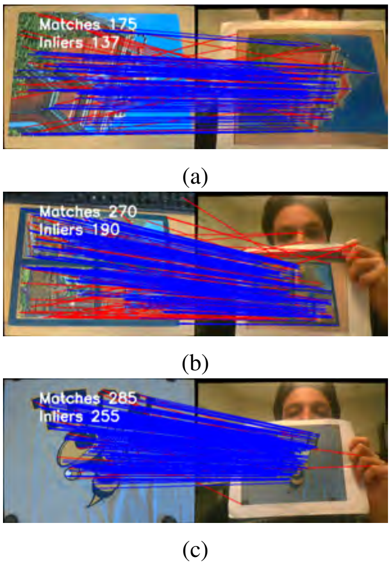


Fig. 3: Object dataset 2: templates used in experiments whose results are shown in table II. Images from this dataset differ from the first dataset in the amount of texture present.



(a)

(b)

(c)

Fig. 4: Results of the outlier rejection method for image matching in deformable surfaces. A set of matches is computed between the target image and the template image using a modified version of RANSAC adapted to work with deformable surface models. The algorithm returns a) 137 true matches from a set of 175 correspondences b) 190 true matches from a set of 270 correspondences c) 255 true matches from a set of 285 correspondences. Blue lines depict true matches and red represent spurious matches.

C. 3D Reconstruction

Deformable 3D reconstruction of a template from a monocular view is achieved using a 3D template and a

Process	1 Object	2 Objects	3 Objects
Undistortion	0.00043	0.00043	0.00043
AKAZE Query	93.41091	120.78618	185.14019
Matching	198.2921	310.19501	438.10461
RANSAC	12.94168	11.39666	12.04759
TPS Warp	0.94709	1.07599	1.45349
TPS Warp Derivatives	1.49202	0.34131	0.49661
Deformable 3D	0.03001	0.05932	0.07824
Normals	0.01208	0.01821	0.02502
Total	**307.13**	**443.87**	**637.35**
AKAZE Template *	*645.202*		

TABLE II: Average computational time required to perform the entire reconstruction pipeline different numbers of objects present in the image. Using a database consisting of 3 templates at a resolution of 640 x 480 pixels. Time expressed in milliseconds.

2D representation of this template. Once a set of clean-up matches is computed an image warp is estimated and then used in the reconstruction by *SfT*. The reconstruction of a developable object surface is exemplified in figure 5.

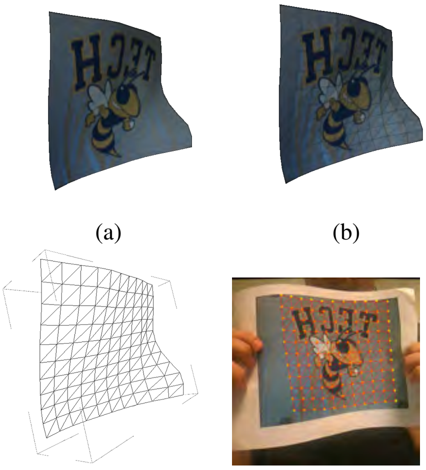

(a) (b) (c) (d)

Fig. 5: 3D Reconstruction from a monocular view. The reconstructed object belong to the dataset presented in 2. a) texture mapping, b) texture + 3D mesh grid, c) 3D mesh grid. d) 2D Original image + 2D grid representation.

D. Online template creation

One of the modules created for this project allows the online creation of new templates for the database, the outline of the creation process described in Section III is illustrated in figure 6, where the steps required to add a new template to the database are presented:

- Flattening of an object of know size
- Acquisition of an image
- Selection of the 4 corners of the object

E. Color based object detection

We have tried to implement a color-based object detection, this step has as principal task to alleviate the computational burden of matching one image with all other images from the database. Hence, detecting objects position in a fast way will reduce significantly the resources needed to match the images since the sear area decreases and thus the number of features to match become smaller.

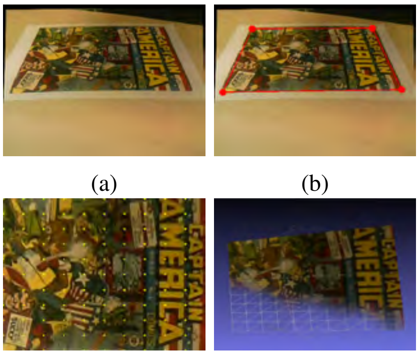

(a) (b) (c) (d)

Fig. 6: Online template creation example a) image of a flatten object, b) final point selection, d) rectified image with a set of overlapped mesh points, e) 3D mesh with its texture mapping.

However, the implementation of the color based object detection did not perform as expected in terms of computational speed gain. In fact the performance decreased drastically from having 2-3 frames per second to spend several seconds to process a single frame. This is due mainly to the computational load generated in the evaluation step, where a series of matrix multiplications and exponentiations are needed to obtain the predicted value. In addition, the number of Gaussians used to model the *PDF* has a strong impact in this process. Despite the fact that the performance was not improved with the implementation of color based detection the algorithm shows potential to correctly detect objects. Some result are shown in figure 7 where an example of a positive detection is presented. In §V we provide some alternatives to improve the detection performance and suggest future work in this particular topic.

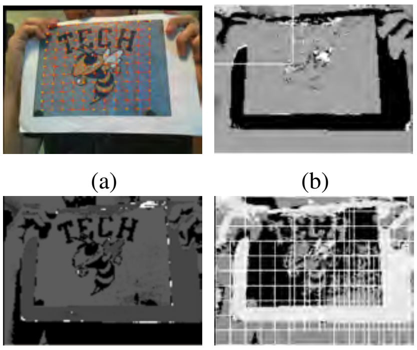
V. CONCLUSIONS

In this section the summary of the project outputs is presented.

A. Summary of contributions

In this project the principal goals were set in §I, the following list summarises the outputs:

1) We implemented an online template creation module, contrary to the original system our software application allows the creation of new templates without stopping the application. The database dynamically expands once a template is created. Allowing to directly perform 3D reconstruction from recent templates.
2) Collaborative data sharing, promoting the collaboration among users the system can share templates, connect to internet and fetch content. The system can acquire new templates from internet.



(a) (b)

(c) (d)

Fig. 7: Color based detector using *GMM's*. The template evaluated corresponds to the third object in the database. Detections are shown as white bounding boxes. a) original image, b) probability of being template 1, c) probability of being template 2, d) probability of being template 3. We can observe a false positive detection in the left top corner of b), true negative detection in c) and multiple true positive detections in d).

3) Creating a color-based object detector. Currently working to improve the detection time and solve related issues.

B. Future work

1) Color-based detector: In §IV the status of the color based detection and its limitations are presented. We propose some ideas to improve performance of the detector in order to achieve real-time detection.

First by changing the colorspace to a more *"human"* like colorspace such L*a*b* will allow to separate the luminance channel. This will create a direct impact in the algorithm since it will make it robust to illumination changes and at the same time will help to reduce the dimensionality of the data.

Second, pre-computing probability Look Up Tables (*LUT*) when a new template is created is a sensible thing to do, since scanning an image with a *LUT* for a single channel image at a resolution of 2560 X 1600 pixels takes on average 17 milliseconds, which will represent a great advantage against the current implementation. A last possible action to speed up the process is to quantise the color distribution in bins, this will reduce even more the data dimensionality with a certain trade of between speed and accuracy.

Another important point is the automatic selection of the threshold value τ, which could be obtained by evaluating the same image that has been used for create the GMM model then test different values and analyse the behaviour of the evaluation. A cross validation steep will be thus needed. And at last, the output from the detector is not refined, multiple windows overlap. A stop criteria must be set in order to stop the detector if the object is already found in a large window avoiding to spend time in redundant information.

2) General aspects: The application uses currently the *GUI* capabilities from the opencv library, enough for a prototype. Moreover a further refinement (or re-implementation) of the *GUI* is definitely needed to make the application more user-friendly and intuitive. A possible option is to use Qt *GUI* creator.

In §III it is mentioned that the user can create new templates, but, if the object cannot be flattened, has a unknown size or is not a developable surface then our basic template creation module fails to deal with the creation process. Thus, another technique must be employed for the template creation. We propose to use a Structure from Motion *SfM* module and to integrate it in the application.

VI. ACKNOWLEDGMENTS

The author gratefully acknowledge the supervision of Adrien Bartoli. This work has been realized in the Advanced Laparoscopy and Computer Vision group. Member of the Image Science for Interventional Techniques lab, ISIT - UMR 6284 CNRS. Université d'Auvergne, France.

REFERENCES

[1] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1442–1456, 2011.

[2] Pablo F. Alcantarilla and A. Bartoli. Deformable 3d reconstruction with an object database. In *British Machine Vision Conf. (BMVC)*, 2012.

[3] Pablo F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE features. In *Eur. Conf. on Computer Vision (ECCV)*, 2012.

[4] Pablo F. Alcantarilla, J. Nuevo, and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *British Machine Vision Conf. (BMVC)*, 2013.

[5] A Bartoli, M Perriollat, and S Chambon. Generalized thin-plate spline warps. In *CVPR'07 – ieee Int'l Conf. on Computer Vision and Pattern Recognition*, Minneapolis, USA, jun 2007.

[6] Adrien Bartoli and Toby Collins. Template-based isometric deformable 3d reconstruction with sampling-based focal length self-calibration. In *CVPR*, pages 1514–1521. IEEE, 2013.

[7] Adrien Bartoli, Y. Gerard, F. Chadebecq, and Toby Collins. On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In *CVPR*, pages 2026–2033. IEEE, 2012.

[8] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

[9] Jeff Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute and Computer Science Division, University of California at Berkeley, 1998.

[10] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. pages 690–696, 1999.

[11] Alessio Del Bue, Xavier Lladó, and Lourdes de Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *CVPR (1)*, pages 1191–1198, 2006.

[12] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[13] Rui Pimentel de Figueiredo, Plinio Moreno, Alexandre Bernardino, and Jos Santos-Victor. Multi-object detection and pose estimation in 3d point clouds: A fast grid-based bayesian filter. In *ICRA*, pages 4250–4255. IEEE, 2013.

[14] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[15] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[16] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[17] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[18] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[19] J. M. M. Montiel, B. Calvo, and A. Agudo. Finite element based sequential bayesian non-rigid structure from motion. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:1418–1425, 2012.

[20] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

[21] Marco Paladini, Adrien Bartoli, and Lourdes Agapito. Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 15–28. Springer, 2010.

[22] Mathieu Perriollat, Richard I. Hartley, and Adrien Bartoli. Monocular template-based reconstruction of inextensible surfaces. In Mark Everingham, Chris J. Needham, and Roberto Fraile, editors, *BMVC*. British Machine Vision Association, 2008.

[23] Daniel Pizarro and Adrien Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.

[24] Mathieu Salzmann and Pascal Fua. Reconstructing sharply folding surfaces: A convex formulation. In *CVPR*, pages 1054–1061. IEEE, 2009.

[25] Appu Shaji, Aydin Varol, Lorenzo Torresani, and Pascal Fua. Simultaneous point matching and 3d deformable surface reconstruction. In *CVPR*, pages 1221–1228. IEEE, 2010.

[26] Bastian Steder, Giorgio Grisetti, Mark Van Loock, and Wolfram Burgard. Robust on-line model-based object detection from range images. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS'09, pages 4739–4744, Piscataway, NJ, USA, 2009. IEEE Press.

[27] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3D reconstruction on mobile phones. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[28] Quoc-Huy Tran, Tat-Jun Chin, Gustavo Carneiro, Michael S. Brown, and David Suter. In defence of ransac for outlier rejection in deformable registration. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV (4)*, volume 7575 of *Lecture Notes in Computer Science*, pages 274–287. Springer, 2012.

[29] Hideaki Uchiyama, Inria Rennes Bretagne-atlantique, Eric Marchand, Universit De Rennes, and Inria Rennes Bretagne-atlantique. Toward augmenting everything: Detecting and tracking geometrical features on planar objects. In *in "IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR11*, pages 17–25, 2011.

[30] Jing Xiao and Takeo Kanade. Uncalibrated perspective reconstruction of deformable structures. In *Tenth IEEE International Conference on Computer Vision (ICCV '05)*, volume 2, pages 1075 – 1082, October 2005.

[31] Junge Zhang, Kaiqi Huang, Yinan Yu, and Tieniu Tan. Boosted local structured hog-lbp for object localization. In *CVPR*, pages 1393–1400. IEEE, 2011.