# Regions detection and tracking for visual servoing using Ultrasound images

Mohammad Alkhatib

*Abstract*— **In the last decade, Ultrasound-Guided Regional Anesthesia (UGRA) gained importance in surgical operations, due to its efficiency of localizing the nerves. Using robotics guidance UGRA, the procedure can be divided into three major tasks, detecting the regions of interest in the US images, tracking these regions in each frame, and using visual servoing to guide the robot. For the detection task, we used an existing fully-automatic detecting approach which applies machine learning technique to detect the median nerve in US images. For tracking task, we propose a new and robust tracking technique by using well-known tracking algorithms (particle filter, mean-shift and Kanade-Lucas-Tomasi(KLT)) combined with Adaptive Median Binary Pattern(AMBP) as texture feature. In this paper, Kalman filter is used as an option to enhance the tracking accuracy, computational cost and to handle disappearance of the target. Using Kalman filter along with the tracking algorithms started by using Kalman filter prediction step to predict the new location, followed by applying the tracking algorithms around the predicted location, then Kalman filter correction step is applied to find the final nerve location. In this paper, we compare between AMBP and different kinds of texture feature extraction methods. The tracking algorithms had been evaluated with and without image filtering procedure. AMBP with particle filter outperforms other methods in both cases filtered and not filtered images.**

## I. INTRODUCTION

Regional anesthesia (RA) performed to block the sensation of pain in specific region of the body by stopping the nerve impulses connection between that region and the central nervous system. Nowadays RA is a well-known procedure in many operation room. Traditionally, RA performed with a blind guidance by relying landmarks on the surface, these landmarks guide the needle insertion operation. The anesthetists perform RA by injecting anesthetic product near the nerve. For that RA requires high experience anesthetists to avoid many risks such as, block failure, local anesthetic toxicity, nerve trauma and neurological and vascular injuries [1]. Ultrasound (US) technique provides good alternative, due to low cost, no radiation, real-time acquisitions and portability properties. Therefore, US is the most frequent technique for RA operations, called Ultrasound-Guided regional anesthesia (UGRA). US allows anatomical structure visualization, which improves the success of the RA operation, and reduce the risk of the RA failure.

Unfortunately, US implies high demand on the operator since the examination require mastery in controlling the US probe. Therefore, robotics systems were developed to control the US probe, to bring assistant to medical staff [2]. To control the robot several solutions based on visual servoing were proposed that depends on extracting information from the image. The objectives of visual servoing are maintaining the visibility of the needle and the target, compensating the patient motion in a way that stabilize the US images on the target, and automatic positioning the probe in a way to retrieve preoperative image. The key difficulties of URGA is to visualize the target region and the insertion needle, and to localize them in the US images. US image quality makes the detection, tracking and visual servoing tasks very challenging, due to the degradation on the visual property of US.

Controlling US probe using a robot arm to keep the visibility of a specific region can be divided into three major tasks [3] [4]. First task is detecting the region of interest (ROI) automatically in the first frame. In this task, US image should be analyzed and enhanced, then by using classifier the ROI should be detected. The second task is tracking the ROI in each frame. Unique areas in the ROI used with tracking algorithms to accomplish this task. The last task is to use visual servoing techniques that control the robot arm to maintain the visibility of the ROI.

The structure of our paper is as follow. In Section II, we present related work for detection and tracking tasks. Section III discusses the proposed tracking technique. Section IV shows the results of the experiments, followed by discussion in Section V. Final conclusions are shown in Section VI.

## II. RELATED WORK

Noble and Boukerroui [5] proposed region segmentation in the US image, where they showed that the accuracy of the detection phase of a specific region in the image depends on three important steps. The first one is US image enhancement. The second step is feature extraction and selection. Here in this step, a suitable feature should be extracted from the US images, and from these features, few of them should be selected to increase the performance of the classifier. The last step is classification. Image enhancement is an important step in the detection phase which is divided into two parts. The preprocessing step of the US image where the US image is segmented into foreground and background regions, and removing the noise in the US images. US images is effected by the speckle noise which reduce the visual quality of the image and it is important to remove these effects. There are several studies discuss different types of despekle filters [6] [7] [8] [9]. There are several studies

Mohammad Alkhatib is with University of Borgogne, France, Mohammad_Alkhatib@etu.u-bourgogne.fr

This work is supervised by Prof. Adel Hafiane and Prof. Omar Tahri, INSA CVL, Universit dOrlans, Laboratoire PRISME EA 4229, Bourges, France, {adel.hafiane, omar.tahri}@insa-cvl.fr

addresses classification in the medical field, but few studies focused on the regional anesthesia problem. In this project, we used the implementation of [10], where Hadjerci et al. used statistical features with SVM to classify the median nerve in the US images.

Tracking is one of the most popular tasks in computer vision, and it is used in many applications such as video compression, medical imaging and robotics. Tracking failure could happen easily under the noise, illumination changes, occlusion, and deformation of the target. Indeed, tracking regions of interest (ROIs) in US imaging modality is very challenging task. Therefore, it is important to use a robust and effective target tracking algorithm. Various methods have been proposed in the literature to address some of these problems. Among tracking algorithms, there are three simple, robust and efficient tracking algorithms, particle filter [11], mean shift [12] and Kanade-Lucas-Tomasi(KLT) [13]. Particle filter and mean shift algorithms used to find the similarity between current image and the target model, and it is based on histogram representations. KLT used frame to frame object tracking, which consists of matching object feature points or descriptors between these images and calculate the displacement vector. In this paper, we proposed a new efficient feature tracker combined with tracking algorithms.

Histogram representations are popular in visual object tracking because of their effectiveness and efficiency in capturing the distribution characteristics of visual features inside the object regions. These representations are not sensitive to the deformation and partial occlusion, but it is sensitive to illumination changes. In order to reduce these effects, a better representation had been introduced by using the texture properties of the target. Texture is an important cue for many US images applications, such as detection and tracking. To improve the efficiency of the tracking, joint color-texture histogram provides a powerful descriptors and more reliable information of the target. Not all the texture methods can be combined with the histogram, like gray-level co-occurrence matrix and Gabor filtering. One of the best texture analysis methods is Local Binary pattern (LBP) due to fast computation and rotation invariance [14]. In [15], Ning et al. used LBP for describing color-texture histogram with mean-shift algorithm. In [16], the author improved paricle filter to handle illumination changes by using LBP as a color-texture histogram. Histograms of Oriented Gradients (HOG) is also a good texture analysis method, in [17], Bilinski et al. performed multi object tracking under occlusion using HOG descriptors.

The previous mentioned texture-based methods have some notable limitations, mostly the sensitivity to noise. Many studies aim to increase the robustness of noisy textures classification. In [18], the author used Median Binary Pattern(MBP) and Gabor filter as texture descriptor to classify and characterize nerves in US images. Liu et al. [19] proposed Binary Rotation Invariant and Noise Tolerant (BRINT) which is a highly effective multi-resolution descriptor for noisy texture classification. Hafiane et al [20] introduced a new texture operator, Adaptive Median Binary Pattern(AMBP) which uses global joint information for robust texture classification. In this paper, we introduce a new feature tracking technique by using AMBP as a target representation to track median nerve in noisy texture US images.

## III. The propose tracking technique

For tracking ROI, first unique areas in the ROI should be extracted. After that, tracking algorithms are used to track these areas. The structure of this section as follow, first we present different kind of texture feature extraction methods. Followed by discussing feature tracking algorithms.

### A. Feature extraction

This section provides seven feature extraction methods, that will be used to determine which one is better for nerve tracking task.

*1) Local Binary Pattern :* The Local Binary Pattern (LBP) is one of the powerful operators for describing image texture features [14]. The simplest, yet very efficient, LBP feature vector can be created by two steps. first, compare a P neighbors of a central pixel to get the corresponding P-bit binary code with regard to the center pixel's gray value as a threshold. Then, the image LBP histogram is computed. To get a coherent descriptor, Normalized histogram should be obtained in case of comparing two different size images.

*2) Compound Local Binary Pattern:* Compound Local Binary Pattern (CLBP) is an extension of the LBP texture [21]. The CLBP operator exploits 2P-bits binary code where the first P-bits is the same as LBP and the second P-bits express the magnitude information of the differences between the center and the neighbor gray values.

*3) Median Binary Pattern:* Median Binary Pattern (MBP) demonstrate very good discrimination properties, and used to provide more sensitivity to micro-structure and noise robustness [22] [23]. MBP operates by comparing a P neighbors and the central pixel to get the corresponding P-bit binary code with regard to their median gray value as a threshold.

*4) Adaptive Median Binary Pattern:* Adaptive Median Binary Pattern (AMBP) is an adaptive approach that uses either LBP or MBP to obtain better threshold depending on the local context [20], and considers image local context variation as an additional information. AMBP uses self-adaptive analysis window size which made it more robust to noise. AMBP joint histogram combines both LBP and MBP depending on the noise and the microstructure information.

*5) Joint Adaptive Median Binary Pattern:* Joint Adaptive Median Binary Pattern (JAMBP) is a new texture classification approach which associates the joint information based on adaptive local analysis [24]. JAMBP starts with representing the levels with multiscale space images. For each image AMBP is computed with different neighborhood variation, then all histograms are concatenated into one.

*6) Histogram of Oriented Gradients:* Histogram of Oriented Gradients (HOG) is used to describe the distribution of intensity gradients or edge directions [25]. HOG steps start with computing the horizontal and vertical centered

gradients, and computing the gradient orientations and magnitudes. the second step is to divide the image into 16x16 blocks with 50% overlap, each block contains 2x2 cells with 8x8 pixels. The third step is to use tri-linear interpolation to quantize the gradient orientation into 9 bins for each block. The last step is to concatenate the histograms.

*7) Historgram Representation:* Using histogram [26] to describe the color distribution does not take into account that two images may have the same histogram. To solve this issue, Kernel (epanechnikov) is used to consider the spatial arrangement of pixels according to the distance from the center [27].

### *B. Feature tracking algorithms*

Feature tracking offers a wide range of application possibilities in computer vision and control theories, such as medical robotics, surveillance, etc. A robust extraction and real-time tracking of features is a big step to success of other tasks (e.g. visual servoing). In this paper, we describe deterministic tracking algorithms combined with different kinds of feature extraction methods. Three of the most robust, efficient and simple methods are particle filter, mean-shift and Kanade-Lucas-Tomasi algorithms. Also in this section, we introduce Kalman filter which will be used as an option with the previous mentioned tracking algorithms. Kalman filter will be used as a prediction and estimation steps.

*1) Particle filter:* Particle filters, known as sequential Monte Carlo methods, is one of the most robust and effective target tracking algorithms, where it represents belief by sets of samples or particles. Particle filter is based on optimal Bayesian estimation and Monte Carlo model [11], and it describes the posterior probability distribution as a set weight of randomly sampled particles $\left\{\tilde{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}, i = 1, 2, ..., N\right\}$, where $\tilde{x}$ is the particle and $\tilde{w}$ is the weight of that sample.

The first step of particle filter is to initialize the particles randomly around the center of the desired object, $\left\{x_0^{(t)}, 1/N\right\}_{i=1}^{N} \; p(x_0)(t = 1, t = 1, 2, 3, ...)$. The initial particle weight is $1/N$ for each particle. Target and candidates models represent the feature histogram, Section III-A, of the selected object and each particle. Particle filter generates particles around the target model to obtain the candidate model of each particle. The normalized particles weight importance estimated by

$$\begin{aligned} w_t^{(i)} &\propto w_{t-1}^{(i)} p(y_t|x_t^{(i)}) \\ \tilde{w}_t^{(i)} &= w_t^{(i)} / \textstyle\sum_{i=1}^{N} w_t^{(i)} \end{aligned} \tag{1}$$

Where $p(y_t|x_t^{(i)})$ is the observation likelihood function. Then particle filter updates the particles by ignore the low weighted particles and reproduce the high weighted ones, and re-initialize the weights by $1/N$. Finally the estimated new center is

$$\hat{x}_t = \sum_{i=1}^{N} \tilde{w}_t^{(i)} x_t^{(i)} \tag{2}$$

*2) Mean Shift :* D Comaniciu et al. [12] applied mean shift algorithm for object tracking as an iterative Kernel based nonparametric low complex algorithm. The key of mean shift algorithm is to find the largest similarity between features in the current frame with that in the target frame, where these features can be represented as a histogram. The aim of mean shift algorithm is to maximize the correlation between histograms. In the mean shift tracking algorithm, a rectangular window is selected as the desired object, then a representative feature histogram is created for the desired object which will be called target model. For each frame, mean shift searches iteratively for the location of the closest region histogram to the target model and find the best candidate by maximizing the similarity function.

In this paper, We used the features from Section III-A to find the representative feature histogram. Among all the dissimilarity measures to compare between two histograms, Bhattacharyya distance is used.

$$\rho[\hat{p}(y), \hat{q}] = \sum_{i=1}^{n} \sqrt{\hat{p}(y)\hat{q}} \tag{3}$$

Where $p$ and $q$ are the target and the candidate models. To search for the new position at each iteration, the Bhattacharyya distance should be minimized. If representative feature histogram from Section III-A is used, the new position can be obtained by

$$x_i = \textstyle\sum_{i=1}^{n_h} x_i w_i / \sum_{i=1}^{n_h} w_i \tag{4}$$

*3) Kanade-Lucas-Tomasi:* Kanade-Lucas-Tomasi (KLT) feature tracker [13] Identifies and tracks good features from one frame to the next one. At any part of the tracking, if any feature is lost, a new feature will be detected. KLT starts with taking the same location of the desired object in the current and next frame. After getting the target and the candidate windows, KLT detect the feature points for these windows. Many types of detecting feature points methods can be used, such as Harris corners, FAST, SURF and minimum eigenvalue methods.

KLT uses each feature point as a center of a descriptor (e.g. 16x16), Section III-A. The next step to match between these features by determining which features comes from corresponding locations in both images. KLT matches between these descriptors by computing the pairwise distance, feature in the first image have the smallest distance with its corresponding feature in the second image.

After finding the matched features, there could be some incorrect matchings, called outliers, which will lead to incorrect results. RANSAC algorithm [28] is applied on the matched features to remove outlies and to provide the transition matrix by estimating the homography between the feature points in the target and candidate windows.

*4) Kalman Filter:* Kalman filter [29] has been embedded in the previous algorithms in order to increase the performance and to handle possible occlusions, by these mean the tracking procedure may be significantly accelerated. The

combined algorithm starts with Kalman prediction, after that the previous algorithms will be applied to get the new center which Kalman filter will use in the measurement step to estimate the new center location.

Kalman filer prediction step defines the motion as a noise covariance. The transition and observation models of kalman filter are

$$x_{k+1} = F_{k+1}x_k + m_k \tag{5}$$

$$z_{k+1} = G_{k+1}x_{k+1} + n_{k+1} \tag{6}$$

Where $x_{k+1}$ is the state vector at time t, and $z_{k+1}$ is the estimated position. $n_{k+1}$ (observation noise) and $m_{k1}$ (process noise) are Gaussian noises with zero mean and $R_k$ and $Q_k$ conveniences . $F_{k+1}$ represents the linear relationship between successive states, while $G_{k+1}$ represents the linear relationship between states and observations.

Using a recursive procedure, Kalman filter computes the minimum mean-square error of the estimated $x_k$ given the measurement $z_1, ..., z_k$.

Kalman filter can be added to mean shift and KLT as a prediction and estimation steps for the new location. On the other hand, for adding Kalman filter to particle filter, Kalman filter used to predict and update the particle.

## IV. Experimentation and results

The experiments were carried out with a core 5 Duo 3.70 GHz processor with 8GB RAM under Matlab. To estimate the accuracy of the compared algorithms we compute the bounding box overlap ratio between the estimated nerve position and the ground truth. In the experiments, we used for particle filter $N = 50$ as the number of particles. In mean shift algorithm, we set $threshold$ equal to $0.0001$. And in KLT algorithm, the initialization of the descriptors is $16 \times 16$ pixels around the feature points.

In this section, we evaluate the performance of the proposed methods. Firstly, we briefly describe the dataset. This followed by nerve tracking experiments which is divided into three categories, nerve tracking on the original images, nerve tracking after segmenting the foreground region (preprocessed images), and nerve tracking after applying despeckles filter (filtered images). finally, nerve tracking experiments after using Kalman filter with the proposed algorithms.

### A. Dataset

Experiments are made on Sonographic videos of the median nerve, which obtained from 10 patients using a Philips machine with a 5-12 MHZ transducer frequency. The dataset was acquired in real conditions at the Medipole Garonne hospital in Toulouse (France). The dataset of a total number of 6857 ultrasound images of the median nerve were used in these experiments. The ground truth was provided by two Regional anesthesia experts.

### B. Nerve tracking

As this paper focused on median nerves, which belong to the foreground (hyperechoic) region, the first step is to extract this kind of region by using morphological reconstruction [30]. Morphological reconstruction applies adaptive histogram equalization on the marker image to retrieve the reconstructed mask image.

It is well known that visual properties of US images are degraded by many affects such as artifacts, signal degradation and speckle noise. Speckle noise caused by the coherent source and noncoherent detector of echo ultrasound imaging systems, the speckle noise and the artifacts make automatic processing more complicated [31]. In this paper, we used various types of despecking filters such as homogeneous mask Area (HMA) [32], bilateral filter [33], Gabor filter [34], speckle reducing anisotropic diffusion (SRAD) [35] and different wavelet filters [36].

*1) Nerve tracking on the original images:* in the experiments, we applied the three algorithms on the original images, without extracting the foreground or applying despeckle filters. Tab. I shows the comparisons between the three algorithms with the proposed feature extraction methods. As shown in Tab. I, particle filter with JAMBP or AMBP achieved the best results, while mean shift and KLT algorithms had less stability and less performance accuracy. Also, it can be seen MBP, LBP and CLBP gave good results with particle filter but with less stability. Fig. 1 shows the results of particle filter with JAMBP and the ground truth of the nerve location.

*2) Nerve tracking on the preprocessed images:* the preprocessing procedure allows to extract the foreground (hyperechoic) region by using morphological reconstruction. In the experiments, we applied the proposed algorithms on the preprocessed images. Tab. I shows the comparisons between the three algorithms with the proposed feature extraction methods. As shown in Tab. I, particle filter with JAMBP or AMBP gave the best results. Mean shift and KLT algorithms had better performance than tracking on the original images, but still have less stability.

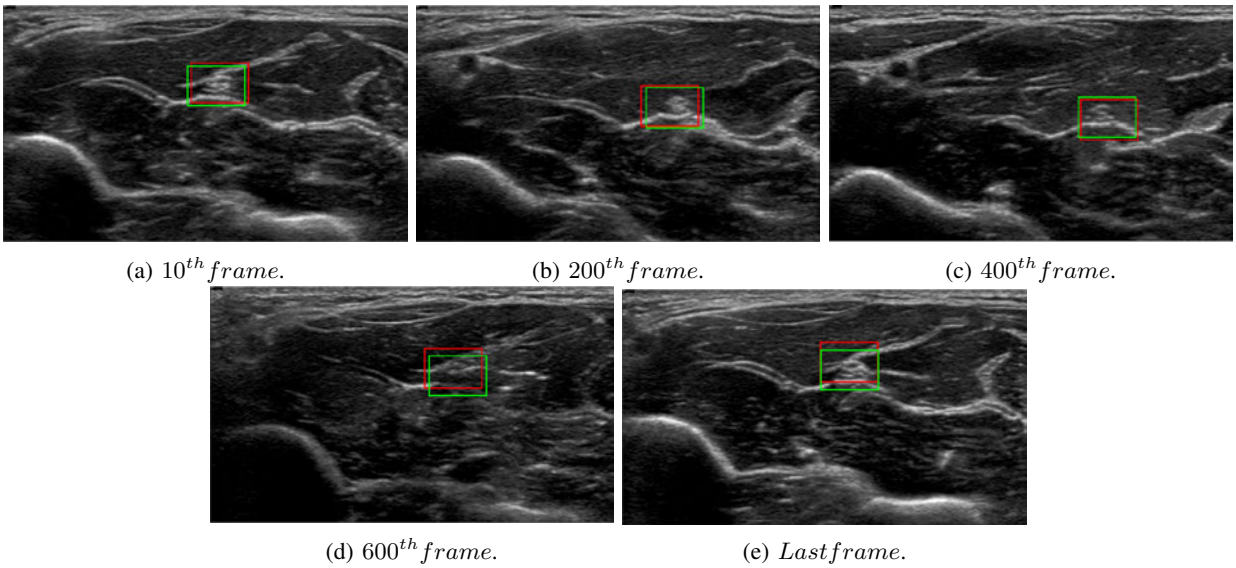
*3) Nerve tracking on the filtered images.:* here, we applied different kinds of despeckle filters on the preprocessed images. Filters were applied after performing foreground extraction process. We compared the performance of the proposed tracking algorithms on the preprocessed and filtered images. For filtered images, wavelet-daubechies filter had been used. As shown in Tab. I, particle filter with JAMBP or AMBP achieved the best result.

### C. Nerve tracking with Kalman filter

In order to achieve more accurate results and less computational cost, the proposed algorithms have been modified by including Kalman filter. Particle filter still have the best results, but adding Kalman filter improved the performance of mean shift and KLT algorithms.

TABLE I: Comparison of the feature extraction methods with the proposed algorithms.

| | Original Images | | | Preprocessed Images | | | Filtered Images | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | PF | MS | KLT | PF | MS | KLT | PF | MS | KLT |
| JAMBP | **0.85** | **0.68** | 0.54 | **0.91** | **0.75** | 0.73 | **0.91** | **0.76** | 0.73 |
| AMBP | 0.82 | 0.60 | 0.49 | 0.89 | 0.70 | 0.70 | 0.88 | 0.73 | 0.70 |
| MBP | 0.76 | 0.46 | 0.47 | 0.87 | 0.65 | 0.70 | 0.88 | 0.67 | 0.68 |
| CLBP | 0.76 | 0.52 | 0.39 | 0.86 | 0.67 | 0.59 | 0.86 | 0.68 | 0.61 |
| LBP | 0.74 | 0.51 | 0.38 | 0.84 | 0.67 | 0.62 | 0.81 | 0.67 | 0.63 |
| Hist | 0.60 | 0.44 | 0.42 | 0.64 | 0.52 | 0.39 | 0.54 | 0.53 | 0.41 |
| HOG | 0.71 | 0.47 | **0.57** | 0.79 | 0.58 | **0.77** | 0.79 | 0.59 | **0.78** |



(a) $10^{th} frame.$ (b) $200^{th} frame.$ (c) $400^{th} frame.$

(d) $600^{th} frame.$ (e) $Last frame.$

Fig. 1: Nerve tracking on the original images using particle filter with JAMBP (red rectangle for particle tracking and green rectangle for the ground truth).

## V. Discussion

This paper set out with the aim of nerve tracking in ultrasound images for regional anesthesia applications. We have introduced three algorithms involving an extensive feature extraction comparison to handle different issue related to nerve tracking. In this section, the discussion will be presented as, the performance of the feature extraction methods with the proposed algorithms, the improvement on the performance after adding Kalman filter to these algorithms, finally discussing the performance of AMBP and JAMBP feature extraction methods.

Seven feature extraction methods were evaluated in Section III-A. Tab. I shows the performance of the feature extraction methods with the proposed algorithms. As it can be seen, using particle filter or mean shift algorithm, JAMBP and AMBP provide the best results. HoG, JAMBP and AMBP perform better while applying KLT. In general, particle filter is more robust to noise, while mean shift and KLT do not perform well under the presence of noise. As shown in Tab. I, applying tracking algorithms on the filtered images does not make a huge improvement than applying the tracking algorithms on the preprocessed images, and in some cases it decreases the accuracy. The computational cost of particle filter depends on the number of particles, while in mean shift on the threshold value, and in KLT on the number of features and the size of the descriptor. Therefore in our experiments, particle filter and mean shift algorithm require less computation time.

In order to increase the accuracy and the robustness of the proposed tracking algorithms and to reduce the computational cost, to deal with partial or total occlusion, Kalman filter was combined with the tracking algorithms. The idea is to predict the position of the tracked object in the new frame based on the object's previous motion. Tab. II provides a comparison of the particle filter, mean shift and KLT algorithms after including Kalman filter. As shown, adding Kalman filter makes a small improvement on particle filter and KLT results, but it makes a huge improvement on the mean shift results.

After evaluating the proposed algorithms, Tab. III shows the tracking performance comparisons between several AMBP joint histogram methods (i.e. $AMBP, l/M/C$ , $JAMBP$). We observe that incorporating the joint histogram scheme increases the tracking performance, and AMBP,2/M/C gives better performance than AMBP,1/M/C or AMBP,0/M/C. Overall the accuracy of AMBP texture operator is competitive and provides good results with the proposed tracking algorithms.

TABLE II: Comparison of the feature extraction methods with the proposed algorithms after adding Kalman filter.

| | Original Images | | | Preprocessed Images | | | Filtered Images | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | PF | MS | KLT | PF | MS | KLT | PF | MS | KLT |
| JAMBP | **0.89** | **0.75** | 0.59 | **0.94** | **0.91** | 0.73 | **0.95** | **0.90** | 0.74 |
| AMBP | 0.86 | 0.72 | 0.51 | 0.92 | 0.88 | 0.71 | 0.93 | 0.85 | 0.70 |
| MBP | 0.78 | 0.53 | 0.47 | 0.89 | 0.83 | 0.69 | 0.91 | 0.82 | 0.71 |
| CLBP | 0.78 | 0.56 | 0.42 | 0.88 | 0.83 | 0.60 | 0.90 | 0.79 | 0.62 |
| LBP | 0.76 | 0.56 | 0.41 | 0.86 | 0.86 | 0.63 | 0.88 | 0.81 | 0.63 |
| Hist | 0.62 | 0.47 | 0.43 | 0.72 | 0.67 | 0.40 | 0.72 | 0.56 | 0.41 |
| HOG | 0.74 | 0.49 | **0.60** | 0.75 | 0.63 | **0.78** | 0.82 | 0.68 | **0.80** |

TABLE III: Comparison of AMBP and JAMBP feature extraction methods with Kalman particle filter.

| | Original | Preprocessed | Filtered |
|---|---|---|---|
| JAMBP | 0.89 | 0.94 | 0.95 |
| AMBP,2/M/C | 0.86 | 0.92 | 0.93 |
| AMBP,1/M/C | 0.82 | 0.88 | 0.88 |
| AMBP,0/M/C | 0.76 | 0.84 | 0.85 |

## VI. Conclusions and Future Works

This dissertation research addresses robotic guidance UGRA, where performing detection, tracking and visual servoing tasks is very challenging. In this project, we detect the nerve region by using an existing approach that uses machine learning technique. Also in this project, we reviewed the basics and nowadays techniques of visual servoing.

For tracking task, we have introduced a new robust nerve tracking technique based on integrating Adaptive Median Binary Pattern(AMBP) with simple feature tracking algorithms. AMBP operator fuses information from LBP and MBP to improve the noise robustness and to have a better encode for the scene microstructure information. In this study, we used three famous tracking algorithms (particle filter, mean-shift and Kanade-Lucas-Tomasi(KLT)) where seven different texture feature extraction methods used to represent the target region. Experimental results indicate that AMBP with particle filter performs much better than the other feature tracking techniques, and provide handling noise suppression without pre-filtering the images. To improve the results, to reduce the computational cost and to handle occlusion possibilities, Kalman filter had been added to the tracking algorithms.

In future work, in order to improve the tracking performance and to handle more types of noise, the proposed tracking approaches need to be improved more by using additional information, also by assessing these approaches on other databases. Another future work is to use our nerve detection and tracking techniques in real time with new ideas on visual servoing to visualize automatically the target regions.

## References


[1] B. C. Tsui and S. Suresh, "Ultrasound imaging for regional anesthesia in infants, children, and adolescentsa review of current literature and its application in the practice of extremity and trunk blocks," *The Journal of the American Society of Anesthesiologists*, vol. 112, no. 2, pp. 473–492, 2010.

[2] R. Alterovitz, M. Branicky, and K. Goldberg, "Motion planning under uncertainty for image-guided medical needle steering," *The International journal of robotics research*, vol. 27, no. 11-12, pp. 1361–1374, 2008.

[3] P. Marhofer, H. Willschke, and S. Kettner, "Current concepts and future trends in ultrasound-guided regional anesthesia," *Current Opinion in Anesthesiology*, vol. 23, no. 5, pp. 632–636, 2010.

[4] G. E. Woodworth, E. M. Chen, J.-L. E. Horn, and M. F. Aziz, "Efficacy of computer-based video and simulation in ultrasound-guided regional anesthesia training," *Journal of clinical anesthesia*, vol. 26, no. 3, pp. 212–221, 2014.

[5] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: a survey," *IEEE Transactions on medical imaging*, vol. 25, no. 8, pp. 987–1010, 2006.

[6] J. Zhang, C. Wang, and Y. Cheng, "Comparison of despeckle filters for breast ultrasound images," *Circuits, Systems, and Signal Processing*, vol. 34, no. 1, pp. 185–208, 2015.

[7] R. Vanithamani and G. Umamaheswari, "Performance analysis of filters for speckle reduction in medical ultrasound images," *International Journal of Computer Applications*, vol. 12, no. 6, pp. 23–27, 2010.

[8] E. Nadernejad, M. R. Karami, S. Sharifzadeh, and M. Heidari, "Despeckle filtering in medical ultrasound imaging," *Contemporary Engineering Sciences*, vol. 2, no. 1, pp. 17–36, 2009.

[9] S. Ambardar and M. Singhal, "A review and comparative study of de-noising filters in ultrasound imaging," 2014.

[10] O. Hadjerci, A. Hafiane, D. Conte, P. Makris, P. Vieyres, and A. Delbos, "Computer-aided detection system for nerve identification using ultrasound images: A comparative study," *Informatics in Medicine Unlocked*, vol. 3, pp. 29–43, 2016.

[11] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2. IEEE, 2000, pp. 126–133.

[12] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[13] C. Tomasi and T. Kanade, "Detection and tracking of point features," 1991.

[14] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[15] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust object tracking using joint color-texture histogram," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 07, pp. 1245–1263, 2009.

[16] D. Ding, Z. Jiang, and C. Liu, "Object tracking algorithm based on particle filter with color and texture feature," in *Control Conference (CCC), 2016 35th Chinese*. IEEE, 2016, pp. 4031–4036.

[17] P. Bilinski, F. Bremond, and M. B. Kaaniche, "Multiple object tracking with occlusions using hog descriptors and multi resolution images," in *Crime Detection and Prevention (ICDP 2009), 3rd International Conference on*. IET, 2009, pp. 1–6.

[18] O. Hadjerci, A. Hafiane, P. Makris, D. Conte, P. Vieyres, and A. Delbos, "Nerve detection in ultrasound images using median gabor binary pattern," in *International Conference Image Analysis and Recognition*. Springer, 2014, pp. 132–140.

[19] L. Liu, Y. Long, P. W. Fieguth, S. Lao, and G. Zhao, "Brint: binary rotation invariant and noise tolerant texture classification," *IEEE*

*Transactions on Image Processing*, vol. 23, no. 7, pp. 3071–3084, 2014.

[20] A. Hafiane, K. Palaniappan, and G. Seetharaman, "Adaptive median binary patterns for texture classification," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 1138–1143.

[21] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[22] A. Hafiane, G. Seetharaman, K. Palaniappan, and B. Zavidovique, "Rotationally invariant hashing of median binary patterns for texture classification," *Image Analysis and Recognition*, pp. 619–629, 2008.

[23] A. Hafiane, G. Seetharaman, and B. Zavidovique, "Median binary pattern for textures classification," in *International Conference Image Analysis and Recognition*. Springer, 2007, pp. 387–398.

[24] A. Hafiane, K. Palaniappan, and G. Seetharaman, "Joint adaptive median binary patterns for texture classification," *Pattern Recognition*, vol. 48, no. 8, pp. 2609–2620, 2015.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[26] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[27] M. P. Wand and M. C. Jones, *Kernel smoothing*. Crc Press, 1994.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[29] V. Karavasilis, C. Nikou, and A. Likas, "Visual tracking by adaptive kalman filtering and mean shift," in *Hellenic Conference on Artificial Intelligence*. Springer, 2010, pp. 153–162.

[30] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms," *IEEE transactions on image processing*, vol. 2, no. 2, pp. 176–201, 1993.

[31] K. Z. Abd-Elmoniem, A.-B. Youssef, and Y. M. Kadah, "Real-time speckle reduction and coherence enhancement in ultrasound imaging via nonlinear anisotropic diffusion," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 9, pp. 997–1014, 2002.

[32] C. P. Loizou, C. S. Pattichis, C. I. Christodoulou, R. S. Istepanian, M. Pantziaris, and A. Nicolaides, "Comparative evaluation of despeckle filtering in ultrasound imaging of the carotid artery," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 52, no. 10, pp. 1653–1669, 2005.

[33] S. Balocco, C. Gatta, O. Pujol, J. Mauri, and P. Radeva, "Srbf: Speckle reducing bilateral filtering," *Ultrasound in medicine & biology*, vol. 36, no. 8, pp. 1353–1363, 2010.

[34] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *JOSA A*, vol. 2, no. 7, pp. 1160–1169, 1985.

[35] Y. Yu and S. T. Acton, "Speckle reducing anisotropic diffusion," *IEEE Transactions on image processing*, vol. 11, no. 11, pp. 1260–1270, 2002.

[36] L. Dong, "Adaptive image denoising using wavelet thresholding," in *Information Science and Technology (ICIST), 2013 International Conference on*. IEEE, 2013, pp. 854–857.

# 3D Reconstruction of Urban Environments using Sensor Fusion

Arjun Balakrishnan

*Abstract*— **It is highly certain that the near-future will see intelligent autonomous vehicles serving the transportation needs of humanity with remarkable reliability and efficiency. The pLaTINUM project envisions a cloud based geographical information system containing three-dimensional map and semantic details, that can serve remote agents (vehicles, gadgets, robots etc.) in localization and navigation and in return, use the observations made by remote agents to update and enrich the existing map. The first contribution of this paper is a brief classification and analysis of the existing odometry, mapping and surface reconstruction techniques using visual and inertial sensors, followed by the design and development of the system based on this analysis. It is observed that vision based surface reconstruction of large scale outdoor maps is relatively less studied area due to the poor performance of stereo vision systems compared to other 3D scanning techniques. The second contribution of this paper is to propose a post-filtering routine for disparity images obtained from conventional stereo vision system, to improve the quality of 3D reconstruction. The improved disparity image is used in an existing surface reconstruction framework, which is originally proposed for RGB-D sensors and small scale indoor sequences, and obtained high quality results.**

## I. INTRODUCTION

There are several existing geographical information systems (GIS) that serve as the references for localization and navigation of mobile platforms. The pioneers among them are undoubtedly Google Maps and its sibling softwares like Google Earth and Google Streetview, which are server based web applications that use a variety of modalities to acquire data such as satellite imagery, aerial imagery, conventional imaging using vehicles and human volunteers, and contributions from its users. Huge operational cost and complexity of data acquisition, limit their information updating frequency to two or three times in an year. Other comparable services like Apple Maps, Bing Maps and Here Maps hardly possess any superiority in these aspects. OpenStreetMap is an open source solution and depends heavily on user feedbacks such as GPS traces, map edits etc., which results in greater update frequency and easy map management. But on the down side, it is server based and not available in 3D. TerraWeb3D is an open source platform provides 3D and 2D maps and particularly suited for mobile platforms because of its lightweight execution, but lacks the user feedback system. By comparing and contrasting these factors, it is evident that the scope is high for a cloud based, open source, multi-modal, dense 3D and 2D geographical information system, which builds and updates maps collaboratively by using its users as mapping agents.

The pLaTINUM (Long Term MappINg for Urban Mobility) project, is proposed as a cloud based geographical information system (GIS) which can provide navigation services to the remote agents and use automatic feedbacks from them to update the data stored in the system. On the cloud, geometric, photometric and semantic information of the environment will be stored as 3D representations on a Geo-localized global frame. This information will be used for the path planning and autonomous navigation of remote agents (vehicles, personal gadgets etc.) and the differences in the environment observed by the agents are processed later to update the current reference map. Due to the advancements in automobile electronics and consumer electronics, vehicles and smart gadgets are embedding multiple sophisticated sensors and cameras to track motion and provide navigation assistance. By combining available (or added) tracking sensors and cameras nearly all personal gadgets and vehicles can be used as efficient sensing unit for map building. Even though on-board processing power on mobile platforms has increased drastically, a cloud based implementation of map building will not pose any extra burden to user end.

In this paper, the primary works done as part of the mapping tasks associated with the pLaTINUM project are presented. Section II presents the review of related works and the development of the total system is described in Section III. Section IV is dedicated to formulation and implementation of the system. Finally, Section V shows the results and its analysis.

## II. REVIEW OF RELATED WORKS

Since the scope of this work spans over several techniques in odometry and mapping, the related works are classified and studied.

### A. Monocular Visual Odometry (Mono VO)

In Monocular VO, 3D position estimation of the points as well as the motion estimation is done using consecutive images from a single camera. Among sparse feature based methods, which uses distinctive and repeatable features, work by Nister et al [26] proposes the use of RANSAC integrated 5-Point minimal solver [24] to calculate the motion from 3D to 2D matching. Here 3D points are computed by triangulating from two consecutive images. Several works like, [6], [14], uses the 5-Point solver inspired by Nister et. al [26]. Strasdat et. al [31] made improvements in regular monocular

This work was conducted at LITIS laboratory, University of Rouen Normandy

A. Balakrishnan is a student from University of Burgundy studying in Masters of Computer Vision arjun.balakrishnan@insa-rouen.fr

This work is supervised by Dr. Pascal Vasseur, University of Rouen Normandy pascal.vasseur@univ-rouen.fr

VO by adopting the key-frame and Bundle Adjustment (BA) [33] optimization. Konolige et al. [12] showcase a 10km VO test on a ground vehicle with windowed-bundle adjustment which showed up to 5 times more accuracy than without it. They also have used the 5-Point random sample consensus (RANSAC) solver but proved to recover structure of the environment and relative pose more accurate than its predecessors thanks to their BA scheme. However in an interesting work, Tardif et. al [32] decoupled the translation and rotation estimation by using different set of points for each, and proved that robust long term monocular VO odometry is possible without BA. They estimated rotation using the points at infinity and translation from recovered 3D map, while outliers were still removed by 5-Point algorithm.

### *B. Stereo Visual Odometry (Stereo VO)*

The technique for estimating relative camera poses and recovering 3D model of an object from set of consecutive images was known as SFM (Structure From Motion) in the computer vision community for a long time. Adapting SFM for odometry is first clearly stated and analyzed by Moravec [21]. In this work, matching features in stereo frames were triangulated and camera pose was estimated as rigid body transformation between two sets of 3D points. By adapting Moravec's method, [18] used error covariance matrix for the triangulated 3D points from binocular cameras to model the uncertainty in motion estimation. Results of this procedure showed more accurate than Moravec's method by reducing the error to 2% of the distance traveled. However, several versions of VO has been proposed by replacing and modifying the components used in above mentioned works. For instance, Lacroix et al. [13] uses dense stereo and identify the features using correlation function around its peaks. But Cheng et al. [3], [16], which demonstrate the actual VO method finally used in real Mars rovers, improved this idea by using curvature of correlation peak to define error covariance matrix for points in image domain and by using RANSAC for outliers removal. In [19], the famous 'good features to track' by Shi and Thomasi [28] was used to detect the keypoints, which followed by filtering of those points by the confidence in depth estimation. Unlike these 3D to 3D matching methods for motion estimation, Nister et al. [25] developed a break through algorithm which used structure to image correspondence (3D to 2D matching) to estimate the motion. Here triangulated points from a stereo pair is matched against their projections in the consecutive frame. Another different motion estimation technique which uses image points correspondence (2D to 2D matching) was introduced by Comport et al. [4]. The idea is to use two stereo pairs to find the matching points in all four images and estimate the motion directly without the need of any triangulation.

### *C. Direct Visual Odometry (Direct VO)*

Direct Visual methods are defined as methods for shape or pose estimation using every pixel intensity information available in images. Contrast to feature based (indirect) methods, direct methods minimizes an alignment error measure defined on parameters like image brightness, brightness-based cross-correlation etc., to estimate unknown parameters of camera motion. Comport et. al [5] uses stereo vision system for direct image alignment on depth images. Newcombe et. al [23] developed a dense tracking and mapping (DTAM) framework, which exhibits high quality pose estimation in real time for moderate movements. In [27] dense depth maps are accurately computed using computationally costly variational formulation, hence not suitable for real time. A semi-dense depth filtering formulation is proposed in [8] which reduced the computational demand and enabled the authors to implement this process on a smartphone for AR applications. Engel et al. [7] adds SLAM capabilities to direct visual odometry by extracting keyframes and performing loop closures using them. Globally consistent semi dense maps can be obtained using their LSD-SLAM method proposed.

### *D. Visual Inertial Odometry (VIO)*

Based on the integration of inertial sensors into visual odometry, VIO systems can be classified into two, tightly-coupled and loosely-coupled systems. In general, IMU measurements and visual pose estimations are treated as independently in loosely-coupled systems, which is the mode used in this work. Konolige et. al [12] incorporates pre-integrated IMU data to optimize the visual pose estimation. Forster et. al [9] focuses on the pre-integration of IMU data between two visual acquisitions, using the manifold structure of rotation groups. Weiss. et. al [35] uses vision only pose estimation to update the pose predictions by IMU using an Extended Kalman Filter (EKF). Loosely-coupled methods allows to integrate IMU with existing VO or visual SLAM mehtods easily. Armesto et. al [1], proposes a generalized sensor fusion algorithm using EKF and Unscented Kalman filter (UKF) specifically for multi-rate sensors. At high rates, IMU can sense the dynamics of the system better than assumed models, which inspired Sirthaya et al. [29] to use an error propagation model for fusion, which allows to model different characteristics of IMU. There are several other methods which use IMU data for bundle adjustment of poses generated by VO, instead of fusing using filters.

### *E. Surface Reconstruction Techniques*

The amount of surface reconstruction (or meshing) techniques, developed for stereo vision based 3D systems are very less. Even though several methods addresses the issue of noisy point clouds, they expect poor outputs from LIDARs or RGB-D cameras, which is dense and accurate enough to identify and remove noise from them. [34] proposes 3D modeling of objects or static environments from multiple views of range images. Frueh et al. [10] generated meshed reconstruction of 8 million points obtained though a 3km drive in urban environments. Zoltan et. al [17] presents a fast reconstruction using incremental surface growing principle, where points are connected using triangles until that surface has no more connections. The idea of superpixels

- homogeneous image patches - is used in [2], but they require RGB-D or LIDAR data to generate reference point clouds. Nawaf et al [22] proposed color and optical flow based superpixel estimation and meshing directly from stereo vision system. They produced smooth and dense mesh of several challenging outdoor scenes, but the problem of long term mapping was not addressed. Volumetric meshing techniques are also popular in the research community due to their online performance and ease of integration with mapping or SLAM techniques. Lu Ma et al. [15] combined stereo vision system (for reliable outdoor operation) with inertial tracking (robust pose estimation) to create online implementation of volumetric fusion on a GPU. Steinbrucker et al. [30] proposes another volumetric fusion method using supervoxels on RGB-D data of indoor sequences.

## III. Design of the System

Based on the analysis presented in Section II, the total system is developed by dividing all the required functionalities into two different blocks as in 1. Odometry block estimates the relative poses of consecutive frames using mono VO, stereo VO and direct VO independently. 5-Point algorithm by Nister [24] was chosen for mono VO as it is the most general method which is scalable to large environments and works effectively in both indoor and outdoor environment. Feature tracking is done by Kanade-LucasTomasi (KLT) method instead of detection and matching proposed in the original work. For stereo VO, 3D to 2D matching method has superiority over others as shown in [26], due to the fact that minimizing 3D positional error is less efficient that minimizing image reprojection error. The odometry block from LSD-SLAM with monocular vision [7] is selected for direct VO, primarily because it is claimed to work both indoors and outdoors. In order to incorporate IMU data to develop VIO system, VO pose estimation from monocular, stereo and direct methods are fused with IMU measurements using a multi-rate EKF filter [1]. After obtaining the final pose estimation by sensor fusion, stereo vision is used to create depth images. 'Fastfusion' proposed in [30] is used because of the efficiency in handling holes, and its ability of performing global optimization while map creation. However, this is designed for RGB-D sensors and indoor environments, but generating a high quality depth image and removing uncertain information using post processing on disparity images is proposed, to adapt this method for stereo outdoor sequences.

## IV. Implementation

### A. Monocular Vision Scheme

*1) Formulation:* Consider two images $I_k$ and $I_{k-1}$ at times consecutive time instants $k$ and$k-1$, with projection matrices $P_{k-1} = K\,[I|0]$ and $P_k = K\,[R|t]$. Since the K is known from calibration, essential matrix E which relates the homogeneous points in these two images $p$ and $p'$ by the epipolar constraint $p'Ep = 0$. These points are chosen from matched or tracked feature points in consecutive frames. 5-Point algorithm proposed by Nister [24] formulates the estimation of E as a linear system and solve with singular value decomposition (SVD). Since essential matrix comprises of rotation and translation up to a scale factor, recovering R and t from E is easily possible. However four potential solutions using different combinations of R and t- $P_A \equiv [R_a|t_u]\,, P_B \equiv [R_a|-t_u]\,, P_C \equiv [R_b|t_u]\,, P_D \equiv [R_b|-t_u]$ - can be observed as valid estimation. This ambiguity is solved *cheirality check* - triangulation of a point and choosing the solution that results this point in front of both camera views.

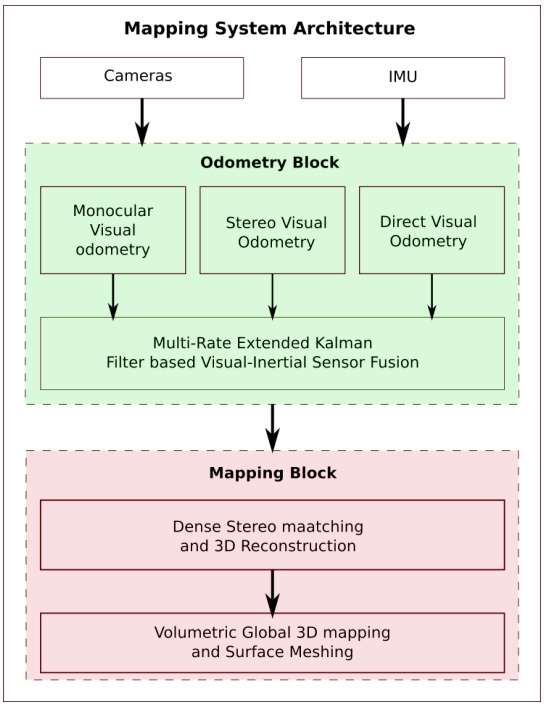



Fig. 1. Block diagram of envisioned mapping system

*2) Outliers rejection:* While tracking features, lost points are identified and the tracked features set is updated accordingly. Since the number of features used for essential matrix estimation is sufficiently higher than 5, RANSAC based outlier removal is incorporated in the essential matrix estimation step. Five points are chosen randomly and an essential matrix is computed, which is verified using the maximum distance from points to corresponding epipolar lines. The process is repeated until maximum (or above a threshold) points satisfies the estimated essential matrix.

### B. Stereo Vision Scheme

*1) Formulation:* Two cameras separated by a distance (baseline) are connected to a rigid body platform. Images from both cameras are acquired in synchronization. For ease of description, two terms are additionally defined: static stereo and temporal stereo. Static stereo considers simultaneous images from left and right cameras, whereas temporal stereo considers consecutive images from same camera (much like a monocular vision system). Using static stereo matching, set of matching features from the previous

image pair $p_{k-1}$ and $p'_{k-1}$ are triangulated to 3D points $X_{k-1}$. Temporal matching is used to find the matching features $p_k$ to $p_{k-1}$, and inter-frame transformation $T_k$ is estimated. The general formulation of this step as proposed in [24], involves minimization of reprojection error as follows:

$$\underset{T_k}{arg\,min} \sum_i \left\| p_k^i - \widehat{p}_k^i \right\|^2 \tag{1}$$

where $\widehat{p}_k^i$ is the projection of triangulated points $X_{k-1}$ on to $I_{lk}$ using projection matrix $P_k$. DLT method proposed in [11], provides efficient and straightforward solution to this Perspective-n-Point (PnP) problem by developing linear algebra system and solving using SVD method.

*2) Outlier rejection:* Stereo VO involves two feature tracking steps- in static stereo and temporal stereo - which increases the degradation effect due to outliers than monocular VO. RANSAC based outlier removal steps are incorporated to both matching processes. For static stereo matching, RANSAC uses fundamental matrix hypothesis whereas essential matrix hypothesis is used in temporal stereo matching. Inliers satisfy both of these constraints are chosen for triangulation. Due to the inherent behavior of stereo matching, triangulation error increases as distance to the 3D point increases. Hence only 3D points less than a threshold distance is considered for 3D to 2D pose estimation. PnP problem is also solved in RANSAC framework, where a subset of inlier matches are chosen for the computation and validated over the reprojection error.

### C. Direct Image Alignment Scheme

To perform direct VO, the works proposed in [7] are used. Even though it is a SLAM system, which includes mapping and loop closures, only the camera pose estimation part is used for our purpose. Instead of tracking the features, the intensity gradients of initial image are computed and 'pixels of interest' (pixels that have high intensity gradients, edges and corners) are initialized along with an arbitrary depth distribution of those points associated with a large variance. The first image is chosen as the first keyframe. Tracking is done by aligning preceding frames to the lastly initialized keyfame. The required pose is represented by an SE(3) transformation and is found by an iteratively reweighted Gauss-Newton optimization that minimizes the variance normalized photometric residual error. When the uncertainty of depth estimation grows beyond a limit new keyframes are selected and the process is repeated.

### D. Multi-Rate Visual-Inertial Fusion

Multi-Rate (MR) sensor fusion is achieved using two techniques: MR-Hold and MR-Sampler. The general implementation of MR-EKF is presented in Fig 2. MR-Hold receives the input and holds at prediction step, while MR-Sampler interfaces outputs at update step. Let us consider IMU data as input ($u_t$) and visual pose estimations as outputs ($Z_t$). An input history vector $U$ is considered to hold the inputs.

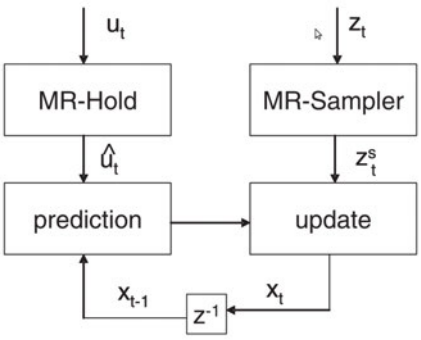



Fig. 2. General MR EKF scheme

A general discrete-time non-linear system can be modeled as:

$$x_k = f\left[x_{k-1}, u_{k-1}, w_{k-1}\right] \tag{2}$$

$$y_k = h\left[x_k\right] + \nu_k \tag{3}$$

where $x_k$ is state of the system, $y_k$ is the output vector, $w_k$ is the system noise and $\nu_k$ is the measurement noise. $f[.]$ and $h[.]$ are non-linear functions governing system dynamics and measurements. It is also important to define noise covariance matrices $P_k$ for system vector, $Q_k$ for system noise and $R_k$ for measurement noise. Let $j$ denote the time instants at interval $T$. Now the steps involved in MR-EKF can be formulated as following:

1) **MR-Hold** block receives the IMU input ($u_k$) for each $T_I$ intervals. But for all $j$ MR-Hold checks whether IMU signal is present or not. If it is present, the input is added to input history vector $U$ and chooses that current input as the input for prediction ($\widehat{u}_k$). If IMU data is not available at that $j$, last $n$ inputs are retrieved from $U$ and extrapolated to generate input at $j$ which is used as $\widehat{u}_k$ for prediction.
2) **MR-Sampler** block is used to generate size varying output vector $y_k$ by sampling visual and inertial inputs. It also chooses measurements function $h[.]$ and measurements noise $R_k$ according to the input signals present at that $j$.
3) Conventional EKF state prediction is done using system dynamics function $f[.]$.
4) Size of the output vector produced by MR-Sampler is checked. If it corresponds to the sampling of both sensors, EKF update is performed. Else, the predicted state and state covariance matrix is used for next iteration.

### E. Disparity Computation and Mapping

To overcome the issues of conventional stereo matching methods, post filtering of disparity images are proposed. Edge-preserved smoothing filters are a class of non-linear filters that can smooth the textures regions while keeping the edges undisturbed. Min et al. [20] proposed fast global smoothing (FGS) filter that achieves such smoothing by

minimizing the combined measure of differences between the original and filtered image, and the gradients in the original image. This method takes an arbitrary image as initialization for the desired output. This idea is exploited in disparity filtering as follows:

1) Initialize $l\_src$ - left image and $r\_src$ - right image.
2) Compute two disparity maps: $lr\_disp$ - left to right matching and $rl\_disp$ - right to left matching.
3) Create a confidence matrix $conf\_mat$: The similarity of disparities in $lr\_disp$ and $rl\_disp$ are used. The closer the disparities of a pixel is, higher the confidence associated with that pixel (in the range [0,1]).
4) Perform element-wise multiplication of $lr\_disp$ by $conf\_mat$: This will give a weighted disparity map $disp\_w$, where uncertain pixels are negligible.
5) Apply FGS filter on $disp\_w$ using $l_src$ as initialization image.
6) Apply FGS filter on $conf\_mat$ using $l_src$ as guidance image.
7) Perform element-wise division of smoothed $disp\_w$ by smoothed $conf\_mat$. The output will be filtered disparity image to be used for 3D point cloud computation.

After applying FGS filtering routine, the irregularities and discontinuities in the disparity images are removed as shown in Fig. 3. A segmentation procedure is adopted to identify the sky from the RGB image using color based thresholding and its connectivity to the top boundary of image. Irrespective of the disparity present in the detected sky regions in the disparity image, all values are discarded to ensure absence of sky in the map. The pose estimations and depth images are used as inputs for 'FastFusion' and performs global optimization by merging corresponding voxels in each frame. An growing octree is used as the data structure to store map data at different scales and the total algorithm is heavily optimized to run real-time on normal CPU with multi-threading. For outdoor scenes observed by the moving vehicles, the overlap between scenes can be less in close areas of the scene, hence a multi-frame consistency check is also introduced. This ensures that a part of the scene is observed in more than a pre-defined number of views, in order it to be considered for meshing. This reduces that chances of holes in the meshed surfaces to a large extent.

## V. Results and Discussions

The results obtained for a standard KITTI dataset using different odometry schemes are presented in Fig. 4. This 360m dataset contains two long mainly-translational motions, a small-radius turn and a halt at the end for more than 40 frames. Each scheme is compared with available ground truth, and clearly, stereo VO shows most accurate tracking. The final positional error for stereo VO is approximately 5% and orientation error is approximately 0.1%. Monocular VO offers very poor performance when large rotation and small translation scenarios, such as the curve in the track in Fig. 4. LSD-SLAM also failed in exactly tracking turn, and it also estimated some motion when the camera was actually stationary. However, considering the translational motion after the turn, monocular VO produced consistent poses and it detected the halt clearly, unlike LSD-SLAM. The errors in monocular VO was mainly due to the wrong translation estimation during the turn, while rotations were sufficiently accurate. All VO results have been improved after fusing with inertial measurements, mainly the sharp turn is tracked better in monocular VO and LSD-SLAM.

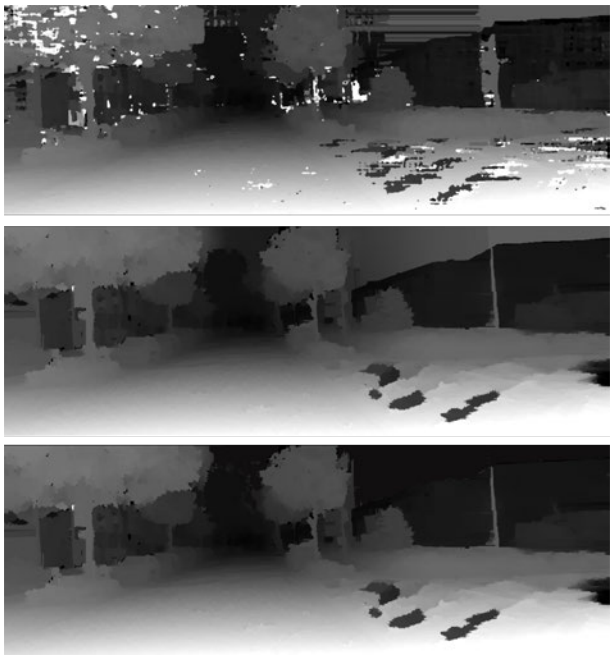


Fig. 3. Results of post-filtering of disparity images. Top: original disparity image, Middle: output disparity image after FGS filtering, Down: disparity image after the removal of sky regions

After the sensor fusion, positional error of stereo VO reduces to below 1%. It is interesting to note that the error due to the estimation of translational motion in LSD-SLAM when the camera was actually stationary, is also reduced using the inertial information that exactly depicts the halt. Fig. 5 shows the result of fusion and mesh reconstruction result for the same dataset. It is evident that the quality of the map is improved greatly after the post-filtering of disparity images.

## VI. Conclusions and Future Works

In this work, development of a routine used for large-scale, dense, surface-reconstructed 3D map is studied and presented. It includes the estimation of poses of a sensor platform containing RGB cameras and IMU sensor. Then using the extracted poses, the dense 3D reconstruction and mapping of the environment is done using stereo matching techniques. Several existing odometry techniques has been extensively tested with several possible adaptations to tackle the issue of long-term robust outdoor motion estimation. The output of the stereo match is often observed as erroneous and non consistent in large scale urban scenes. To address this issue, a post-processing method for disparity images is proposed which uses an edge-preserving smoothing filter

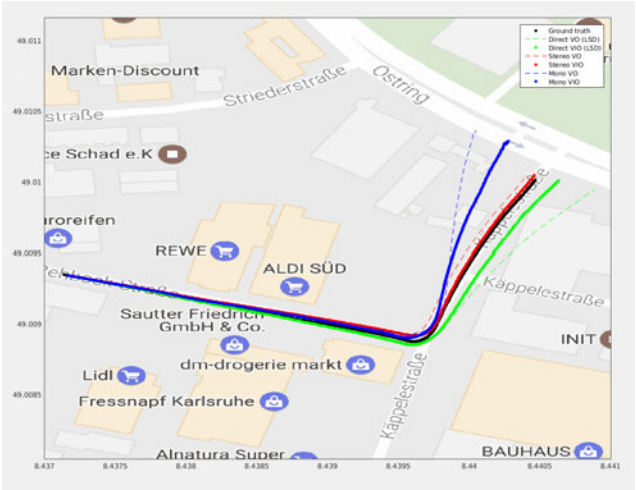



Fig. 4. Results of different visual-inertial odometry methods for a standard KITTI dataset

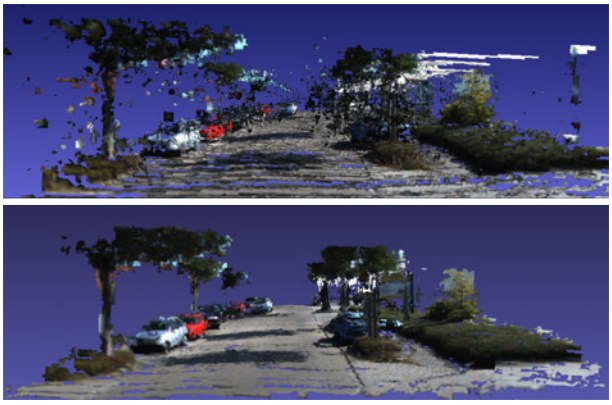


Fig. 5. Results of mapping block: meshed reconstruction of a scene. Top: meshed reconstruction using original disparity images. Bottom: meshed reconstruction using filtered disparity images as proposed in Section

based on the confidence of disparity estimation and the structural information available in the original images. This improved the disparity results considerably without much degradation in actual depth estimation. Apart from this, sky regions are segmented out from the scenes to facilitate relevant map creation. A volumetric fusion method is used to merge the scenes from different views and creating the dense map by reconstructing the surfaces. The generated maps are having enough quality to detect objects, understanding the scene etc.

The future work for this research are identified based on the results obtained. Since the visual odometry methods used are computationally efficient, it can be extended directly to real time with slight optimizations in feature tracking. A tightly-coupled indirect visual-Inertial odometry method is planned for the next stage to overcome the challenges present in estimating the pose while camera is not moving or while having the presence of moving objects in the scene. The implementation of the fusion method on a GPU is also proposed as future work. High quality GPS system will be incorporated with the existing sensor unit which will serve as a global optimization measurement for map creation.

## VII. ACKNOWLEDGMENTS

This project is developed under the collaboration of LITIS, Le2I, LAGADIC (INRIA) and MATIS (IGN) and funded by the French National Research Agency (ANR).

## REFERENCES

[1] Leopoldo Armesto, Josep Tornero, and Markus Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *The International Journal of Robotics Research*, 26(6):577–589, 2007.

[2] András Bódis-Szomorú, Hayko Riemenschneider, and Luc Van Gool. Efficient edge-aware surface mesh reconstruction for urban scenes. *Computer Vision and Image Understanding*, 157:3–24, 2017.

[3] Yang Cheng, Mark Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 903–910. IEEE, 2005.

[4] Andrew I Comport, Ezio Malis, and Patrick Rives. Accurate quadri-focal tracking for robust 3d visual odometry. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 40–45. IEEE, 2007.

[5] Andrew I Comport, Ezio Malis, and Patrick Rives. Accurate quadri-focal tracking for robust 3d visual odometry. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 40–45. IEEE, 2007.

[6] Peter Corke, Dennis Strelow, and Sanjiv Singh. Omnidirectional visual odometry for a planetary rover. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 4007–4012. IEEE, 2004.

[7] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[8] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1456, 2013.

[9] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

[10] Christian Frueh and Avideh Zakhor. Constructing 3d city models by merging ground-based and airborne views. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–562. IEEE, 2003.

[11] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] Kurt Konolige, Motilal Agrawal, and Joan Sola. Large-scale visual odometry for rough terrain. In *Robotics research*, pages 201–212. Springer, 2010.

[13] Simon Lacroix, Anthony Mallet, Raja Chatila, and Laurent Gallo. Rover self localization in planetary-like environments. In *Artificial Intelligence, Robotics and Automation in Space*, volume 440, page 433, 1999.

[14] Maxime Lhuillier. Automatic structure and motion using a catadioptric camera. In *Proceedings of the 6th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, 2005.

[15] Lu Ma, Juan M Falquez, Steve McGuire, and Gabe Sibley. Large scale dense visual inertial slam. In *Field and Service Robotics*, pages 141–155. Springer, 2016.

[16] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.

[17] Zoltan Csaba Marton, Radu Bogdan Rusu, and Michael Beetz. On fast surface reconstruction methods for large and noisy point clouds. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3218–3223. IEEE, 2009.

[18] Larry Matthies and STEVENA Shafer. Error modeling in stereo navigation. *IEEE Journal on Robotics and Automation*, 3(3):239–248, 1987.

[19] Annalisa Milella and Roland Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on*, pages 21–21. IEEE, 2006.

[20] Dongbo Min, Sunghwan Choi, Jiangbo Lu, Bumsub Ham, Kwanghoon Sohn, and Minh N Do. Fast global image smoothing based on weighted least squares. *IEEE Transactions on Image Processing*, 23(12):5638–5653, 2014.

[21] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.

[22] Mohamad Motasem Nawaf, Md Abul Hasnat, Desiré Sidibé, and Alain Trémeau. Color and flow based superpixels for 3d geometry respecting meshing. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 153–158. IEEE, 2014.

[23] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[24] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[25] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. Ieee, 2004.

[26] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.

[27] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2609–2616. IEEE, 2014.

[28] Jianbo Shi et al. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[29] S. Sirtkaya, B. Seymen, and A. A. Alatan. Loosely coupled kalman filtering for fusion of visual odometry and inertial navigation. In *Proceedings of the 16th International Conference on Information Fusion*, pages 219–226, July 2013.

[30] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Volumetric 3d mapping in real-time on a cpu. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2021–2028. IEEE, 2014.

[31] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2010.

[32] Jean-Philippe Tardif, Yanis Pavlidis, and Kostas Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531–2538. IEEE, 2008.

[33] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustmenta modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[34] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318. ACM, 1994.

[35] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 957–964, May 2012.

# Emotion Assessment Based On Remote Photoplethysmography

LI Peixi, Yannick BENEZETH and Richard MACWAN

Le2i FRE2005, CNRS, Arts et Métiers, Univ. Bourgogne Franche-Comté, Dijon, France

*Abstract*— **Heart Rate Variability (HRV) is a well-known measure of the human emotion, because the HRV value is related to the human Autonomic Nervous System (ANS) activity. Photoplethysmography (PPG) is a technology which can be used for the contact Heart Rate (HR) measurement. This technology is much cheaper than the conventional technologies such as Electrocardiography (ECG). Inspired by the principles of the contact PPG technology, the remote HR measurement using a camera has been proposed. However, the non-contact method has much more noise than the contact measurement. So a good Region of Interest (ROI) is very necessary for the remote measurement. In this paper, we utilize the facial landmark detection to improve the ROI detection and compare it with the state of the art methods. This paper also shows that the remote monitoring of HRV can be implemented by simply using a camera. The HRV values obtained by the remote method are almost the same with the ground truth in frequency domain. An experiment was done for the remote assessment of the negative emotion based on HRV. The Very Low Frequency (VLF), Low Frequency (LF) and High Frequency (HF) components of HRV were extracted by applying the Fast Fourier Transform (FFT). Besides the conventional power ratio LF/HF, a new feature (VLF + LF)/HF is proposed in this paper to assess the emotion. The result showed that both LF/HF and (VLF+LF)/HF are good features to assess the emotion remotely. The emotion states of 13 of a total 16 subjects were correctly reflected by these features.**

## I. INTRODUCTION

The study of HRV is a practical and convenient method to assess the psychological stress and working cognitive load of a human. The HR monitoring can be implemented by using photoplethysmography (PPG). A PPG equipment is used for optically getting the plethysmogram. In the device, the pulse oximeter is utilized for illuminating the skin and measuring changes of the light absorption. The amount of light absorption is a function of the blood volume [1]. With the change of the cardiac pulse, the blood volume of micro vascular all over the human body also get changed, so the Blood Volume Pulse (BVP) measured at the peripheral body tissues (such as the fingertip) can be utilized as a parameter of the cardiac cycle measurement. With such principle, the PPG can measure the local BVP, and the HR can be obtained based on this BVP

As a more convenient way of HR measurement, the remote PPG method using the camera was proposed by Jones et al. [2]. The scientific principle of remote PPG for HR measurement is almost the same with the contact PPG method. Instead of using the LED and professional photo-detector, the remote PPG method can simply make use of ambient light or normal light sources and a camera, which is much more convenient than the contact method. However, there is much more noise of remote PPG measurement. Some special processing algorithms should be implemented before calculating the HR from the BVP measured by the remote method.

The HRV can be computed from BVP or HR. Many research studies have indicated that the HRV is a good parameter for human emotion assessment and animal emotion interpretation [3] [4].

The main tasks addressed in this paper are:

1) Improve the precision of ROI detection with facial landmark detection and compare different ROI detection methods to get the best approach to satisfy the remote HR measurement.

2) Analyze the HRV calculated From BVP. The HRV from remote PPG should be almost the same with HRV from the ground truth (contact PPG). The time domain and frequency domain of the HRV are analyzed to see if the HRV from remote sensor and contact sensor match very well with each other. This is the foundation of the emotion detection.

3) Detect the negative emotion with non-contact HRV method.

The first task is described in section II, and the second and third tasks are described in section III.

## II. IMPROVEMENT OF THE STATE OF THE ART METHODS FOR PRECISE ROI DETECTION

### A. Background

This section explains the improvement of the state of the art methods for precise ROI detection with landmark detection. The steps of the remote PPG method for HR measurement are:

1) ROI detection. Some approaches are VJ face detector with KLT tracking [5] [6] and skin pixel classification [7].

2) Image processing based on selected ROI (spatial averaging, etc) to get the RGB signals.

3) Signal channels selection and combination to get the pulse signal. The digital camera sensor has RGB color channels. Different approaches have been proposed for color channel selection, such as PCA [8], Green and Chrominance [9].

4) Denoise: the pulse rate of a healthy subject falls within the frequency range of 40 to 240 beats per minute. The parts that are not in this range should be eliminated. Band pass filter [10] can be used in this step.

5) The computation of HR from BVP. The peak detection and processing is used to get the HR.

To get the best method, HR data of different ROI detection methods with different channel selection methods were evaluated.

### B. State of The Art

The state of the art of remote PPG method for ROI detection are VJ face detector [5] with KLT tracking [6] and skin pixel classification [7]. The RPG channel selection methods are Green, PCA [8] and Chrominance [9].

*1) ROI Detection:* there are two state of the art methods:

**VJ Face Detector with KLT Tracking.** The VJ Face Detector [5] is a machine learning algorithm for the human face detection. This detector is able to achieve high detection rates with the fast speed of image processing. KLT feature tracker [6] is an algorithm for feature extraction. Once the VJ Detector finds the face, the identified feature points are stably tracked. This approach utilized the method proposed by Shi & Tomasi [11].

**Skin/Non-skin Pixel Classification.** There exists many skin detection methods. O Conaire et al. proposed an algorithm which adaptively chooses the thresholds for foreground detection for multi-spectral video frames in order to maximize the mutual information between the foreground maps of visual and thermal infrared images [12]. A dynamic programming algorithm is described to efficiently investigate the search-space of all possible pairs of thresholds.

Figure 1 shows the face detection and skin detection (O Conaire method) of a volunteer. The black parts are the pixels that were eliminated in the skin classification, where it can be seen that the eyes and the background are discarded.

Fig. 1. The face detection (left picture) and skin classification (right picture) of a volunteer

*2) PPG Signal Selection and Combination:* The optical absorption of hemoglobin varies across the light spectrum [13], so different approaches have been proposed for color channel selection and combination: PCA, Green and Chrome.

**Green Channel.** The wavelength which is around 560 nm has the highest absorption coefficient [13]. The wavelength of 560nm is the green channel. So the most straightforward way to is to select the green channel.

**Principal Component Analysis (PCA).** Lewandowska et al. proposed a robust method of measuring the pulse rate from human face image captured by a webcam [14]. The desired signal was measured by proper channel selection with PCA processing, instead of the independent component analysis (ICA) processing. The experiment was done by comparing PCA method with the ICA method. The result indicated that the PCA can achieve similar accuracy with ICA method and with much less computation time. This is very important since computational efficiency is taken into consideration if the experimental time is limited.

**Chrominance.** De Haan and Jeanne proposed a method by using the chrominance signals [9]. Their contribution is proposing the "skin-tone standardization", where the normalized skin tone, [R,G,B]/$\sqrt{R^2+G^2+B^2}$, is supposed to be the same for all under the white light $[R_s, G_s, B_s]$ = [0.7682, 0.5121, 0.3841].

And the $[R_s, G_s, B_s]$ are the standardized skin tone estimated coefficients. They have done a lot of experimental tests, and the signals can be represented as

$$S = 3(1-\frac{\alpha}{2})R_f - 2(1+\frac{\alpha}{2})G_f - \frac{3\alpha}{2}B_f \tag{1}$$

where $\alpha = \frac{\sigma(X_f)}{\sigma(Y_f)}$. $\sigma(X_f)$ and $\sigma(Y_f)$ are the standard deviation of $X_f$ and $Y_f$. $X_f$ and $Y_f$ are the band-pass filtered version of $X_s$ and $Y_s$:

$$\begin{aligned} X_s &= 3R_n - 2G_n \\ Y_s &= 1.5R_n + G_n - 1.5B_n \end{aligned} \tag{2}$$

where R,G,B is the RGB value of a pixel. By this chrominance investigation, they improved the motion robustness.

*3) The Computation of The HR:* After the PPG channel selection, the BVP signal can be obtained. As mentioned before, the Butterworth Bandpass Filter [10] was used to discard the frequency range of 40 to 240 beats per minute of the BVP. After this processing, the beats of the BVP can be found by the Peak Detection with regard to the time. Then the HR was computed by frequency analysis of BVP beats. Apply the Fast Fourier Transform (FFT) of the BVP signal and get the Power Spectral Density (PSD). Detect the peak of the PSD. The frequency of the peak is considered as the precise HR. For instance, if the frequency of the peak of the PSD is 1.5HZ, the HR is 1.5 $\times$ 60 = 90/minute.

### C. Methodology

The face detector and skin detection are not fully satisfying. The face detector with KLT tracker cannot discard the unwanted pixels. For the skin classification, the threshold parameter needs to be manually set. So the facial landmark detection is used to improve the state of art methods for ROI detection. Many researchers have proposed different approaches for the landmark detection. For the task, we used the algorithm proposed by Kazemi et al. [15], which was implemented within the Dlib library [16]. This Dlib implementation was selected for the test, because it processes much faster than other implementation, and it provides the facial contour that many other implementations do not have. Without the face contour, a proper ROI is not possible to be obtained. Figure 2 shows the landmarks and the ROI of the volunteer. The black parts are the pixels that are eliminated. It can be seen that the background, eyes and the mouth were discarded.

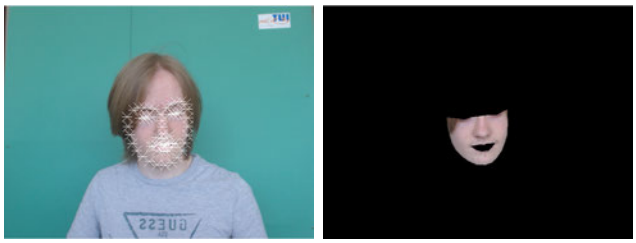


Fig. 2. The facial landmarks and the desired ROI of a student

### D. Experiment

About 55 Bachelor students and PhD students from IUT and LE2I lab in Dijon were organized to attend the experiment. These volunteers were made to sit one meter before the webcam. The focal length of the webcam was adjusted to get a clear image. The index finger was asked to be stably put into the contact PPG sensor to get the ground truth. The experiment was done with good ambient light. For each student, a video of about one minute was recorded.

### E. Results and Discussion

*1) The Evaluation Metrics:* There were 55 videos to process. And for each sample, there are three ROI detection methods with three signal selection methods, which is $3 \times 3$ = 9 tests for one video. There were $55 \times 9$ = 495 tests for the entire dataset. To evaluate the performance of the remote PPG method compared with the ground truth, some evaluation metrics are used:

- Pearson Correlation Coefficient R. This feature is used to evaluate the correlation between the remote PPG result with contact PPG measurement.
- MAE. The Mean Absolute Error (MAE) is the average of the absolute differences between the estimated values and the ground truth where all individual differences have equal weight.
- MAE5 is the MAE which discards all the outliers with an error bigger than 5.
- Mean5 is the ratio of the number of the non-outliers over the total number. The point with an error bigger than 5 is considered as an outlier.
- Mean2.5 is the ratio of the number of the non-outliers over the total number. The point with an error bigger than 2.5 is considered as an outlier.
- RMSE5. RMSE is Root Mean Square Error. It is a quadratic scoring rule that measures the average magnitude of the error. It is the square root of the average of squared differences between prediction and actual observation. Like the MAE5 feature, RMSE5 feature considered the points with the error bigger than 5 as outliers.
- SSE is the sum of the squared errors of the prediction function.
- MeanSNR is the average of the signal-to-noise ratio.

Calculate the average metrics value of 55 samples for each method, and the result is shown in table I. According to the table, the skin detection with chrom method is the best of all. The SSE, MAE, MAE4 and RMSE5 of the Skin/Chrom methods are the lowest. It has the highest Mean2.5 and Mean5 of all, which means it has very few outliers. The landmark detection with Chrom also have a quite low MAE, MAE4 ,RMSE5 and SSE, but it has a lower Mean2.5 and Mean5. This means the outliers of the landmarks detection are more than the skin detection with Chrom method. The correlation R of Skin/Chrom combination is closer to 1 than other methods, which means the correlation of Skin/Chrom methods between the remote method and the ground truth is very high. The R of landmark/Chrom method is worse than skin/Chrom. The mean SNR of the Skin/Chrom method is also the best, which means the performance of the remote PPG is very good in the frequency domain.

The landmark detection is not as perfect as expected. A possible reason of this may be that the landmark detection cannot eliminate the hair and beard. It is possible to get better ROI detection by improving the facial landmark detection such as adding the forehead and neck and discarding the hair and beard, or implementing a mixed ROI detection by combing the landmarks detection with the skin classification. This will be the future work.

## III. Remote Emotion Detection

### A. Background

The Power Spectral Density (PSD) of the HRV can be obtained by utilizing Fast Fourier Transform (FFT) of the HRV signal. This PSD data is used to distinguish the sympathetic activity and parasympathetic activity of total ANS activity. The power spectrum is divided into three main frequency ranges. The very low frequency range (VLF) (0.0033 to 0.04 Hz), representing slower changes in HR, is an index of sympathetic activity, while power in the high frequency range (HF) (0.15 to 0.4 Hz), corresponding to the faster changes in HR, is primarily caused by the parasympathetic activity. The frequency range around the 0.1 Hz regions is called the low frequency (LF) band and is also often referred to as the baroreceptor band, because it corresponds to the feedback signals of blood pressure sent from the heart back to the brain, and at the same time it affects the HRV waveform as well. The LF band is more complex, because it can reflect a mixed activity of the ANS parasympathetic branch and the ANS sympathetic branch. It has been shown in many studies that during mental or emotional stress, the sympathetic activity increases and the parasympathetic activity decreases. While in the relaxed state, the parasympathetic activity increases [17].

### B. State of The Art

The human emotion can be detected in different ways. The HRV based on ECG (and contact PPG recently) is widely used. As explained before, the components of the HRV in frequency domain can be used to detect the human emotion stimulation such as stress. The VLF and LF components will increase with these emotions, while the HF component will increase when a person is calmed down and relaxed. The ratio of LF/HF is widely used for emotion detection [18].

TABLE I

THE AVERAGE EVALUATION VALUES OF EACH COMBINATION FOR ROI DETECTION

| The Methods | SSE | R | Mean2.5 | Mean5 | MAE | MAE5 | RMSE5 | MeanSNR |
|---|---|---|---|---|---|---|---|---|
| Chrom/Skin | 1.84 | 0.80 | 0.72 | 0.93 | 2.88 | 1.48 | 3.81 | 4.63 |
| Chrom/Face | 2.33 | 0.61 | 0.64 | 0.84 | 5.82 | 1.58 | 8.61 | 1.97 |
| Chrom/Landmark | 2.21 | 0.72 | 0.70 | 0.91 | 3.22 | 1.54 | 4.98 | 3.08 |
| PCA/Skin | 2.41 | 0.48 | 0.58 | 0.76 | 8.61 | 1.72 | 11.53 | 0.49 |
| PCA/Face | 3.17 | 0.14 | 0.36 | 0.50 | 20.98 | 1.64 | 24.73 | 3.16 |
| PCA/Landmark | 3.31 | 0.31 | 0.45 | 0.63 | 13.36 | 1.85 | 17.66 | 1.56 |
| Green/Skin | 2.36 | 0.46 | 0.60 | 0.79 | 7.73 | 1.62 | 10.83 | 1.01 |
| Green/Face | 2.88 | 0.30 | 0.46 | 0.61 | 15.05 | 1.62 | 18.33 | 1.77 |
| Green/Landmark | 3.29 | 0.38 | 0.46 | 0.62 | 12.65 | 1.71 | 16.53 | 2.74 |

The electrodermal activity (EDA) was also used for emotion detection. Some results showed that the EDA can reflect the emotion activities within the same subject [19]. But the data of EDA cannot be obtained with a remote method.

### C. Methodology

The ratio of LF/HF of HRV has been widely used for emotion detection. In this paper, the (VLF + LF)/HF is proposed as a new feature to assess the emotion, because the VLF component can also reflect the sympathetic activities. There are two works for this part: the precision analysis of remote HRV and the remote emotion detection.

*1) The Computation of HRV in Time Domain/Frequency Domain:* The HRV is measured by the variation in the beat-to-beat interval. The previous section has explained that the HR is calculated by applying FFT and considering the frequency of the peak of the PSD curve as the HR. The HRV and its frequency components can also be computed from BVP. There are four steps: 1. Detect the peaks of the BVP. 2. Calculate the difference in time between the peaks. 3. Interpolate the signal with the frame number and get the HRV in time domain. 4. Apply FFT on the HRV and get the PSD of the HRV in frequency domain. The signal is interpolated with the frame number, so that the differences of the time coordinates keep constant which is necessary for the FFT processing.

*2) From HRV to Human Emotion Detection:* The ratio of LF/HF of HRV is the state of the art for emotion detection. In this paper, the (VLF + LF)/HF is proposed as the new feature to assess the emotion, because the VLF component can also reflect the sympathetic activities. Both LF/HF and (VLF + LF)/HF are studied to check if these values are distinguishable in different emotion states.

### D. Experiment

There are two tasks of the experiments:

**The Analysis of The HRV.** Although the skin classification with Chrominance method can achieve high accuracy of BVP and HR measurement, the experiments have proved that the HRV measurement is much more sensitive than HR measurement and more difficult to be obtained remotely. So a new experiment with higher precision should be done to assess the precision of the remote HRV. This experimental set up and procedures are similar with that of the ROI detection, except that: 1. The GO camera was used instead of the previous webcam. This GO camera has a USB 3.0 connection, so the resolution can be set to 1024 $\times$ 786, which is much higher than previous webcam's resolution. 2. Two direct current (DC) light sources were used instead of the ambient light. These light sources have higher brightness intensity than the ambient light.

**The Human Emotion Recognition.** In this study, the negative external stimulation such as stress and anxiety are expected to be detected. The experimental set up is the same with the HRV analysis. Every volunteer had two different tests and each test lasted two minutes: 1. The "anxiety/stress" test: a video with strong "negative emotion" was played and shown to the person during the test. And this is considered as the "anxiety" or "stress" sample. This video can be the clips of horror films, the CCTV videos of traffic accidents, disturbing images with noisy music, etc. 2. The "normal/relaxed" test: make the person look at the camera and relax.

### E. The Evaluation of The Experimental Data

**The HRV Analysis.** Previous section has explained that the VLF, LF and HF components of the PSD are the metrics to detect the human emotion. So instead of using MAE and MeanSNR (proposed in section II), the accuracy of HRV is evaluated by computing the integral of power density of VLF, LF and HF components of the PSD curves and comparing them with the ground truth, which are much more clear for this specified research. The equations are:

$$\begin{aligned} VLF_{accuracy} &= 1 - \frac{|rVLF_{PSD} - VLF_{PSD}|}{VLF_{PSD}} \\ LF_{accuracy} &= 1 - \frac{|rLF_{PSD} - LF_{PSD}|}{LF_{PSD}} \\ HF_{accuracy} &= 1 - \frac{|rHF_{PSD} - HF_{PSD}|}{HF_{PSD}} \end{aligned} \tag{3}$$

Where $rVLF_{PSD}$ is the integral of the VLF component of the remote method, $VLF_{PSD}$ is the integral of the VLF component of the contact method (ground truth), and so forth.

**The Negative Emotion Detection.** The conventional LF/HF ratio and the proposed (VLF+LF)/HF ratio are computed respectively for comparisons of different emotion states. The mean values of the LF/HF ratio and

(VLF+LF)/HF ratio are computed for different emotion states respectively.

### F. Results and Discussion

*1) The Precision Analysis of HRV measured by Remote PPG (rPPG):* The HRV in time/frequency domain computed from the remote PPG is compared with the HRV from the contact PPG. Figure 3 showed an example of the comparison of HRV computed from remote PPG and contact PPG in time domain. It can be observed that the result is very good. The HRV curves of remote method match very well with the contact method. The most important part is the frequency domain, which can be used to detect the human emotion. Figure 4 showed an example of the HRV between remote PPG and contact PPG in frequency domain. It can be seen that the HRV in frequency domain also match very well.

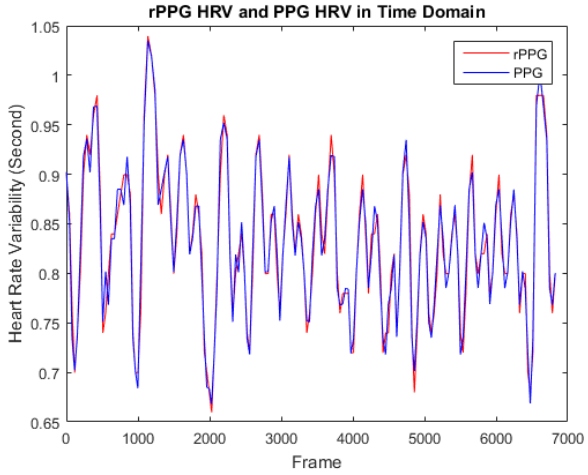



Fig. 3. The HRV between remote PPG (rPPG) and contact PPG in time domain

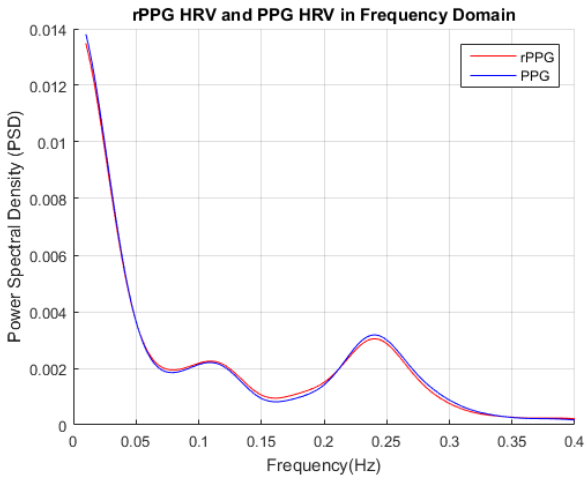



Fig. 4. The HRV between remote PPG (rPPG) and contact PPG in frequency domain

Table II shows the precision of the frequency components of HRV compared with the ground truth based on the evaluation method proposed in the previous section. It can be seen that accuracy of the VLF are almost 100%, and this is because the VLF component is too small and the errors are also small enough to ignore. The other components also have very high accuracy, and most of them are above 95%. It can be concluded that the quality and the precision of HRV obtained by remote method are very good. And this is the foundation of the emotion detection.

TABLE II
THE ACCURACY OF HRV OF DIFFERENT COMPONENTS IN FREQUENCY DOMAIN

| **The Frequency Component** | **VLF (%)** | **LF (%)** | **HF (%)** |
|---|---|---|---|
| Subject1 | 99.99 | 97.29 | 98.14 |
| Subject2 | 99.99 | 98.99 | 98.21 |
| Subject3 | 99.99 | 98.33 | 94.91 |
| Subject4 | 99.99 | 94.75 | 91.23 |
| Subject5 | 99.99 | 98.84 | 98.59 |
| Subject6 | 99.99 | 99.16 | 98.91 |
| Subject7 | 99.99 | 98.62 | 97.28 |
| Subject8 | 99.99 | 95.30 | 93.42 |
| Subject9 | 99.99 | 99.95 | 99.72 |
| Subject10 | 99.99 | 95.30 | 93.73 |
| Subject11 | 99.99 | 97.29 | 98.14 |

*2) The Emotion Detection:* Figure 5 and 6 showed the LF/HF feature and (VLF + LF)/HF feature of the volunteers in different emotion states. 13 participants of the total 16 (81.25%) have higher values of both features in the negative emotion state than the relaxed state. These results prove the theory that the VLF and LF reflects the sympathetic activities and HF reflects the parasympathetic activities and also prove that these features are distinguishable in different emotion states with remote PPG method.

We didn't normalize the data to be make them distinguishable in different subjects. This will be the future work. Results of 3 participants did not perform very well, and this is because they were not really relaxed during the "relaxed states" of the experiments due to the lack of skills and experience in psychological experiments. Improving the experimental set up for creating desired emotion states will be the future work as well.

## IV. CONCLUSIONS AND FUTURE WORK

There are two main sections in this paper: 1) Improvement of the precise ROI detection with landmark detection. 2) HRV based remote emotion detection. For the first part, the state of art methods for ROI detection, namely VJ face detector with KLT tracking and the skin/non-skin classification, were compared with the proposed landmark detection. The signal selection methods such as Green, PCA and Chrominance were also implemented and compared. Experiments were done with a cheap webcam and a contact PPG as the ground truth. The results showed that the skin detection with Chrominance channel selection has the highest accuracy and fewest outliers among all the methods. The facial landmark detection with chrom is not bad, but the error is bigger than the Skin/Chrom method and has more outliers. The second

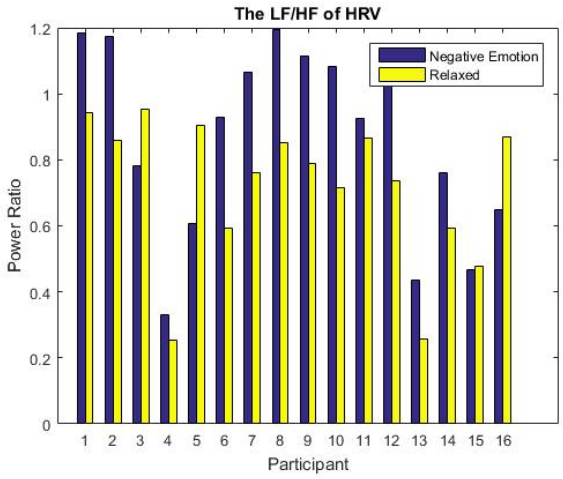



Fig. 5. The individual LF/HF in different states

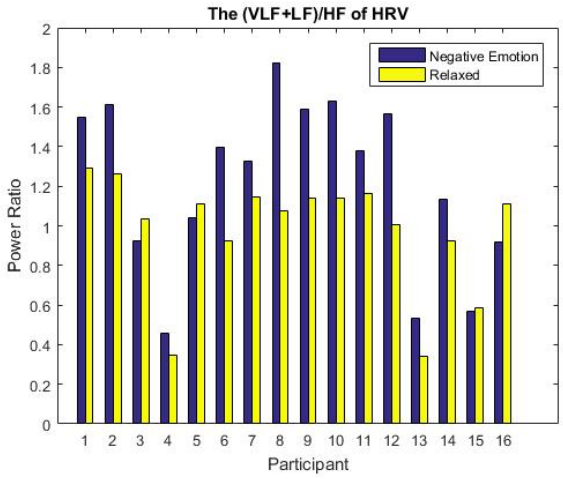



Fig. 6. The individual (VLF+LF)/HF in different states

part discussed the HRV precision and remote human emotion detection. A new experiment with better camera and DC light sources was set up for the sake of higher precision. The results showed that very good HRV can be obtained based on the Skin/Chrom method and the careful experimental set up, and the accuracy of the VLF, LF and HF of most of the samples were over 95%. The remote emotion detection was done by creating two different emotion states for the volunteers. The negative emotion states were stimulated by playing some disturbing videos and noise, and the normal states were created by making them sit and relax. The new feature (VLF+LF)/HF ratio was proposed and calculated as well as the conventional LF/HF ratio. The results showed that 13 samples of 16 (81.25%) have higher values in both LF/HF and (VLF+LF)/HF during negative emotion state and have lower values in relaxed state. These frequency features of HRV are distinguishable with remote PPG method, which proved that the remote emotion assessment can be achieved with these features.

Lots of work can be done in the future. It is possible to get better ROI detection by improving the facial landmark detection such as adding the forehead and neck and discarding the hair and beard, or implementing a mixed ROI detection by combining the landmark detection with the skin classification. The experiments of the emotion detection can be improved by using a bed and making the people lie on the bed to fully relax, and organizing a competition for the volunteers to put more effective stress. The data can be normalized so that they are separable between different subjects.

## REFERENCES


[1] Babchenko, A. *et al.* Photoplethysmographic measurement of changes in total and pulsatile tissue blood volume, following sympathetic blockade. *Physiological measurement* **22**, 389 (2001).

[2] Jones, J. W., Glassford, E. J. & Hillman, W. C. Remote monitoring of free flaps with telephonic transmission of photoplethysmograph waveforms. *Journal of reconstructive microsurgery* **5**, 141–144 (1989).

[3] Quintana, D. S., Guastella, A. J., Outhred, T., Hickie, I. B. & Kemp, A. H. Heart rate variability is associated with emotion recognition: direct evidence for a relationship between the autonomic nervous system and social cognition. *International Journal of Psychophysiology* **86**, 168–172 (2012).

[4] Boissy, A. *et al.* Assessment of positive emotions in animals to improve their welfare. *Physiology & Behavior* **92**, 375–397 (2007).

[5] Viola, P. & Jones, M. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, I–I (IEEE, 2001).

[6] Lucas, B. D., Kanade, T. *et al.* An iterative image registration technique with an application to stereo vision (1981).

[7] Henriques, J., Caseiro, R., Martins, P. & Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. *Computer Vision–ECCV 2012* 702–715 (2012).

[8] Lewandowska, M. & Nowak, J. Measuring pulse rate with a webcam. *Journal of Medical Imaging and Health Informatics* **2**, 87–92 (2012).

[9] de Haan, G. & Jeanne, V. Robust pulse rate from chrominance-based rppg. *IEEE Transactions on Biomedical Engineering* **60**, 2878–2886 (2013).

[10] Butterworth, S. On the theory of filter amplifiers. *Wireless Engineer* **7**, 536–541 (1930).

[11] Shi, J. *et al.* Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, 593–600 (IEEE, 1994).

[12] Conaire, C. O., O'Connor, N. E. & Smeaton, A. F. Detector adaptation by maximising agreement between independent data sources. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–6 (IEEE, 2007).

[13] Zijlstra, W. G., Buursma, A. & van Assendelft, O. W. *Visible and near infrared absorption spectra of human and animal haemoglobin: determination and application* (VSP, 2000).

[14] Lewandowska, M., Rumiński, J., Kocejko, T. & Nowak, J. Measuring pulse rate with a webcam a non-contact method for evaluating cardiac activity. In *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, 405–410 (IEEE, 2011).

[15] Kazemi, V. & Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1867–1874 (2014).

[16] King, D. E. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research* **10**, 1755–1758 (2009).

[17] McCraty, R. Science of the heart: Exploring the role of the heart in human performance [null]. *Boulder Creek, CA: HearthMath Research Center* (2001).

[18] McDuff, D., Gontarek, S. & Picard, R. Remote measurement of cognitive stress via heart rate variability. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, 2957–2960 (IEEE, 2014).

[19] Kim, K. H., Bang, S. W. & Kim, S. R. Emotion recognition system using short-term monitoring of physiological signals. *Medical and biological engineering and computing* **42**, 419–427 (2004).

# Structure from motion aided motion segmentation for camera Advanced Driver Assistance System (ADAS)

Yannick PORTO and Sebastian Carreno

***Abstract*— This paper focuses on *motion segmentation* helped by 3D information obtained from *structure from motion* in the context of camera ADAS. First a review of the main motion segmentation methods for moving cameras is presented. Afterwards, an approach is proposed in three steps: *detection*, *segmentation* and *tracking*. Moving points are detected according to their deviation from the static 3D point constraints. Then a *segmentation* is proposed and discussed with two different methods : one through Mean-Shift clustering and an other one through Graph-cuts segmentation. Bounding-boxes are formed around the segments and tracked using a particle filter. A new dataset is realized from a rear fish-eye camera embedded on a car in order to experiment and get results for the described approaches. The thesis concludes with an analysis of the pros and cons of these approaches and future work suggestions.**

**Keywords : motion segmentation, structure from motion, static 3D point constraints, mean-shift, graph-cuts, particle filter**

## I. INTRODUCTION

Advanced Driver Assistance Systems (ADAS) make use of computer vision techniques in order to help drivers and increase safety on roads. They are used to identify lanes, recognize traffic signs or detect obstacles, among many other functionalities. Real time computer vision solutions capable of perceiving the 3D environment around the vehicle are now embedded inside ADAS. Reconstruction can be performed using many accurate sensors but the request of reducing cost is made. Therefore structure from motion (SfM) is needed to get 3D information of the surroundings of the car from monocular cameras.

SfM allows to get time-to-collision to static obstacles but fails in measuring a correct estimation of distance to moving objects. However, their detection is important for collision avoidance, especially in parking scenarios where objects or living-beings can appear behind the vehicle. That is why motion segmentation have to be performed on the 2D image which is provided by the camera. Motion segmentation allows to know which object in movement a point or a pixel is belonging to by assigning a label based on its features. Information coming from a SfM algorithm can help for the segmentation.

The output is to detect any kind of objects or living-beings moving behind the car, and to provide their central position in the 2D frame as well as their respective size. Figure 1 shows the ground-truth of the dataset, presented in section IV, which is the target of the algorithm that needs to be implemented. The choice of a rear fish-eye camera is made because of its large field of view which can cover all the space behind the vehicle. The pre-calibrated fish-eye camera and a CAN sensor provides the information about the scene and the vehicle movements. So intrinsic and extrinsic camera parameters are given before-hand, and one can get undistortion of the image and transformation from vehicle coordinates system to camera coordinates system from these parameters. A SfM algorithm is provided too, from which tracks in 2D and 3D can be obtained.

Fig. 1: Ground truth of the recorded dataset, showing the bounding boxes encapsulating the moving objects.

## II. RELATED WORK

Motion segmentation has been extensively studied during past decades, starting from 2D visual information to techniques using now 3D cloud of points. Background subtraction methods appeared to be successful in the context of static cameras but fail whenever there is an ego-motion. So most of the other previous works rely on the computation of an optical flow.

Motion segmentation using 2D data can be classified in three different categories, which are: the detection of moving objects with sparse interest points, the analysis of dense motion, or the use of semantic segmentation refining the flow. Sparse moving points are first detected thanks to the estimation of the ego-motion, which is the 3D motion of a camera within an environment. Then they can be clustered according to their motion, color, or other features, as in [2], [7]. Dense flow segmentation like [1] or [11] provides a smooth colormap for the motion of all the pixels in the image allowing the distinction of each moving objects. It is accurate but requires extensive computation time. In parallel, semantic segmentation can provide the type of the object we are trying to segment in order to apply a flow suitable to this

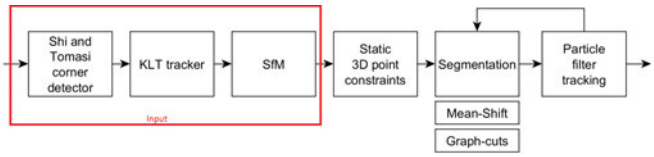



Fig. 2: Overview of the proposed method.

type, what leads to really precise delineation of the objects [9].

On the other hand, motion segmentation can be applied on 3D data. Works of Franke et al. [3], [8] have brought interesting solutions for ADAS using stereo-cameras. A multi-Kalman filter based approach is designed including 3D position and velocity of points as well as inertial sensor information for ego-motion estimation. Errors in the 3D flow vectors are corrected by this way and groups of similar motion can be formed. Other works [6] connect points in the scene flow by delaunay triangulation and discard some edges according to the motion difference of their corresponding vertices in the graph. In the domain of 3D data coming from monocular cameras, detection of moving points has been studied with the constraints of static 3D points. The deviation of a point to these constraints allows to apply thresholding, in order to know if this point is moving or not, or to serve as feature for a segmentation algorithm [5]. The choice of an affine camera model can be made too for 3D reconstruction with structure-from-motion so that subsplace-based motion segmentation can be applied. All point trajectories are embedded inside a single matrix from which subspaces corresponding to common motions are formed. It is realized by assuming the motion of each one of the rigidly moving objects living in a four-dimensional subspace maximum. The number of moving objects is obtained by projecting the data onto an optimal number of dimension, with Singular Value Decomposition (SVD) for instance, then segmentation can be realized with spectral clustering [10], [12].

## III. METHODOLOGY

The developed and proposed approach is a combination of three main steps which are: detection of moving points using static 3D point constraints following [4], point-wise and pixel-wise segmentation using Mean-Shift and Graph-cuts, and tracking of bounding-boxes with a particle filter. An overview of the described approach can be found in Fig. 2.

### A. Pre-computed input

Tracks of interest points in 2D and 3D are provided beforehand for the segmentation. Shi and Tomasi corner detector is used to pick up points in the image which are then tracked by a Kanade-Lucas-Tomasi feature tracker. A SfM algorithm provides the 3D position of these points in the vehicle coordinates.

This corner detector and tracker is good in the sense that it provides numerous points in the image which are easy to track accurately. The bad side is that we often have more points belonging to the background than to the objects since images of cars or pedestrians contain important homogeneous regions whereas ground planes contain a lot of corners. So the proposed algorithm needs to cope with really few points describing the objects in a really wide set of points picked on the background.

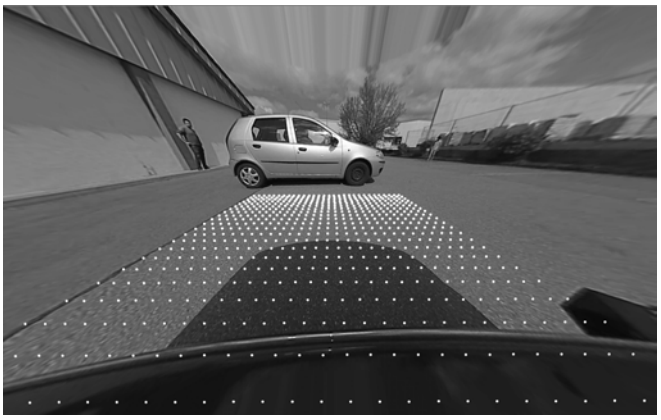


Fig. 3: Re-projection of 3D points lying on the ground to the image plane using the computed rotation and translation matrix.

### B. Camera positions

Camera position from frame to frame and relatively to the ground is important for detecting points which are moving.

We therefore need the fundamental, rotation and translation matrices of the camera, relating the previous pose of the camera to the current one. For that, un-distortion of the image is applied thanks to the calibration parameters. The fundamental matrix is computed with a 8 points algorithm followed by RANdom SAmple Consensus (RANSAC) in order to take as less outliers as possible, since we will have to detect these ones later on. Rotation and translation of the camera are obtained by applying SVD on the essential matrix, which is computed from the fundamental and intrinsic matrix, and cheirality constraint.

Rotation and translation between the ground plane and the camera are found with Perspective-n-Point problem. Since we have the corresponding points representing the road in 3D and on the 2D frame, we can obtain a non-linear relation between them which can be solved by Direct Linear Transform (DLT). Fig. 3 shows the re-projection of a generated 3D pattern with coordinates lying on the ground after the computation of the rotation and translation matrices relating the camera and the ground plane.

### C. Static 3D point constraints

When reconstructing a scene in 3D from multiple views, not all the points can be taken into account; there are always inliers and outliers. Inliers, which are 3D static points, have to follow some constraints. If a point is too far from these constraints, it is defined as outlier. This is how motion aliasing is usually detected in 3D reconstruction and how points which are not reliable are discarded. In our scenario,

it is the opposite, we are exactly interested by these points, which can be real outliers or points on moving objects. The deviation of a point from these constraints constitutes therefore a feature for determining if a point is belonging to a moving object or not. Three different constraints are used to detect the moving points: the epipolar constraint, the depth constraint and the height constraint.

The fundamental matrix relates two points in a rigid scene by the equation :

$$\mathbf{x}'\mathbf{F}\mathbf{x} = 0. \tag{1}$$

The **epipolar constraint** states that the correspondence $\mathbf{x}'$ of a static point $\mathbf{x}$ should lie on the epipolar line $\mathbf{F}\mathbf{x}$. The distance from the point $\mathbf{x}'$ to the line $\mathbf{l'_e} = \mathbf{F^T}\mathbf{x}$ constitutes a first measure $d_e$ for the detection of the moving points.

The **positive depth constraint** is the fact that all points seen by the camera must lie in front of it. If a 3D point is going faster than the vehicle on which the camera is mounted, the triangulation of its corresponding points in the 2D frames will result in a 3D point measured behind the camera, what violates the constraint. The detection of overtaking traffic is therefore possible with this constraint.

In the same manner, the **positive height constraint** states that all 3D points must lie above the road. If a 3D point moves with a smaller distance than the camera movement, it means there is a vehicle moving slower than the ego-vehicle and results in a triangulated 3D point under the road.

A selection of only one of the spatial constraint (depth or height) can be made at computation time by tacking the closest point following the constraints [4]. For that, the position of the given point relative to the horizon line is evaluated. If this point lies under the horizon line, then its corresponding point following the constraints is its projection on the road, possible thanks to the previously computed camera pose. It it lies above the horizon, then the point projected onto the plane at infinity is the closest point following one of the spatial constraint, the depth constraint in this case.

For the spatial constraint, the measure $d_s$ taken into account is the distance to the border line $l_b$, computed in eq. 2. This line is perpendicular to the epipolar line $l_e$ and passing through the border point $x_b$, which is either the point on the road or on the plane at infinity.

$$\mathbf{l_b} = \begin{bmatrix} 0 & (\mathbf{x_b})_3 & 0 \\ -(\mathbf{x_b})_3 & 0 & 0 \\ (\mathbf{x_b})_2 & -(\mathbf{x_b})_1 & 0 \end{bmatrix} \mathbf{F}^T\mathbf{x}', \tag{2}$$

where $(.)_1$, $(.)_2$ and $(.)_3$ mean the first respectively the second or third component of a vector.

### D. Segmentation

Two different methods for segmentation have been implemented and discussed:

- One is considering a segmentation of the detected moving points, by thresholding above a deviation of 2 pixels, in order to group them according to their position and velocity with Mean-Shift.
- The other segmentation is a pixel-wise segmentation with Graph-cuts which is using the metric distance computed in the previous section for spreading the membership of a moving object to the neighboring pixels, as in [5].

*1) Mean-Shift:* This segmentation step will allow us to group the pre-computed moving points according to which moving object they belong to. The main problem is that the number of clusters is unknown beforehand and has to be found on the fly. This is why the clustering method should not take as parameter an initial number of clusters. This is one prerequisite of our motion segmentation target since any number of objects or living-beings can appear behind a car.

Mean shift algorithm is a nonparametric clustering technique which does not require prior knowledge about the number of clusters, and does not constrain the shape of the clusters. If dense regions are found in the feature space, then they are considered as modes and all nearby features are assigned to this region, cluster.

A certain kernel and an associated bandwidth are required for Mean-Shift. A gaussian kernel is chosen but a fixed bandwidth can not handle all the different types of objects since their size and density of points differ a lot from one object to the other. Therefore, an adaptive bandwidth is applied to each point by taking its k-nearest neighbor. $k$ is chosen according to the number of moving points detected.

Position and velocity of the points serve as features for the segmentation since it can at the same time group points which are nearby and also differentiate two different objects which are close and going into a different direction. Bounding boxes are formed by encapsulating all the points in a same cluster.

*2) Graph-cuts:* This segmentation step consists of assigning a label, foreground or background, this time on all the pixels of the frame according to how they are connected to the points showing a deviation from the static 3D point constraints. Graph-cuts method is chosen for the segmentation since it can force some pixels to belong to a region or not and run fast enough. It represents the pixels of a frame and their membership to a label or an other as

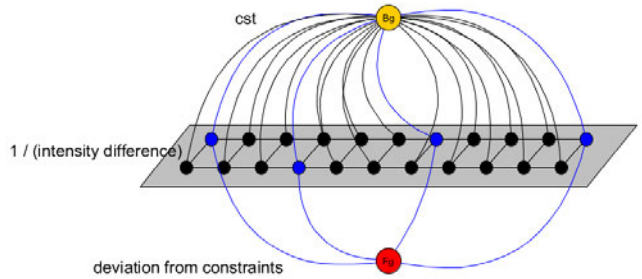



Fig. 4: Weights applied to the edges of the graph. The yellow node represents the source which is the background, the red node represents the sink which is the foreground, the blue nodes are the tracked points and the black nodes all the remaining pixels in the image.

(a) A bicycle and a trolley in a normal scenario

(b) A car in a crossing traffic

(c) Two pedestrians in a preceding traffic

(d) A trolley and two pedestrians in a crossing traffic

Fig. 5: Examples of situations recorded in the dataset.

a graph connected to a source $s$ and a sink $t$ with some weights.

Fig. 4 shows the weights applied to the tracked points and the remaining pixels. The tracked points are connected to the foreground node with a weight corresponding to their deviation from the constraints. All the pixels in the frame are connected to the background with a small constant in order to spread a background prior since more pixels should normally belong to the background than the foreground. The pixels are connected between each other too, with a $4 \times 4$ neighborhood, and the weight applied to these edges is the inverse of their grayscale value difference.

If enough points are detected on a single object, then this one will be properly segmented and we will have the binary mask representing the object. Bounding boxes are drawn here around each disconnected mask.

### E. Bounding boxes tracking

In order to have a memory of the object features over time, we need to track them. First a template model has to be created from the bounding-box detected in the previous section. If we want to get its position in the next frame, the best correspondence to this template needs to be found. The chosen model is a combination of spatial gray-scale and texture distribution build from Local Binary Pattern inserted side by side in the same histogram. The spatial information is obtained by applying an Epanechnikov kernel to the template model.

Matching is then realized with a particle filter, generating a set of particles around the previous target location. This position is inserted inside the state equation as a state vector in order to solve the data association problem. Weights are applied to each particle proportionally to the Bhattacharyya coefficient between the template model and the model formed by the patch around the particle. The weighted mean constitutes the new position of the detected object.

Once a new detected segment is considered as being already tracked, by comparing overlapping areas and similarities between histograms, the model of the tracked bounding box is updated with the one of the new detection with a certain weight.

## IV. EXPERIMENTS AND RESULTS

### A. Experiments

The dataset handled for the evaluation of the segmentation has been captured by us in order to have specific scenarios. It has been captured from a rear fish-eye camera attached to a car delivering images of $1280 \times 720$ pixels. It consists of 15 videos in which different kind of objects (a car, some pedestrians, bicycles, a supermarket trolley, a ball) can be detected while the car is moving. Some of the videos (number 8 to 15) show typical scenes which happen in most of the usual parking lots (example in Fig. 5). Some others try to catch the special cases where the movement of moving objects is either degenerate or non-degenerate. For instance, videos 1 and 2 show two pedestrians and one bicycle crossing behind the car, what are non-degenerate movements. On the contrary, videos 3, 4, 5 and 6 show two pedestrians going in the opposite direction or in the same direction than the embedded camera, what are degenerate cases. The dataset

TABLE I: Results obtained with the point-wise segmentation approach.

| | DR | FR | CMD | SD |
|---|---|---|---|---|
| Crossing traffic | 69% | 36% | 11pxl | 66pxl |
| Preceding traffic | 81% | 18% | 23pxl | 120pxl |
| Normal scenarios | 65% | 41% | 24 pxl | 85pxl |

is therefore split into three categories which are: crossing traffic, preceding traffic, and normal scenarios.

A ground-truth of this dataset is build by labeling manually each frame of the 15 videos. The labels correspond to the bounding boxes encapsulating the different moving objects.

The segmentation is evaluated in term of overlapping area, and difference in size and position. The different measured KPI's are:

- **Detection rate (DR) =** $\frac{NDT}{NT} \times 100\%$.
  Number of detected targets over the number of targets to detect.
- **False detection rate (FR) =** $\frac{NFF}{NF} \times 100\%$.
  Number of false detected frames over the number of frames in the all sequence.
- **Centers of mass distances (CMD) =** $\frac{||CM_{GT} - CM_{DT}||}{NDT}$.
  Euclidean distance between the center of mass of the ground-truth and the detected bounding box over the number of detected targets.
- **Size differences (SD) =** $\frac{||size_{GT} - size_{DT}||}{NDT}$.
  Difference between sizes of bounding box of ground-truth and detected moving object over the number of detected targets .

where :

- **NF:** number of frames
- **NT:** number of targets that need to be detected
- **NDT:** number of correct detected targets, which occur when a segment overlaps the ground-truth region with less than $30\%$ error.
- **NFF:** number of false detected frames, which are the frames where $\geq 1$ target is not correctly detected.

### *B. Results*

The approaches explained before have been implemented in C++ with some parts in C code so that they can be later on re-used for embedded systems. The algorithm is run on a Intel Core i7-4810MQ CPU of 2.80GHz. Results have been observed first with the point-wise segmentation approach where encapsulation of the points belonging to the same clusters forms the bounding boxes. Table I shows the results obtained with this method. Due to the lack of points from the optical flow on the moving objects, Mean-Shift can not include some part of the objects or pedestrians in the segments, what leads in some cases to big errors in the overlapped region and size difference. However, when enough points are provided, usually more than 5 for an object, the segments overlap the correct area and show a detected center of mass not to far from the ground-truth region.

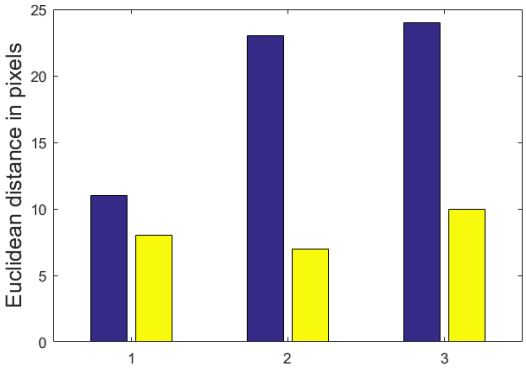



Fig. 6: Comparison of **CMD** results using both approaches: point-wise segmentation in blue and pixel-wise segmentation in yellow. The results are shown for the three types of sequence.

Big differences are observed between the scenarios because of the movements and types of object to detect. Preceding traffic scenarios show best results in term of **DR** because object position do not change dramatically from frame to frame when it is following or running away from the camera. These scenarios have been realized with pedestrians, what explains an important **SD** compared to the ground truth due to the fact that a missing foot or a missing head can affect heavily the segmentation. Crossing traffic and normal scenarios show similar results because they contain similar types of objects, which are pedestrians, cyclists, cars and trolleys. The performances of the algorithm in term of **DR** and **FR** decreases because of the number of objects to detect which can vary from 2 to 5 in the same scene and it can loose the track of one of them during occlusion or absence of movement.

Results have been observed for the other approach including a pixel-wise segmentation but this approach suffers even more from lack of points and need at least 10 points to segment an object correctly. We therefore observe the results in term of precision when a target is detected compared to the other approach. Fig. 6 and 7 show the improvement in terms

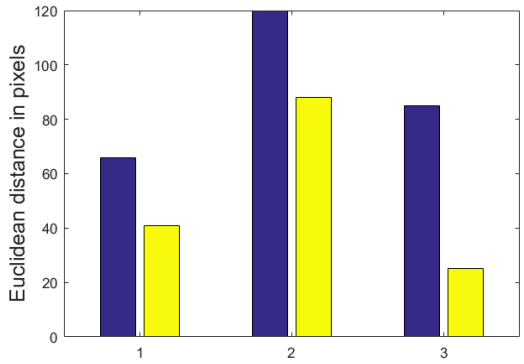



Fig. 7: Comparison of **SD** results using both approaches: point-wise segmentation in blue and pixel-wise segmentation in yellow. The results are shown for the three types of sequence.

TABLE II: Average processing time per frame needed for both approaches, from the detection of moving points until the tracking step.

| | Point-wise | Pixel-wise |
|---|---|---|
| Processing time | 30ms | 500ms |

of position and size respectively where the deviation from the ground-truth in pixels is divided by 2. A comparison of the processing time needed for both approaches can be observed too in table II. The program needs in average around **30ms** to terminate for the point-wise segmentation. The camera used for the recording has a frame rate of 30fps, so we can achieve a segmentation of the moving objects in real time with this approach, by not considering the time needed for the computation of the structure from motion. The pixel-wise segmentation show a processing time much more important with an average of 500ms.

## V. CONCLUSIONS AND FUTURE WORKS

### A. Conclusions

In this paper, two motion segmentation methods have been developed for detecting moving objects with the help of structure from motion. Both approaches success to obtain bounding boxes from few points provided for describing moving objects out of a large set of points belonging to the background. This study proves that although the motion of points induces error in the SfM process, this one can be used as a measure for motion segmentation. The first approach shows that this measure can help a real-time segmentation algorithm which can be embedded inside a car for collision avoidance. The second approach demonstrates that the accuracy of the segmentation can be improved by moving the segmentation to a pixel level, but decreases at the same time the computation speed.

### B. Future Works

Future works can be done by providing more points for describing the objects, what can improve the results consistently. Determining the speed of the object and computing an approximative position in the 3D world is a further step for collision avoidance. For that, the detected object position on the ground plane could be studied, so as the estimation of its scale, or even the measurement of the free space behind the car. This would allow the driver to know the time-to-collision to the static world as well as to the moving objects.

## REFERENCES

[1] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, pages 282–295, Berlin, Heidelberg, 2010. Springer-Verlag.

[2] Aurélie Bugeau and Patrick Pérez. Detection and segmentation of moving objects in complex scenes. *Comput. Vis. Image Underst.*, 113(4):459–476, April 2009.

[3] Uwe Franke, Clemens Rabe, Hernán Badino, and Stefan Gehrig. 6d-vision: Fusion of stereo and motion for robust environment perception. In *Proceedings of the 27th DAGM Conference on Pattern Recognition*, PR'05, pages 216–223, Berlin, Heidelberg, 2005. Springer-Verlag.

[4] J. Klappstein, F. Stein, and U. Franke. Monocular motion detection using spatial constraints in a unified manner. In *2006 IEEE Intelligent Vehicles Symposium*, pages 261–267, 2006.

[5] Jens Klappstein, Tobi Vaudrey, Clemens Rabe, Andreas Wedel, and Reinhard Klette. Moving object segmentation using optical flow and depth information. PSIVT '09, pages 611–623, 2008.

[6] Philip Lenz, Julius Ziegler, Andreas Geiger, and Martin Roser. Sparse scene flow segmentation for moving object detection in urban environments. In *in IEEE Intelligent Vehicles Symposium, 2011*, pages 926–932.

[7] Björn Ommer, Theodor Mader, and Joachim M. Buhmann. Seeing the objects behind the dots: Recognition in videos fromamoving camera. *International Journal of Computer Vision*, 83(1):57–71, 2009.

[8] C. Rabe, U. Franke, and S. Gehrig. Fast detection of moving objects in complex scenarios. In *2007 IEEE Intelligent Vehicles Symposium*, pages 398–403, June 2007.

[9] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J. Black. Optical flow with semantic segmentation and localized layers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and gpca. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–310–II–316 Vol.2, June 2004.

[11] Li Xu, Jianing Chen, and Jiaya Jia. A segmentation based variational model for accurate optical flow estimation. ECCV '08, pages 671–684, 2008.

[12] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part IV*, ECCV'06, pages 94–106, Berlin, Heidelberg, 2006. Springer-Verlag.

# SfT-from-CNN — Using Deep Learning to Aid Deformable 3D Reconstruction

Armine Vardazaryan* and Adrien Bartoli**

*University of Burgundy
**Institut Pascal, EnCoV - UMR 6602 CNRS (ex-ISIT)

***Abstract*— In the last few years a lot of research has been done on the topic of 3D reconstruction with convolutional neural networks from a single image. The majority of these works are affected by the inherent ambiguity of inferring 3D from 2D, as an infinite number of shapes can produce the same 2D projection. In contrast, in Shape-from-Template the template-based 3D reconstruction has been proven to be unambiguous, as long as the object deforms isometrically.**

**Our work aims to combine the two methods. We restrict our network to a specific object and embed the template of the object into the network. By ensuring the deformation is near-isometric we avoid ambiguity during reconstruction. However, instead of solving the task analytically, our method is based on deep learning. In addition to the task of 3D prediction, we use the embedded template and the visual cues from the input image to estimate the focal length.**

**In broader terms, the task that we are addressing in this work combines three separate problems: detection, registration and reconstruction. These are still considered generally open problems in the field of computer vision. Despite the complexity of this task, our method produces good results both on synthetic and real datasets.**

## I. INTRODUCTION

3D reconstruction is one of the fundamental tasks in computer vision. A lot of research has been done on the subject of capturing the 3D geometry and appearance of objects from images. Subsequently, there exist many approaches of 3D reconstruction. However, the areas of research closest to our method are Structure-from-Motion (SfM) and Shape-from-Template (SfT). The first category of algorithms, SfM-based methods, require multiple images of the scene for reconstruction and deal with rigid objects only. In contrast, SfT-based methods [20], [3], [7] can solve the task of 3D reconstruction from a single image, given the template of the object a priori. With SfT, if the object is deforming isometrically or near-isometrically, the solution to the 3D reconstruction with SfT is unique. The fact that this problem is well-posed and solvable with SfT, gives rise to the idea of solving the task with convolutional neural networks (CNN).

In the recent years, many novel methods of solving 3D reconstruction from a single image with CNN were proposed. In contrast to analytical methods, these approaches typically do not involve geometrical computations. Instead, they rely on the powerful deep neural architectures, which are able to learn hidden patterns in data with highly non-linear decision functions. Although CNN-based reconstruction methods can demonstrate very good results, a common issue with these approaches is the inherent ambiguity of recovering 3D from 2D.

In this work we propose a method of 3D reconstruction that adopts principles from both CNN-based methods and SfT. Similar to existing CNN-based works [11], [8] we use a deep neural architecture to estimate the physical geometry of the object from a single image. However, we avoid the issue of ambiguous 3D ground truth by integrating the template of the object into the network itself. In particular, we address the task of reconstruction of a single and specific object that the network detects in a given image.

The task we are trying to solve with this work is twofold. First, given an image of the object, producing the 3D coordinates of the points belonging to the object's surface. Second, estimating the effective focal length of the camera from the given image.

Several assumptions about the object are made. *(i)* The reconstructed object is a single specific object. It is present (at least partially) in all images in the training dataset. *(ii)* The reconstructed points do not change their position in the template and are always on the surface. *(iii)* In the experiments with a deformable object, the deformation is random but constrained with the deformation model. It is assumed to be near-isometric. *(iv)* The texture of the object is known. With these conditions , we consider two cases for the object model: rigid and deformable.

In the rigid case, the object does not undergo deformation. Although there are infinite possible 3D shapes that would cast the same 2D projection, the network is able to determine the correct shape. This is ensured by the training on a large dataset during which, the network learns the template and the pose implicitly.

In the case of a deformable model, the network is trained to determine the extent of the deformation present in the image. This is attained by training the network on a large quantity of samples with deformations that are random but limited to the deformation model.

We evaluate our method on both synthetic and real datasets, in the case of a rigid model. In the case of a deformable model, however, we test only on synthetic data, as no real 3D dataset with appropriate deformations was available.

Our main contributions are summarized below:

- We are the first to undertake the task of point set generation for physics-based deformable objects from

a single image.

- We avoid the ambiguity of 2D-3D inference by incorporating a template of the object into our network.
- In addition to the task of 3D point set generation, we jointly estimate the effective focal length corresponding to the input image.

## II. RELATED WORK

### A. 3D Reconstruction from single image

3D reconstruction from a single image is a difficult and often ill-posed task that has been addressed with CNNs in various recent works. The target of the reconstruction can be 3D human skeleton landmarks/joints [25], [24], faces [18], [19], [26], rigid objects [8], [11], [30], [23], [28], etc.

The 3D face reconstruction task has been addressed with deep nets in many recent works [18], [19], [26]. In [18], [19] and[34] the network is trained to take a single image of a face and a vector of statistical parameters as input and produce a correction of 3DMM coefficients. Instead of a single pass that would improve the results marginally, they introduce a feedback loop that iteratively corrects the prediction. In contrast, [34] proposes a method that uses no synthetic data and is render-free. This method, intended mainly for 3D face recognition, is trained on real images.

Similar to these approaches our proposed method also aims to reconstruct a deformable object with a shape prior. Unlike our method, these works use statistical representation of object modeling.

Methods reconstructing rigid objects generally use physics-based object modeling. Additionally, during training some of this methods use multiple viewpoint images as input [23], [32], while others, like us, rely on 3D data and corresponding image for training[8], [11], [30]. The cornerstone of [8] is their novel layer called 3D-LSTM, which is an extension of standard LSTM framework incorporating 3D convolution and enforcing the preservation of local structure in the volume. However, more relevant to our work is [11]. [11] introduces a generative network called PointOutNet, which outputs a 3D point set in the camera frame. Although its output format is similar to ours, [11] differs from our method in their approach to dealing with the inherent ambiguity of multi-class 3D inference from 2D: they add an input channel that acts as a seed for introducing statistical randomness. In contrast, our method uses a shape prior (learned by the network) to eliminate the uncertainty. Additionally, for real world images, they segment the object beforehand, whereas our method shows good results even with a cluttered background.

### B. Focal Length Estimation

One of the tasks that we aim to solve with this work is focal length estimation. Many approaches exist to determining the focal length from a single image. Some of these methods use prior information about the image [4], [2], while others rely solely on visual cues and the content of the image[16], [27], [13], [22], [33], [6], [14], [5], [9], [10], [29].

Similar to our work, [4] and [2] propose template-based methods that are able to both estimate the focal length of the input uncalibrated image and reconstruct the 3D geometry of a specific object. In contrast to our work however, these methods propose analytical solutions to the problem, while we use CNNs for determining both the 3D shape and the focal length.

Other methods estimate the focal length by detecting specific patterns or objects in the image, such as a planar calibration grid [33], [13], [22], [27] coplanar circles [6], vanishing points [5], [9], [10], etc.

The closest of these methods to ours is a CNN-based approach described in [29]. Their proposed network, DeepFocal, is able to determine the focal length from images 'in the wild': uncalibrated images found on the World Wide Web. No additional constraints are applied.

Our approach combines the template-based methods[4], [2] with the CNN-based method [29] in a deep convolutional architecture, that is able to predict the focal length from a single image enabled by the template of the reconstructed object.

## III. METHODOLOGY

As a 3D model shape for prediction we use a pillow-shaped textured mesh. Because we use only a single object for training and expect the network to output 3D information about it, the CNN will, essentially, learn the template shape of the model. Thus, the template in this SfT method is not an explicit one, but embedded into the network itself. It is also noteworthy that while we expect the network to be able to predict 3D points, the input image contains no cues as to the locations of the vertices in the mesh. The training process will teach the network to extract the features necessary for such prediction with learned filters that are specialized for the task.

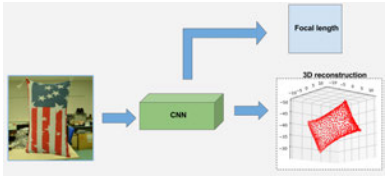



Fig. 1: The pipeline of our method.

In addition to the main 3D shape prediction task, the network is also trained to predict the focal length of the camera corresponding to the input image. This is beneficial for two reasons: first, the intrinsics of the camera can be recovered and used to backproject the vertices onto image plane, and second, the method serve as initialization for other SfT methods that require focal length for reconstruction. We are estimating a single value for the focal length instead of the calibration matrix by making the assumption that the principal point is in the center of the image, no skew is present and that the pixels are square.

We chose two different models for the object: rigid and deformable. In the rigid case, the scene in sample images

changes, but all vertices are kept static relative to each other. With the deformable model a low frequency near-isometric deformation is applied to the object. Here, the network learns the nature of the deformation, which combined with the knowledge about the average shape of the object, enables the network to produce accurate predictions.

### A. Data Generation

The task of training a CNN requires a large amount of data, which in practice means that a synthetic dataset must be created. Reconstruction of objects from real images necessitates that the dataset be as realistic as possible. To represent the variation in natural scenes and cameras, we apply different poses to the model and random focal length, lighting and background to each image (random in a certain range). In addition, the model undergoes a deformation in half of the samples.

We use Blender to render realistic images and add different conditions to the imaged scene. The applied conditions are as follows: *(i)* random rigid transformation of the model, *(ii)* model deformation, *(iii)* random focal length (20-50mm), *(iv)* light sources with random locations and emission values, *(v)* random background to simulate clutter (images taken from SUN[31] dataset).

These conditions are needed to help the CNN successfully deal with real images, despite the large variability of real-life scenes. However, even with the addition of the simulated conditions, the network underperforms with real ground truth images. To bridge the performance gap between the two datasets, some ground truth samples were mixed with the synthetic data for training.

In some of the conducted experiments the lighting and background are constant for the purpose of measuring the effect of environmental factors on the quality of prediction. For the dataset with a deformable object, the model is subjected to a random low-frequency near-isometric deformation.

### B. Architecture

We use VGG16[21] as a basis for our network, but modified to serve our purpose. First, we removed the Softmax operation and used a simple linear layer on the output. Second, we changed the number of neurons in the last fully-connected layer to 3006: our model has 1002 vertices and correspondingly, 3006 coordinates (Fig. 2).

While the main branch of the network is responsible for 3D shape prediction, an additional branch is intended for estimation of the focal length. The focal length prediction network connects directly to the convolutional layers and is trained separately. It takes the extracted feature maps from convolutional layers, and uses one fully-connected layer with ReLU[17] activation to learn the focal length. The ReLU layer is followed by a single linear output unit to aggregate the prediction into a single value.

*Loss Function:* The loss function should be able to accurately quantify the differences between two point clouds. Since the correspondences between the two point sets are known, a simple root-mean-square error (RMSE) is effective in measuring the discrepancy between the two sets.


Fig. 2: The architecture of our network is a modified VGGNet.

$$\text{RMSE} = \sqrt{\frac{\sum \left(P_{i_{\text{pred}}} - P_{i_{\text{true}}}\right)^2}{n}} \tag{1}$$

In the equation above $P_{i_{\text{pred}}}$ and $P_{i_{\text{true}}}$ are the predicted and true $i$-th coordinate, respectively.

For evaluation purposes we express the RMSE error in percents by dividing by the longest side of the object and multiplying by 100.

For the focal length prediction we use the relative error as loss function:

$$\text{E}_F = \frac{|F_{pred} - F_{true}|}{F_{true}} \tag{2}$$

Here, $F_{true}$ and $F_{pred}$ are the ground truth and estimated focal lengths (in pixels), respectively.

## IV. Training

Instead of retraining the whole network we chose to take advantage of the network's original weights and fine-tune the network on our data instead. First, we fixed the neural layers in the secondary branch for focal length prediction and train the main branch with 1 as loss. Next, we freeze the main branch, and train the auxilary part with the second loss function 2.

All images are preprocessed before being fed into the network with a whitening procedure.

For all training we use the Adam optimizer[15] with a fixed learning rate and train until convergence without regularization.

For our implementation of the network we used Tensorflow [1]. The VGG16 model implemented with Tensorflow and its weights were taken from [12].

## V. RESULTS

### A. Experimental Setup

We evaluate our method on both synthetic and real examples. To that end, we created four synthetic datasets for evaluation, with about 30000 examples in each. The four synthetic datasets have the following features: *(i)* rigid model, random pose, focal length, *(ii)* rigid model, random pose, focal length, lighting, background, *(iii)* deformable model, random pose, focal length, *(iv)* deformable model,

random pose, focal length, lighting, background. To clarify, in the experiments by 'random' we mean random in a limited range. The range was selected to mimic realistic images as closely as possible. In addition to the synthetic datasets, a real ground truth dataset was utilized for evaluation. However, this dataset covers only the rigid case, so the cases with a deformable model were tested on synthetic data only.

### B. Experiments

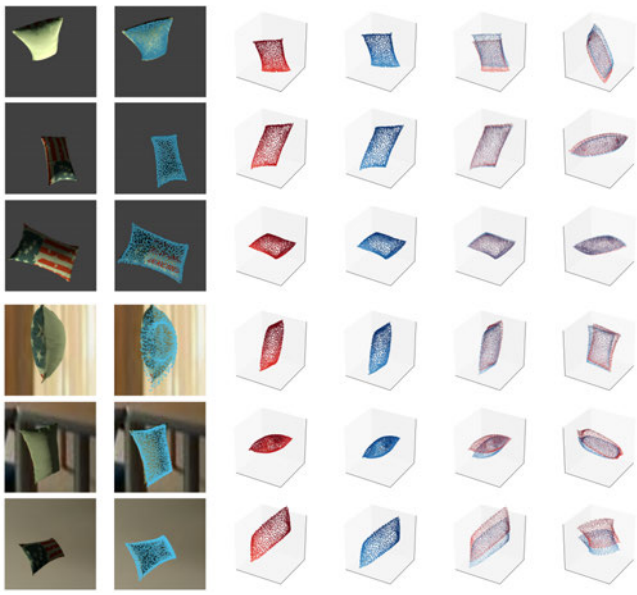
*1) Rigid Model:* For this part of experimentation the model remains rigid while the scene conditions change randomly. As mentioned, two synthetic datasets and a real ground truth dataset are used for evaluating the performance with a rigid model. In the first experiment, to successfully reconstruct the 3D geometry, the network has to solve two additional problems: pose estimation and focal length prediction. Without correct pose estimation the recovered geometry will not have the right 3D orientation. Similarly, a good estimation of the focal length is essential for reconstruction of shape and the depth in the camera frame.

For the second experiment with a rigid model, the dataset contains additional environmental conditions (random lighting and background) while the object remains rigid. This setup is designed to mimic real images, helping the network bridge the gap between synthetic and real data. With the presence of a cluttered background the networks' filters learn to search for this particular object despite the clutter. In Fig. 3 we can see that with the added complexity of the scene the prediction accuracy decreases slightly. Despite that, the results indicate that our network is able to detect the object and learn its pose, even in the presence of unknown lighting conditions.

In addition to synthetic data, we added a small number of real examples to the synthetic training set, with the purpose of improving the results on the ground truth test set. The corresponding test results are shown in figure 4. In the figure we see that the performance on ground truth is very similar to that on synthetic data.

*2) Deformable Model:* The second part of experiments are on a deformable model. The two types of scenes are identical to those of the rigid dataset. However, here the object went through a random deformation defined by the model. Some of the obtained results on this dataset are presented in figure 5. From the figure it can be seen that the network is able to recognize and reconstruct the nature of the deformation that is presented in each particular image. However, the deformation also introduces an ambiguity, especially when only part of the deformation is visible. From the results in Fig. 5 we can see that the performance has dropped slightly compared to the previous setup. This finding is similar to the rigid cases: complex lighting and background make the task of detection and reconstruction more challenging.

*3) Focal Length Prediction:* In the experiments above the network was trained for both 3D shape prediction and effective focal length estimation. To visualize the quality of focal length prediction, we have first projected the true 3D points using the estimated focal length onto the input image (6b), then imaged the predicted points with the predicted focal length, overlaid with the true projection (6c).



(a) (b) (c) (d) (e) (f)

Fig. 3: Results of estimating the rigid model with basic and full synthetic experimental setups. (a, c) - Original image and 3D ground truth. (d) - the obtained 3D reconstruction. (b) - The projection of the reconstruction onto the input image. (e, f) - The reconstruction (blue) and ground truth (red) overlaid, seen from two different angles.

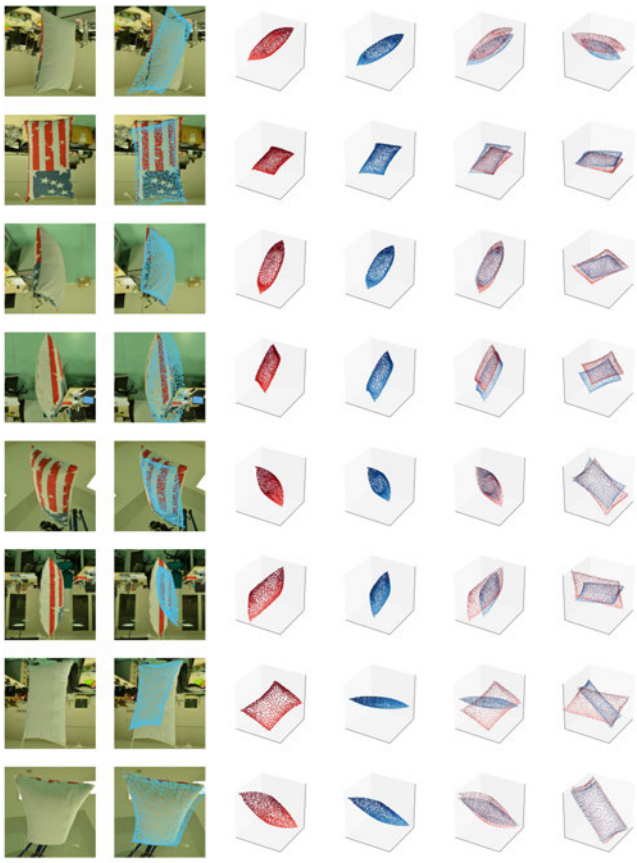
### C. Discussion

TABLE I: Quantitative results of the presented experiments.

| | Rigid | | | Deformable | |
|---|---|---|---|---|---|
| | Basic | Full | GT | Basic | Full |
| 3D error (RMSE %) | 4.85 | 8.8 | 9.37 | 9.06 | 11.5 |
| FL relative error (%) | 21.0 | 23.0 | 37.0 | 30.0 | 16.0 |

The qualitative results are presented in table I, where the RMSE error for 3D and the relative error for focal length are both expressed in percents. The RMSE percents are computed relative to the object length. From the table we see that the performance of 3D prediction drops gradually from simpler to difficult cases. Even the addition of deformation only impacts the performance significantly, raising the error to 9%.

The performance of focal length estimation seems to indicate that the more complex the scene the better the prediction would be. On the other hand, for the rigid case the results for the two types of synthetic datasets are almost identical. A possible reason for that could be the size of the dataset, as the number of samples in each dataset is slightly different. In particular, the ground truth dataset is much smaller than the synthetic ones, and accordingly, the



(a) (b) (c) (d) (e) (f)

Fig. 4: Results of reconstruction on the real ground truth. Red point set is ground truth, blue is reconstruction.

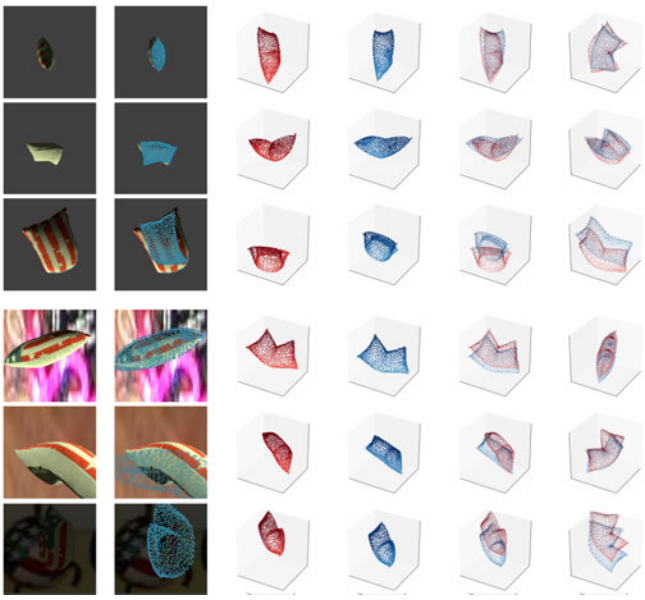
network does not seem to have learned well to extract the necessary information.

*Failed Cases:* From the presented information, it is clear that our method has a certain range of conditions where the prediction is optimal. While this is a compound problem with many factors, we can see from some of the failed cases (Fig. 7) that scenes with occlusion, visually ambiguous poses or poor illumination are likely to result in inadequate reconstruction.

## VI. CONCLUSION

In this article we presented our method that solves the task of 3D reconstruction with CNNs. The most important feature of our method is its ability to predict 3D deformations from a single image. In addition, an auxiliary task we solve here is focal length prediction.

Our method of 3D estimation has several advantages. First, in contrast to most existing methods, we use a physical model of the object as an output format. This representation is superior to statistical representations, as no information is lost in the process of compressing the data and the details are preserved. Second, we avoid the inherent ambiguity of inferring 3D from 2D by embedding a knowledge about the



(a) (b) (c) (d) (e) (f)

Fig. 5: Results of reconstruction the deformable model with basic and full synthetic experimental setups. Here, red is ground truth, blue is reconstruction.

shape and size of the object into the network. And finally, we use the embedded template and the visual cues from the input image to estimate the focal length.

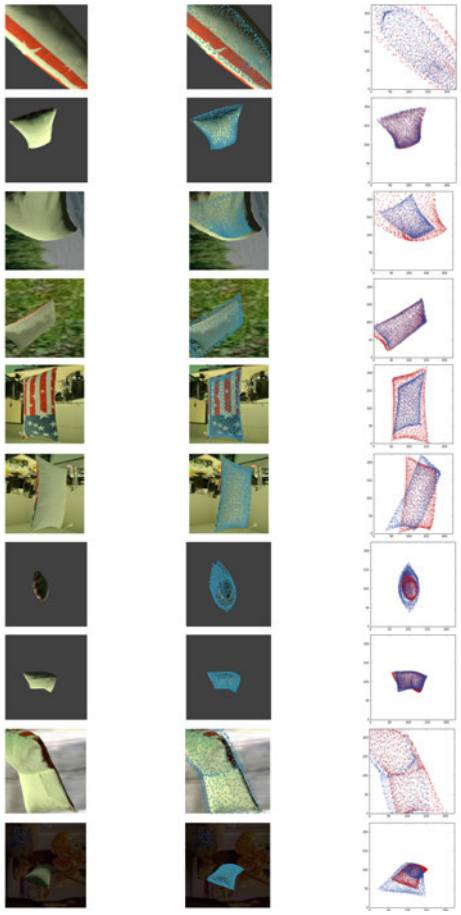
In the future, our work can be extended to incorporate object detection and 3D estimation refinement. Currently, we make the assumption that the object is always visible in the image. If given an image that does not contain the object, the network will hallucinate an average 3D representation. Therefore, an important addition to our network could be a detection layer with a binary output. Next, an important addition would be a prediction refinement network. One possible way of incorporating it can be a small network that improves the prediction iteratively by taking the image features and the 3D estimation and producing an improved prediction.

## VII. ACKNOWLEDGMENTS

This research was possible thanks to the funding received from the EUs FP7 through the ERC research grant 307483 FLEXABLE.

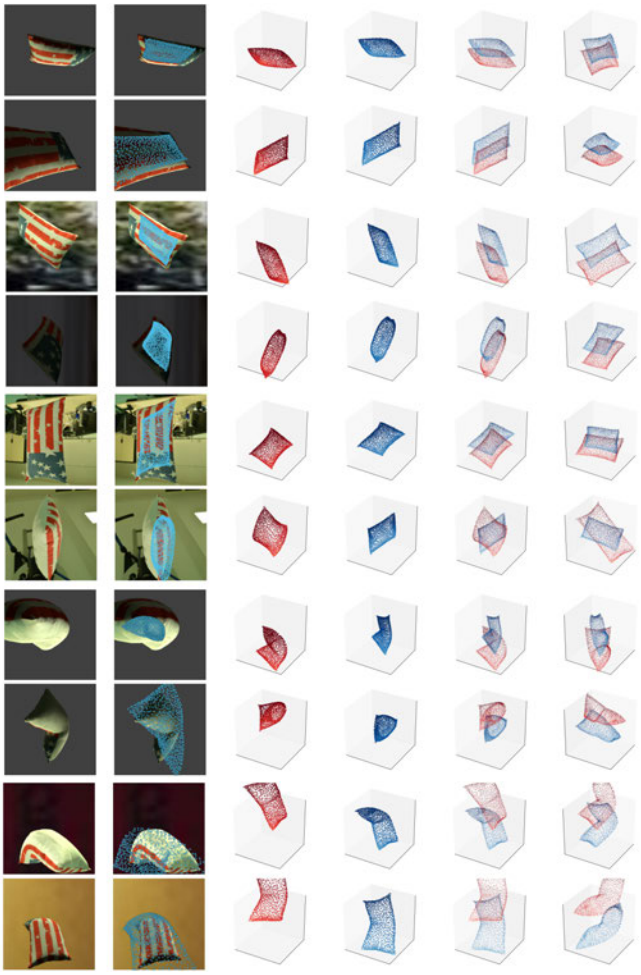
## REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.



(a) (b) (c)

Fig. 6: Results of focal length prediction. (a): original image, (b): ground truth projected on the image with estimated focal length, (c): 3D reconstruction with estimated focal length (blue) and ground truth (red).



(a) (b) (c) (d) (e) (f)

Fig. 7: Failure cases. Here, red is ground truth, blue is reconstruction.


[2] A. Bartoli and T. Collins. Template-based isometric deformable 3d reconstruction with sampling-based focal length self-calibration. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1514–1521, June 2013.

[3] A. Bartoli, Y. Grard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-template. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118, Oct 2015.

[4] A. Bartoli, D. Pizarro, and T. Collins. A robust analytical solution to isometric shape-from-template with focal length calibration. In *2013 IEEE International Conference on Computer Vision*, pages 961–968, Dec 2013.

[5] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[6] Qian Chen, Haiyuan Wu, and Toshikazu Wada. *Camera Calibration with Two Arbitrary Coplanar Circles*, pages 521–532. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[7] A. Chhatkuli, D. Pizarro, A. Bartoli, and T. Collins. A stable analytical framework for isometric shape-from-template by surface integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):833–850, May 2017.

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer International Publishing, 2016.

[9] Roberto Cipolla, Tom Drummond, and Duncan P Robertson. Camera calibration from vanishing points in image of architectural scenes.

[10] Jonathan Deutscher, Michael Isard, and John MacCormick. Automatic camera calibration from a single manhattan image. *Computer VisionECCV 2002*, pages 373–377, 2006.

[11] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *CoRR*, abs/1612.00603, 2016.

[12] Davi Frossard. VGG in TensorFlow model and pre-trained parameters for vgg16 in tensorflow. https://www.cs.toronto.edu/~frossard/post/vgg16/, note = Accessed: 2017-05-27.

[13] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, Jun 1997.

[14] Guang Jiang and Long Quan. Detection of concentric circles for camera calibration. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 333–340 Vol. 1, Oct 2005.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[16] Jean-Francois Lalonde, Srinivasa G Narasimhan, and Alexei A Efros. Camera parameters estimation from hand-labelled sun sositions in image sequences. *Robotics Institute*, page 232, 2008.

[17] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Frnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress, 2010.

[18] E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 460–469, Oct 2016.

[19] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. *CoRR*, abs/1611.05053, 2016.

[20] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1054–1061, June 2009.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] P. F. Sturm and S. J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, page 437 Vol. 1, 1999.

[23] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. *Multi-view 3D Models from Single Images with a Convolutional Network*, pages 322–337. Springer International Publishing, Cham, 2016.

[24] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Fusing 2d uncertainty and 3d cues for monocular body pose estimation. *CoRR*, abs/1611.05708, 2016.

[25] Denis Tomè, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *CoRR*, abs/1701.00295, 2017.

[26] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *CoRR*, abs/1612.04904, 2016.

[27] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, August 1987.

[28] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. Surge: Surface regularized geometry estimation from a single image. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 172–180. Curran Associates, Inc., 2016.

[29] S. Workman, C. Greenwell, M. Zhai, R. Baltenberger, and N. Jacobs. Deepfocal: A method for direct focal length estimation. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1369–1373, Sept 2015.

[30] Jiajun Wu, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. Tenenbaum, Antonio Torralba, and William T. Freeman. *Single Image 3D Interpreter Network*, pages 365–382. Springer International Publishing, Cham, 2016.

[31] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, June 2010.

[32] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. *CoRR*, abs/1612.00814, 2016.

[33] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000.

[34] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 146–155, June 2016.

# Synchronization of Projective Frames

Vishnu Veilu Muthu
**Université de Bourgogne, 720 Avenue de l'Europe, 71200 Le Creusot, France**

***Abstract*— Synchronization of projective frames is a method of integrating sets of projectively reconstructed matrices in such a way that they differ from the true reconstruction by a single global projective transformation. Projective reconstruction is a method of computing the structure of a scene from images or corresponding points taken with uncalibrated cameras. This results in a scene structure and camera motion that may differ from the true geometry by an unknown 3D projective transformation. Both in traditional and standard reconstruction methods, most method suffers from some common drawbacks like multiple solution and sign or scalar variations. To handle such problems, we use a technique called "Global Synchronization". Synchronization is a method of localizing network of cameras or identifying a localizable sub-network and analyzing a better way to connect with the network. The essence of the synchronization problem is most conveniently modeled by a graph, where the points are associated to the nodes, and the direction constraints to the edge. Our method is thus global, where no initial point is needed, and a globally-optimal solution can be found without the prior information about the scene such as identical camera positions or orientations, smooth camera trajectory and 3D points. Unlike traditional projective reconstruction method, our method can handle real-world difficult cases like missing data in a unified manner. The underlying paper demonstrate the performance of the new algorithm on challenging image sequences both in real and synthetic data (by adding error and missing data).**

## I. INTRODUCTION

Synchronization of projective frames is a method of integrating sets of projectively reconstructed matrices in such a way that they differ from the true reconstruction by a single global projective transformation. Projective reconstruction is a method of computing the structure of a scene from images or corresponding points taken with uncalibrated cameras. This results in a scene structure and camera motion that may differ from the true geometry by an unknown 3D projective transformation. This projective transformations are the significant relation that connects between the projective space. The task of reconstruction is to determine the unknown quantities of configuration of the corresponding 3D points and the locations of the cameras that projects the images. This gives the general idea behind the projective reconstruction and the problem of estimating the projective transformation matrix. In theory, many standard methods are available for the projective reconstruction. Both in traditional projective reconstruction methods and standard methods, most method suffers from some common drawbacks like multiple solution and sign or scalar variations. The process of solving those methods requires large iteration process and may not converge or only converge to a local minimum. To handle such problems, we use a optimal solution called "Global Synchronization". In the field of science, with the pieces of information from single block or unit, there is always a need for a better way to communicate with other units. This rule also applies with many new technology like internet, cloud,etc. Such idea is utilized in this work. Synchronization is a method of localizing network of cameras or identifying a localizable sub-network and analyzing a better way to connect with the network. In general, monocular cameras can measure line of sight to the other cameras but are not able to easily determine the distances and the position which leads to localization problem. As such, this problem has appeared under various forms in different settings, such as sensor network localization and formation control in the controls and robotics communities, structure from motion in computer vision, and graph drawings in the discrete mathematics community. The essence of the synchronization problem is most conveniently modeled by a graph, where the points are associated to the nodes, and the direction constraints to the edge.

Our method also can handle real world difficult cases like missing data in a unified manner. This problem is important due to its applications in the analysis of dynamical scenes where cameras are deployed in a static configuration or are mounted on robots. The more advanced practical applications such as super-resolution imaging [7] , multiple eye calibration [ [9], [13]], video compression , and camera auto-calibration can be featured using this technique.

### A. Paper Organization

This paper is organized into different section starting with a brief introduction. Section 2 provides a detailed discussion on projective reconstruction and transformation. Then comes the description of the synchronization in Section 3. The algorithm is discussed in detail in Section 4. Section 5 of the paper highlights the setup, experiments and the results. In Section 6 we address conclusion and future work.

## II. PROJECTIVE RECONSTRUCTION

Consider n stationary 3D points distributed in space as $X_j = [x_j, y_j, z_j, 1]^T \in \mathbb{R}^3$ where $j = 1, ..., n$. Under m projective cameras $P_i$ where $i = 1, ..., m$, the 3D point $X_j$ is projected onto image points $x_{ij} = [u_{ij}, v_{ij}, 1]^T \in \mathbb{R}^2$. The image point $x_{ij}$ represents the image coordinates of the jth 3D point seen in the ith image. It is generally not possible that every points will be location in every image frame, so only a subset of all possible $x_{ij}$ are given. The projective reconstruction is to determine the camera projection matrices

$P_i$ and the 3D point locations $X_j$ such that the projection of the j-th point in the i-th image is the measured $x_{ij}$. Assuming a pinhole (projective) camera model, this relationship is expressed as a linear relationship as:

$$\mathbf{x_{ij}} \simeq \mathbf{P_i}\,\mathbf{X_j} \tag{1}$$

where $P_i$ is a $3\times4$ matrix of rank 3, $X_j$ and $x_{ij}$ are expressed in homogeneous coordinates, and the equality is intended to hold only up to an unknown scale factor $\lambda_{ij}$ . Therefore,the projection equation is give by,

$$\lambda_{\mathbf{ij}}\,\mathbf{x_{ij}} = \mathbf{P_i}\,\mathbf{X_j} \tag{2}$$

The solution to the reconstruction problem may only be determined up to an unknown projective transformation T, applied both to points and cameras. A projective transformation for the model of 3D space containing world points is shown by,

$$\mathbf{X} \mapsto \mathbf{T}\,\mathbf{X} \tag{3}$$

where T is a non-singular $4\times4$ matrix representing a mapping between homogeneous coordinates. From this relationship, it is seen that the determination of camera matrices $P_i$ and points $X_j$ cannot be unique, given only corresponding image coordinates $x_{ij}$ . Consider

$$\begin{aligned}\mathbf{x_{ij}} &= \mathbf{P_i}\,\mathbf{X_j}\\ &= (\mathbf{P_i}\,\mathbf{T^{-1}})(\mathbf{T}\,\mathbf{X_j})\\ &= \mathbf{P'_i}\,\mathbf{X'_j}\end{aligned} \tag{4}$$

In this relationship, new points $X'_j = T\,X_j$ are defined in terms of points $X_j$ , and similarly new camera matrices $P'_i = P_i\,T^{-1}$ in terms of the camera matrices $P_i$. Since both $(P_i, X_j)$ and $(P'_i, X'_j)$ give rise to the same projected image coordinates $x_{ij}$ , there is no way to choose between these two solutions to the reconstruction problem. In fact, there exists a complete family of solutions to the problem, corresponding to all possible choices of the matrix T. All such solutions are related to each other by the application of a projective transformation, and are hence called projectively equivalent. The above analysis does not rule out the possibility that other solutions to this reconstruction problem exist, not related to a particular obtained solution by any projective transformation. In our work we are using three view reconstruction method as an advantage to determine the transformation relation between the two overlapped projective view. If three images of a scene are available, and point correspondences are known across all three views, then such linear methods can be extended to three image reconstruction, using the trifocal tensor. The projective reconstruction of n views can be done by the standard factorization algorithms like Tomasi-Kanade factorization [34] and Sturm-Triggs [27] methods.

### *A. Collineation T*

The projective reconstruction is done by chaining partial reconstructions from 3 views that are obtained using a 6-points procedure described in [19] or Sturm-Triggs iteration described in [27]. After the process of reconstruction, triplets of projection matrices are obtained, which is related to the correct (Euclidean) one by a collineation of the 3D space. Thus the true transformation can be computed by a method called synchronization, which will be discussed in the later chapters. The collineation T between two sets of reconstructed matrices can be obtained only if they have two overlapping projectively equivalent matrices. A reference projective frame is fixed, that is the one associated to the first triple $\{P_a, P_b, P_c\}$, subsequent triples of $P_i$ with an overlap of two like $\{P'_a, P'_b, P'_d\}$ for the triples of $P'_i$ can be brought to the same frame by computing the proper collineation T. So let $P_a$ and $P_b$ be the same camera in two different projective frames, i.e., $P'_a$ and $P'_b$ represents the same camera in two different triplets. They are related by an unknown collineation T:

$$\mathbf{P_i}\,\mathbf{T} \simeq \mathbf{P'_i} \tag{5}$$

In the next step, the elements of the matrices are reshaped to vector column-wise arrangement. This is denoted by introducing the $vec$ operator. Thus the equation can be rewritten as,

$$\mathbf{vec}(\mathbf{P_i}\,\mathbf{T}) \simeq \mathbf{vec}(\mathbf{P'_i}) \tag{6}$$

Let us consider a and b two vectors of $\mathbb{R}^n$ be $vec(P'_i)$ and $vec(P_i\,T)$. Their equality up to a scale can be written as: $rank[a; b] = 1$. This is to say that all minors of [a; b] are zero. There are n(n-1)/2 of such order-two minors, and they can be obtained by multiplication of b by a suitable $n(n-1)/2\times n$ matrix that contains the entries of a. Let us call this matrix $[a]_\times$ in analogy to the $\mathbb{R}^3$ case, where equality up to a scale reduces to $a \times b = 0$. Since by construction, $a$ belongs to the null-space of $[a]_\times$, its rank is at most n - 1. Hence $a \simeq b$ gives rise to the linear system of n(n-1)/2 equations $[a]_\times b = 0$ where only n-1 of them are independent. The matrix $[a]_\times$ is composed by n-1 blocks arranged by rows. The the $i^{th}$ block has (n - i) rows and n columns (i = 1,...,n - 1):

$$\mathbf{B_i} = \begin{bmatrix} 0_{1\times(i-1)} & -a_{i+1} & a_i & 0 & 0 & \dots & 0 \\ 0_{1\times(i-1)} & -a_{i+2} & 0 & a_i & 0 & \dots & 0 \\ 0_{1\times(i-1)} & -a_{i+3} & 0 & 0 & a_i & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & & \dots \\ 0_{1\times(i-1)} & -a_n & 0 & 0 & 0 & \dots & a_i \end{bmatrix} \tag{7}$$

$$[\mathbf{a}]_\times = \begin{bmatrix} B_1 \\ B_2 \\ B_{n-1} \end{bmatrix} \tag{8}$$

Thus equality of two vectors $vec(P'_i)$ and $vec(P_i\,T)$ of $\mathbb{R}^n$ up to a scale can be written as,

$$[\mathbf{vec}(\mathbf{P'_i})]_\times\,\mathbf{vec}(\mathbf{P_i}\,\mathbf{T}) = \mathbf{0} \tag{9}$$

Using the properties of the Kronecker product, the unknown vec(T) can be separated from the previous equation as,

$$[\mathbf{vec}(\mathbf{P'_i})]_\times\,(\mathbf{I_{4\times4}} \otimes \mathbf{P_i})\,\mathbf{vec}(\mathbf{T}) = \mathbf{0} \tag{10}$$

Since the coefficient matrix has rank at most 11, at least two camera matrices are needed to stack-up the 15 equations required to compute the 4 x 4 matrix T up to scale. As said

earlier, this is the reason why the projective reconstruction processes triples of cameras with an overlap of two matrices. Let $P_i$ be the real projective matrix of the camera associated to the i-th image in the set with the infinity projective plane between different views. Observe that in a Euclidean frame $H_i$ would be the homography induced by the true infinity plane. However, since our cameras are uncalibrated, the projective matrices are defined in a projective frame where the infinity plane corresponds to a generic plane in the Euclidean frame. Bringing all the sets of matrices into a common projective frame ensures that the space plane associated to homographies $H_i$ is the same. In this way we obtain an estimate for a fixed reference plane that does not depend on a particular choice of the corresponding points which generate the projective reconstruction. This has clear advantages over other strategies such as tracking 3D points belonging to a plane, or by considering the dominant collineation.

## III. Synchronization

Synchronization is the problem that deals with of global set of cameras (images) or network of sensor framework (orientation or motion). A network is made of cameras (images) in photogrammetry or sensors that a 3D reference frame is attached that represent its position and angular attitude. This properties is referred as "orientation" in photogrammetry or "motion" in Computer Vision, if one consider discrete samples of a moving camera. Mathematically orientation or motion is described by groups of elements with direction. Therefore the goal is to recover location and/or attitude of a bunch of cameras or sensors organized in network. The links of this network (edges of a graph) are relative measures of one cameras or sensors with respect to the others. The "global" adjective means that we are interested in solutions that consider all the measurements at once, as opposed to "incremental" approaches that grow a solution by adding pieces iteratively (such as resection-intersection in the context of structure-from-motion). There are two different type of branch address this problem.

The threading methods treat the problem of finding an initialization based on local homographies as a searching algorithm and generally determine a reference image from the set of images. The coordinate frame of reference image acts as the projective frame. More complex threading methods are based on finding paths in the topology graph. The corner stone of all threading methods is the cost they impose on edges and paths: Kang et al. accumulate the residual error [21], Marzotto et al. impose constraints on the ratio of overlapping area and residual error [26] for Structure from Motion (SfM).

Batch approaches find a global solution by linearly approximating the motion model. For instance, Govindu uses such an approach for SfM in order to build a linear system by uses quaternions [17] or Lie algebra [18]. Trying to solve a similar SfM problem, Sturm [33] uses homographies as input to compute the initial rotation and translation for nonlinear optimization, then factorization method, followed by averaging known rotations.

### A. Related Work

This paper is a extended work of Francesco et al [24] computing parallax maps from monocular and uncalibrated video sequences and Federica et al [4] estimating camera motion in the context of structure-from-motion. Synchronization is a topic that connects with many field like computer vision, sensor networks, automatic controls, robotics, graph theory (parallel rigidity), topography/surveying. They are mainly used as a graph embedding problem, where the sought cameras or sensors locations correspond to an embedding in the 3D euclidean space of the network or graph. Several instances of synchronization have been studied in the literature:

- $\Sigma = \mathbb{Z}_2$ sign synchronization [10];
- $\Sigma = R$ time synchronization [16], [22] (from which the term synchronization originates);
- $\Sigma = R^d$ state / translation synchronization [30], [36];
- $\Sigma = SO(d)$ rotation synchronization (a.k.a. rotation averaging) [6], [8], [14], [20], [23], [25], [32];
- $\Sigma = SE(d)$ rigid-motion synchronization (a.k.a. motion averaging) [3], [5], [18], [29], [35], [37];
- $\Sigma = SL(d)$ homography synchronization [31];
- $\Sigma = \mathbb{S}_d$ permutation synchronization [28].

When considering the attitude part of orientation described by a rotation (element of SO(3)) the problem is also known as rotation averaging Hartley et al. (2013) [20] (or, registration Martinec and Pajdla (2007) [25]). This "averaging" problem can be generalized to any group, giving rise to the Group Synchronization problem ("synchronization" is used in the acceptation of Giridhar and Kumar (2006) [16]), which can be stated as follows: given a graph where edge labels corresponds to noisy measures of the ratio of the unknown labels of adjacent vertices, find the vertex labels. According to the chosen group we have several instances. We are particularly interested in the "rigid-motion synchronization" problem Govindu(2004); Agrawal (2006); Torsello et al. (2011); Tron and Vidal (2014) [ [1], [18], [35], [37] ]which can be seen as a stage of structure-from-motion that starting from the epipolar graph (nodes are cameras or images and edges represent epipolar geometry) find globally consistent orientations or motions for the cameras or images, i.e,. solves the "network orientation" in the acceptation of Fraser (2005) [15] without using points. Thus we optimize and utilize the technique in synchronization of projective frames.

### B. Graph Theory

Let $\Sigma$ be a group with unit element $1_\Sigma$ and $\vec{G} = (V, E)$ be a finite simple digraph, with $n = |V|$ vertices and $m = |E|$ edges. A $\Sigma$-labelled graph is a digraph with a labelling of its edge set by elements of $\Sigma$, that is a tuple $\Gamma = (V, E, z)$ where $z : E \to \Sigma$ is a labelling of the edges such that if $(u, v) \in E$ then $(v, u) \in E$ and $z(v, u) = z(u, v)^{-1}$. Hence, we may often consider $G$, the undirected version of $\vec{G}$.

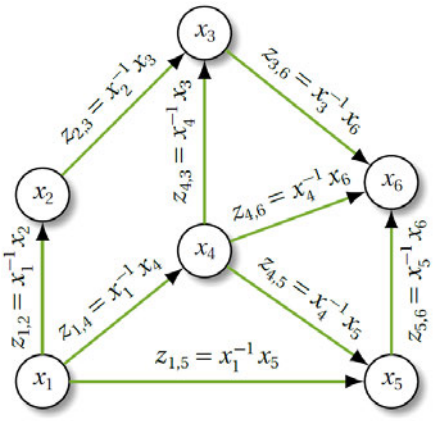



Fig. 1: Graph model

The cycle $v_1v_2, v_2v_3 \dots v_\ell v_1$ in $\Gamma$ is null cycle if and only if,

$$\mathbf{z(v_1, v_2) \cdot z(v_2, v_3) \cdot \ldots \cdot z(v_\ell, v_1) := z(v_1v_2, v_2v_3, \ldots v_\ell v_1) = 1_\Sigma} \tag{11}$$

The model of the graph is shown in the (Fig. 1). Let $\Gamma = (G, z)$ be a $\Sigma$-labelled graph for $G = (V, E)$ and $x : V \to \Sigma$ be a vertex labelling. Then $x$ is a consistent labelling if and only if for each edge $e = (u, v) \in E$ we have

$$\mathbf{x(v) = x(u) \cdot z(e) \iff z(e) = x(u)^{-1} \cdot x(v)} \tag{12}$$

The E edges with outlying labels is solved from the graph theory in ( [12], [11]).

### C. Synchronization in $(\mathbb{C}^{d\times d})$

Synchronization method is extended to $\mathbb{C}^{d\times d}$ in the work. In graph theory, let $\Gamma = (G, z)$ be a $\Sigma$-labelled graph for $G = (V, E)$. Let $\tilde{x} : V \to \Sigma$ be a vertex labeling. The equation is given by,

$$\mathbf{z(u, v) = x(u)^{-1} \cdot x(v) \quad \forall\ (u, v) \in E} \tag{13}$$

The consistency error of $\tilde{x}$ is defined as,

$$\epsilon(\tilde{\mathbf{x}}) = \sum_{\mathbf{(u,v)\in E}} \mathbf{f(\tilde{z}(u, v) \cdot z(u, v)^{-1})} \tag{14}$$

where $\tilde{z}$ is the edge labeling induced by $\tilde{x}$: $\tilde{z}(u, v) = \tilde{x}(u)^{-1} \cdot \tilde{x}(v)$. In Algebraic form, let us assume now that we are given a symmetric function $f : \Sigma \to \mathbb{C}^{d\times d}$ with a unique minimum at $1_\Sigma$ and $f(1_\Sigma) = 0$. Suppose that $\Sigma$ is a group which admits a matrix representation through $d \times d$ matrices (i.e., $\Sigma$ can be embedded in $\mathbb{C}^{d\times d}$), where the group operation reduces to matrix multiplication and $1_\Sigma = I_d$. Let $U$ be the vector containing the vertex labels as block matrices $X$ and $Z$ be the matrix containing the edge labels as block matrices $T$. This is an instance of the (multiplicative) synchronization problem. All the vertex/edge labels can be collected in two matrices $U \in \mathbb{C}^{dn\times d}$ and $Z \in \mathbb{C}^{dn\times dn}$ respectively, which are matrices composed of $d \times d$ blocks:

$$\mathbf{U} = \begin{bmatrix} X_1^{-1} \\ X_2^{-1} \\ \dots \\ X_n^{-1} \end{bmatrix}, \quad \mathbf{U^{-b}} = \left[X_1, X_2, \dots X_n,\right], \qquad \mathbf{Z} = \begin{bmatrix} I & T_{1,2} & \dots & T_{1,n} \\ T_{2,1} & I & \dots & T_{2,n} \\ \dots & & & \dots \\ T_{n,1} & T_{n,2} & \dots & I \end{bmatrix} \tag{15}$$

For a complete graph, the consistency constraint rewrites

$$\mathbf{Z = UU^{-b}} \tag{16}$$

If the graph $G$ is not complete, $Z$ is not fully specified, or equivalently, it has zero entries in correspondence of missing edges, whereas $UU^{-b}$ is fully specified, hence we shall write the constraint as

$$\mathbf{(Z - UU^{-b}) \circ A = 0} \tag{17}$$

where $A$ is the (0-1) adjacency matrix of $G$ (with zero diagonal) and $\circ$ is the Hadamard product. The previous equation can be rewritten as

$$\mathbf{Z_A = (UU^{-b}) \circ A}. \tag{18}$$

were $Z_A = Z \circ A$ represent the matrix of the available measures with zero entries in correspondence of missing edges (and along the diagonal).

$$\epsilon(\mathbf{x}) = \sum_{\mathbf{(u,v)\in E}} \|\mathbf{z(u, v) - x(u)^{-1} \cdot x(v)}\|^{\mathbf{2}} \tag{19}$$

In matrix form, the cost function of the synchronization rewrites

$$\epsilon(\mathbf{U}) = \|\mathbf{(Z - UU^{-b}) \circ A}\|_{\mathbf{F}}^{\mathbf{2}} \tag{20}$$

The Hadamard product with $A$ mirrors the summation over the edges of $E$ in the definition of the consistency error.

### D. Noiseless case

Let us consider the "noiseless" case, i.e. $\epsilon = 0$, and let us start assuming that the graph is complete, Since $U \in \mathbb{C}^{dn\times d}$ then we have $U^{-b}U = nI$. The rank of $U$ is $d$ as,

$$\mathbf{Z = UU^{-b} \iff \underset{\wedge\ \mathrm{rank(Z)=d}}{ZU = nU}} \tag{21}$$

the solution $U$ is the eigenvectors of $Z$ associated to eigenvalues $n$. Since $Z$ have rank four (because $Z = UU^{-b}$), all the other eigenvalues are zero, so $n$ is also the largest eigenvalues of $Z$.

### E. Missing edges case

In the case with missing edges, the graph is not complete and the adjacency matrix $A$ plays an important role in solving the system,

$$\mathbf{Z_A = (UU^{-b}) \circ (A \otimes \mathbb{1}_{d\times d}) \iff Z_A U = (D \otimes I)U} \tag{22}$$

where $D = \text{diag}(A\mathbf{1})$ is the degree matrix of the graph ($A\mathbf{1}$ is the sum of the rows of $A$).

$$\mathbf{Z_A} = \begin{bmatrix} I/\zeta_1 & T_{1,2} & \dots & T_{1,n} \\ T_{2,1} & I/\zeta_2 & \dots & T_{2,n} \\ \dots & & & \dots \\ T_{n,1} & T_{n,2} & \dots & I/\zeta_n \end{bmatrix} \tag{23}$$

where,

$$\mathbf{A_{i,j}} = \begin{cases} 1, & \text{if } \mathrm{T_{i,j}} \text{ is known.} \\ 0, & \text{otherwise.} \end{cases}, \quad \zeta_\mathbf{k} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{A_{i,k}} \tag{24}$$

The adjacency matrix $A$ gets "inflated" by the Kronecker product with $\mathbb{1}_{d\times d}$ to match the block structure of the measures. we can rewrite the equation as,

$$\mathbf{Z_A} = (\mathbf{U}\mathbf{U}^{-\mathbf{b}}) \circ (\mathbf{A} \otimes \mathbb{1}_{\mathbf{d}\times\mathbf{d}}) = \text{diag}(\mathbf{U})(\mathbf{A} \otimes \mathbf{I_d})\,\text{diag}(\mathbf{U^{-1}}) \tag{25}$$

which implies that

$$\begin{aligned} \mathbf{Z_A U} &= \left((\mathbf{U}\mathbf{U}^{-\mathbf{b}}) \circ (\mathbf{A} \otimes \mathbb{1}_{\mathbf{d}\times\mathbf{d}})\right)\mathbf{U} \\ &= \text{diag}(\mathbf{U})\mathbf{A}\,\text{diag}(\mathbf{U^{-1}})\mathbf{U} \\ &= \text{diag}(\mathbf{A1})\mathbf{U} = \mathbf{D}\mathbf{U} \end{aligned} \tag{26}$$

If $\epsilon = 0$, the solution $U$ is the eigenvectors of $(D \otimes I_d)^{-1}Z_A$ associated to the largest d eigenvalues of $(D \otimes I_d)^{-1}Z_A$ which are approximately equal to multiplicity 1's. $(D \otimes I_d)^{-1}Z_A = (D \otimes I_d)^{-1}(Z \circ A) = Z \circ ((D \otimes I_d)^{-1}A)$, Hence $(D \otimes I_d)^{-1}Z_A$ and $(D \otimes I_d)^{-1}A$ are similar, i.e., they have the same eigenvalues. But $(D \otimes I_d)^{-1}A = P$ is the transition matrix of the graph, which has four approximate 1's as the largest eigenvalues, if the graph is connected.

### *F. Noisy case*

When noise is present, the minimum consistency error is not 0, hence the null-space and the eigen solutions do not coincide. If $Mx = \lambda x \iff x \in (M - \lambda I)$, the eigen solutions are the eigenvectors corresponding to $\lambda_1$,$\lambda_2$,$\lambda_3$ and $\lambda_4$, the largest eigenvalues of $M$, while the null-space solution is the (n-d) kernel of $M - dI$; the two are equivalent only if $\lambda' s = 1$. Moreover, while the eigen solution computes the eigen decomposition of $(D \otimes I_d)^{-1}Z_A$ which is related to the null-space of (assuming $\lambda' s \approx 1$) $(D \otimes I_d)^{-1}Z_A - I = (D \otimes I_d)^{-1}(Z_A - D)$, the null-space solution considers the matrix $Z_A - D$. This matrix has the same kernel of $(D \otimes I_d)^{-1}(Z_A - D)$ only when $Z_A - D$ is rank deficient, which is not true in presence of noise. The kernel of $Z_A - D$ is approximately equal to the kernel of $(D \otimes I_d)^{-1}(Z_A - D)$.

## IV. Algorithm structure

### *A. Methodology*

Sets of projective reconstruction matrices are computed from the correspondence points. Each partial reconstruction is determined up to an unknown projective transformation. The main goal of the work is to compute such unknown projective transformations to bring all the partial reconstruction into the same reference system. Let $P_s = 3 \times 4$ be the projective matrix of camera $s$ in a world reference system where we have $s = 1, 2, .., n$. Then, the projective matrix of camera $s$ in partial reconstruction $i$ is given by $P_s^i = 3 \times 4$. The relation between $P_s^i$ and $P_s$ is given by the equation,

$$\mathbf{P_s^i} = \mathbf{P_s}\mathbf{T^i} \tag{27}$$

where, $T^i = 4 \times 4$ is the unknown projective transformation that maps the world reference system into partial reconstruction $i$. From the different combinations of partial reconstruction we can obtain the relation between transformation T as,

$$\begin{aligned} \mathbf{P_s^i} &= \mathbf{P_s}\mathbf{T^i} \\ \mathbf{P_s^j} &= \mathbf{P_s}\mathbf{T^j} \end{aligned} \Rightarrow \mathbf{P_s^j} = \mathbf{P_s^i}\mathbf{T_i^{-1}}\mathbf{T^i} \Rightarrow \mathbf{T_{ij}} = \mathbf{T_i^{-1}}\mathbf{T^i} \tag{28}$$

here the projective transformation that maps reconstruction $i$ into reconstruction $j$ is given by $T_{ij}$. $T_{ij}$ can be computed from the equality of 2 vectors up to scale, assuming that reconstructions $i$ and $j$ share 2 cameras, this is discussed in detail in the section projective reconstruction. The matrix of projective transformation is inverted when the direction is reversed.

$$\mathbf{T_{ij}} = \mathbf{T_{ji}^{-1}} \tag{29}$$

As discussed in the section synchronization the graph $G = (V, E)$ is constructed. Each vertex corresponds to a partial reconstruction of 3 views and it is labeled with an unknown transformation $T_i$. Each edges corresponds to an overlap of 2 cameras between 2 partial reconstructions, and it is labeled with a unknown transformation $T_{ij}$. The goal is to estimate vertex labeling, given on edge labeling to estimate $T_i$ given a redundant set of $T_{ij}$ such that,

$$\mathbf{T_{ij}} = \mathbf{T_i^{-1}}\mathbf{T_j} \tag{30}$$

It is solved by eigenvalue decomposition method and this method is known as synchronization. Once $T_i$ is known, several projective matrix of the same camera can be computed by,

$$\mathbf{P_s} = \mathbf{P_s^i}\mathbf{T_i^{-1}} \,\mathbf{and}\, \mathbf{P_s} = \mathbf{P_s^j}\mathbf{T_j^{-1}} \tag{31}$$

From many $P_s$ matrices, a final single $P_s$ matrix is obtained by a simple matrix arithmetic mean and this method is known as single averaging.

### *B. Flow chart*

The algorithm is designed with different stages to fit the system. The input is given by the set of images and output is obtained by the real projective matrices of the images. The flow chart is shown in the (Fig. 2)

- In the stage one (Pre-processing), for each images 100 feature points are extracted with strong corner using Harris corners method. All the sets of feature points are combined into group of all combination of triplets which is easy for doing the projective reconstruction. Point matching and RANSAC/SIFT algorithm are used to compute the correspondence points of the triplets.
- In the stage two (Reconstruction), the projective reconstruction is done from 3 views that are obtained using

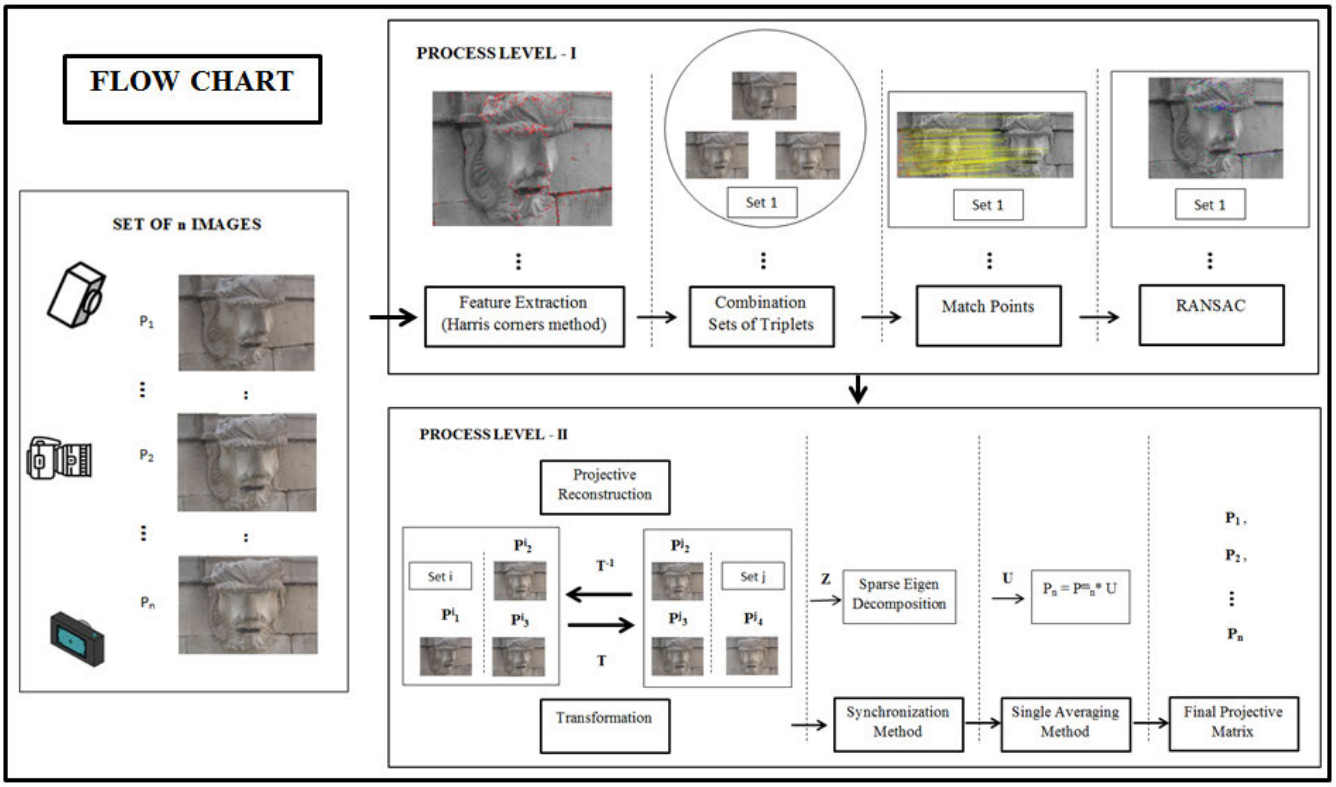



Fig. 2: Flow chart diagram

a 6-points procedure described in [19] or Sturm-Triggs iteration described in [27].

- In the stage three (Transformation), the transformation of the projective matrices is computed with 2 overlapping matrices. The transformation matrix is normalized my dividing with the 4th root of the determent to make the determent 1. Since the projective transformation have both complex and negative eigen values, the normalized transformation matrix is complex. Then these matrix are arranges in the global matrix as $Z$.
- In the stage four (Synchronization), the sparse eigen decomposition is computed with the highest 4 eigen values are taken as $U$. To remove the complex values, $U$ is divided by the first block $X1$ with $4\times4$ matrix. By this way, the vector in the complex plane is arranged in the same direction with different scaling. Now the scale is removed by using least square method and the complex terms are removed.
- In the stage five (Single averaging), the final computed transformation is multiplied with each sets of reconstructed matrices, an arithmetic mean is made to get the real projective matrix.

### C. Applications

*1) Dynamic camera:* Using Structure-From-Motion method, with images or videos as input data advanced sensor fusion techniques can be done with enhanced knowledge of orientation and tie-points computing epipolar or trifocal geometry. More robust methods results in relative rotations and translations computation by solving a motion synchronization problem can be studied by braking into rotation synchronization or translation synchronization or solved both in one step. The global methods can be seen as an effective and efficient way of computing path planning in robotics and 3D reconstruction using the projective reconstructed matrices.

*2) Static camera:* An approach based in static camera is analyzed here. Many real time application based on surveillance camera for security purposes. Multiple eyed devices which is similar to the eyes of house flies, has the best application to the method. Devices like 360 degree cameras,etc can be easily synchronized. Virtual reality techniques can be done with the application like video games and live streaming system.

## V. Experiments & Results

An implementation of the proposed methods in Matlab served for the experiments. There are two types of setup designed for testing a static camera and dynamic camera environment. First we generated about 100 normally distributed 3D points around the camera center. For the viewer those points are uniformly distributed around the static camera center. For dynamic camera setup, we take a random radius of a sphere and the camera position is made to slide on the surface with different internal parameters. A small change in the distance that the views create images for which the geometric transform of a point from one image to the other will produce points at infinity. Then for the static camera

setup, we randomly picked n rotations R1,...,Rn to simulate different views for the camera. Simulating a camera mounted on a tripod we used all the 3 degree of freedom for rotation and translation. From those projected views and with the help of 3D points, we generate the image points for each views. There is a percentage of missing points randomly selected in the image points. Later, this setup is experimented by adding random normal noise for understanding the algorithm better.

### A. Synthetic experiments

We ran experiments on 10 different levels of random noise [0:0.05:5] in steps of noise = 0.05 pixels. Each experiment consisted of 100 random runs. We used a setup composed of n = 20 images. By using more iteration process it is easy to determine the rate of change in error using arithmetic mean or median. The graph shows (Fig. 3) the progressive error after 100 iteration of random samples and comparing with Element-wise Factorization method [38].

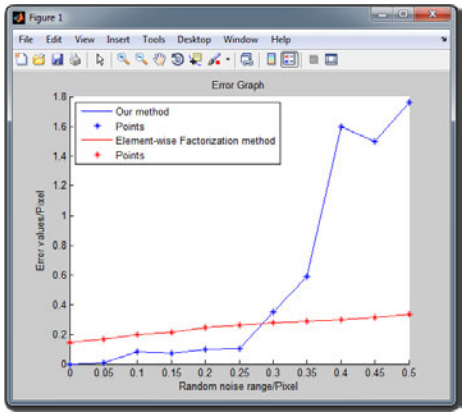



Fig. 3: Graph after 100 iterations

The distribution of 2D points due to the growth in the error is shown in the Fig. 4. A single 3D point is considered and all the similar projective matrix are used to generate the 2D points. From the image we can clearly see the change of the distance grows as the noise increased.

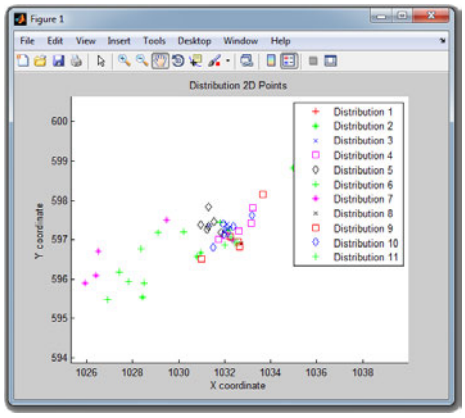



Fig. 4: Noise distribution graph

### B. Real experiments

For the real data experiments a image sequence of n = 20 undistorted frames with a resolution of 640x480 pixels has been used. The images have been aligned using features extraction (Fig. 5) which had been robustly matched (Fig. 6) using RANSAC as described in previous section.

Fig. 5: Feature extraction

Even though it is impossible to determine the ground truth in real examples as for the one denoted here, an overlay of the plots of the frame borders found from the methods reveals, that the impact of the methods can visually be bigger as one might expect.

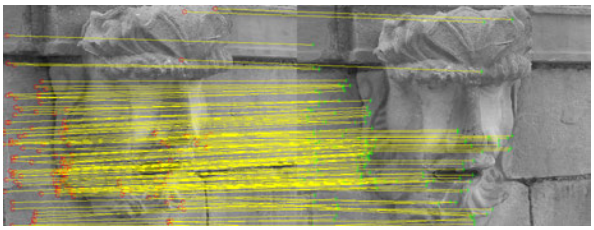


Fig. 6: Feature Matching

In regard to the observation made during the synthetic experiment concerning the mean of the reprojected error, one can suppose that the results varies that reflect the ground truth.

### C. Error Method

Calculating the error in the important work to determine the difference between the ground truth and the obtained value. It is also used as a cost functions which may be minimized in order to determine the over-determined solutions. These methods are used for computing the error.

*1) Algebraic error:* The algebraic error is the simplest method to determine the error in the probabilistic scale. The method uses a simple Frobenius norm between the ground truth and the value obtained. The error e is called the residual of a vector or matrix and it is this norm of error that is minimized. The components arise from the individual correspondences that generate each element of the matrix by normalizing both the values. Though the error does not give further information, it is simple to determine the range of the algorithmic problem.

*2) Transformation error:* Transformation error is used to determine the error between the sets of reprojected matrices before and after the synchronization process with the ground truth set. This method also uses norm between T and T' (before and after the synchronization) which is calculated as described in the projective reconstruction section. The result gives the cost function of the error, which can be minimized through iteration process to get better reprojected matrices.

*3) Reprojection error:* An alternative method of quantifying error in each of the two images involves estimating a correction for each correspondence. The amount necessary to correct the measurements in each of the two images in order to obtain a perfectly matched set of image points. One should compare this with the algebraic error which measures the correction that it is necessary to make to the measurements in one image in order to get a set of perfectly matching points. The process is computing the transformation of the 3D points of the ground truth with the sets of Transformation matrix $U$ after synchronization to new 3D points. These points are projected to 2D points with the final sets of the output projective matrices. Then the mean between the real and the final correspondence points will give the error in the pixel range.

## VI. CONCLUSIONS AND FUTURE WORKS

### A. Conclusions

This research paper is about the extended work for the synchronization problem, which has been faced in many application. The method serves as a common base that can be used in any type of problem with global interface. The work also gives a description about computing the projective transformation between projective matrices and the working of the algorithm both theoretical and practical methods. The ways of computing the error and applying it as a cost function is additional boost in the algorithm. Both theoretical and practical ideas discussed shows the robustness of the application with the algorithm.

The good feature of the algorithm is that the method is easier to implement on different environment as the proposed analytical methods intended to find a solution without an initialization and multiple iterative process. We tested this method with its performance to the maximum limits of behavior under the influence of noise and missing data.

The limitation is based on the connectivity of the graph and if the graph is not completely connected then there is a drift in the synchronization process. The limitation in the single averaging method will be based on the transformation error which lies too for from the projective matrix with the same set of similar matrices. Even with these limitations, from the experimental results showed that this approach is working good and output with very satisfactory results.

### B. Future Works

The future work can be extended based on the theoretical and practical approach. The next level will be based on the working of information and ideas that are required to solve the problem, they are listed as follows.

- Based on the focus on the reconcilement of fundamental or essential matrices between pair of views and the working on the viewing graph.
- Explore the analogy between computing epipolar scales and the reconcilement of essential matrices, and try to extend it to fundamental matrices.
- In the case of essential matrices the scales are needed to bring the reconcilement into a synchronization problem, so the varying scale can be determined.
- The method based on tracking the 3D points from the final projective matrix, this way the relation between the image points and the projective matrices can be analyzed.
- An alternative idea of using the iterative methods, in the synchronization and the single averaging method to adjust the values of the matrices for better projection plane.
- This algorithm can be improved and implemented in the real time applications for the devices that are embedded in automobile systems for optimized tracking and navigation.


## ACKNOWLEDGMENT

I want to thank my supervisors, Prof. Andrea Fusiello and Dr. Federica Arrigoni for giving me an opportunity to work under their guidance and providing continual support throughout my work.


## REFERENCES


[1] M. Agrawal. A lie-algebraic approach for consistent pose registration for general euclidean motion. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1891–1897, 2006. [Cited on page 3.]

[2] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on. IEEE, 2012.*, 2012. [Not cited.]

[3] F. Arrigoni, A. Fusiello, and B. Rossi. Spectral synchronization of multiple views in se(3). *SIAM Journal on Imaging Sciences*, 2016. [Cited on page 3.]

[4] Federica Arrigoni, Andrea Fusiello, and Beatrice Rossi. Camera motion from group synchronization. *3D Vision (3DV), 2016 Fourth International Conference*, 2016. [Cited on page 3.]

[5] F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, and J. Goncalves. A solution for multi-alignment by transformation synchronisation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, 2015. [Cited on page 3.]

[6] Sharp G. C., Lee S. W., and Wehe D. K. Multiview registration of 3d scenes by minimizing error between coordinate frames. In *eccv*, pages 587–597, 2002. [Cited on page 3.]

[7] D. Capel and A. Zisserman. Automatic mosaicing with super-resolution zoom. *Computer Vision and Pattern Recognition (CVPR), 1998 IEEE Conference on. IEEE*, 1998. [Cited on page 1.]

[8] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *iccv*, 2013. [Cited on page 3.]

[9] I.-Hsien Chen and Sheng-Jyh Wang. An efficient approach for dynamic calibration of multiple cameras. *IEEE Transactions on Automation Science and Engineering*, 6, 2009. [Cited on page 1.]

[10] M. Cucuringu. Synchronization over z2 and community detection in signed multiplex networks with constraints. *Journal of Complex Networks*, page 469506, 2015. [Cited on page 3.]

[11] Marek Cygan, Fedor V. Fomin, ukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Micha Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer International Publishing, ISBN:9783319212746, 2015. [Cited on page 4.]

[12] Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlstrm. Clique cover and graph separation: New incompressibility results. *ACM Transactions on Computation Theory (TOCT)*, 2014. [Cited on page 4.]

[13] J. Dias, A. de Almeida, H. Araujo, and J. Batista. Improving camera calibration by using multiple frames in hand-eye robotic systems. *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop*, 1991. [Cited on page 1.]

[14] Arrigoni F., Magri L., Rossi B., Fragneto P., and Fusiello A. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 491–498, 2014. [Cited on page 3.]

[15] Clive Fraser. Network orientation models for image-based 3d measurement. *ISPRS Archives*, XXXVI-5/W17, 2005. [Cited on page 3.]

[16] A. Giridhar and P.R. Kumar. Distributed clock synchronization over wireless networks: Algorithms and analysis. *Proceedings of the IEEE Conference on Decision and Control*, pages 4915–4920, 2006. [Cited on page 3.]

[17] Govindu. Combining two-view constraints for motion estimation. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE*, 2001. [Cited on page 3.]

[18] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 1. IEEE*, pages 684–691, 2004. [Cited on page 3.]

[19] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. [Cited on pages 2 and 6.]

[20] R.I. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision*, 2013. [Cited on page 3.]

[21] E.-Y. Kang, I. Cohen, and G. Medioni. A graph-based global registration for 2d mosaics. *ICPR*, 2000. [Cited on page 3.]

[22] Richard Karp, Jeremy Elson, Deborah Estrin, and Scott Shenker. Optimal and global time synchronization in sensor nets. Technical report, Center for Embedded Networked Sensing: University of California, Los Angeles, 2003. [Cited on page 3.]

[23] Wang L. and Singer A. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: a Journal of the IMA*, 2(2):145–193, 2013. [Cited on page 3.]

[24] Francesco Malapelle, Andrea Fusiello, Beatrice Rossi, Emiliano Piccinelli, and Pasqualina Fragneto. Uncalibrated dynamic stereo using parallax. *Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium*, 2013. [Cited on page 3.]

[25] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEEr*, 2007. [Cited on page 3.]

[26] R. Marzotto, A. Fusiello, and V. Murino. High resolution video mosaicing with global alignment. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004*, 2004. [Cited on page 3.]

[27] Sturm P and Triggs B. factorization based algorithm for multi-image projective structure and motion. *ECCV, Springer-Verlag*, page 709720, 1996. [Cited on pages 2 and 6.]

[28] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multiway matching problem by permutation synchronization. *Advances in Neural Information Processing Systems 26*, pages 1860–1868, 2013. [Cited on page 3.]

[29] D. M. Rosen, C. DuHadway, and J. J. Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *icra*, pages 5822 – 5829, 2015. [Cited on page 3.]

[30] W.J. Russel, D.J. Klein, and J.P. Hespanha. Optimal estimation on the graph cycle space. *IEEE Transactions on Signal Processing*, 59(6):2834 – 2846, 2011. [Cited on page 3.]

[31] P. Schroeder, A. Bartoli, P. Georgel, and N. Navab. Closed-form solutions to multiple-view homography estimation. *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 650–657, Jan 2011. [Cited on page 3.]

[32] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30(1):20 – 36, 2011. [Cited on page 3.]

[33] P. Sturm. Algorithms for plane-based pose estimation. *Computer Vision and Pattern Recognition, 2000. CVPR 2000*, 2000. [Cited on page 3.]

[34] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, page 137154, 1992. [Cited on page 2.]

[35] A. Torsello, E. Rodola, and A. Albarelli. Multiview registration via graph diffusion of dual quaternions. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE*, pages 2441 – 2448, 2011. [Cited on page 3.]

[36] R. Tron, L. Carlone, F. Dellaert, and K. Daniilidis. Rigid components identification and rigidity enforcement in bearing-only localization using the graph cycle basis. In *IEEE American Control Conference*, 2015. [Cited on page 3.]

[37] R. Tron and R. Vidal. Distributed 3-D localization of camera sensor networks from 2-D image measurements. *IEEE Transactions on Automatic Control*, 59(12):3325–3340, 2014. [Cited on page 3.]

[38] Yuchao, DaiHongdong, and LiMingyi He. Element-wise factorization for n-view projective reconstruction. *Computer Vision ECCV 2010. ECCV 2010. Lecture Notes in Computer Science, vol 6314. Springer, Berlin, Heidelberg*, 2010. [Cited on page 7.]

# Skin Lesion Classication using Deep Learning Neural Networks

CHRYSANTHI KATRINI

***Abstract*— Melanoma is a type of skin cancer and it is characterized from the experts as the most aggressive. An early diagnosis and a surgery removal can give to the patient almost $99\%$ survival rate. Several Computer-Aided Diagnosis (CAD) systems have been proposed to assist dermatologists in an early diagnosis. This thesis, it is dealing with the processing of color images that depict images of patients with possible melanoma. The main point is to build a system to identify cases that could be potentially dangerous. The system performs feature extraction using the SIFT and SURF algorithm and these features fed into several classifiers such as Support Vector Machines (SVM), K-Nearest Neighbor (K-NN) and Convolutional Neural Network (CNN) and achieve $94,51\%$ accuracy.**

**Keywords: Skin cancer, SIFT, SURF, Medical image segmentation, Deep Learning, Convolutional neural networks**

## I. INTRODUCTION

Melanoma is a very dangerous dermatological disease and unfortunately more and more people suffering from this. As in every disease, an early diagnosis could save a life. A dermatologist must be sure and quick for the diagnosis. But it is not every time so easy to be right in brief time. There are many countries around the world where hospitals or even doctors are rare[1].
How a doctor will do a diagnosis if he can not be near the patient? Nowadays there are several ways for people to communicate from distance, and all this technology can be used for medical reason. This work proposes an automatic classification of moles to assist dermatologists in their diagnosis.

### A. Methodology

The important steps to diagnosis of melanoma are [2] :-

1) Detect correctly the area with the lesion.
2) Segment this area from the outer area.
3) Extract features from the lesion.
4) Classify the images based on the extracted features.

The Vienna dermoscopy dataset was used for this research which contains 5380 skin lesions images with their ground truth.

### B. Image Segmentation

Image segmentation refers to partitioning of an image into region. Otsu automated thresholding will be used as this method is better for thresholding lesions from the background skin[3].

This work is supervised by Nikolaos Papadakis,Professor in Technological Educational Institute of Crete, npapadak@cs.teicrete.gr

Otsu method measures the spreading of the pixel levels each side of the threshold and considers that the given image is separated in two different classes of pixels, foreground and background, and the threshold is responsible to separate these classes. In the segmented image, the bigger area is assumed that is the main region of interest [4].
Edge detection is a technique for finding objects boundaries within images. It is a technique which is used for image segmentation purposes and it based on the detection of a distinct break in continuity in brightness. By applying an edge detector to an image, it will reduce the amount of data as its result will be the boundaries of objects. The points of an image where brightness is changing, called edges. At this point, it is important to apply an edge detector algorithm for finding the borders of the lesion.The examinated edge detectors were Sobel, Prewitt and Canny.

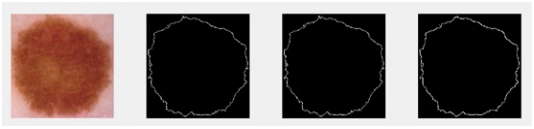


Fig. 1. Edge detectors

Original Sobel Prewitt Canny

### C. Features Extraction

A very important part in object recognition projects is the features extraction from the image. During this procedure, the segmented image is being processed using a user/programmer-selected method of features extraction, with only purpose to gathering a set of characteristics that efficiently describe the most important information. Many approaches have been proposed, analyzed and tested for features extraction. These approaches can be divided into four main categories according the edge, shape, texture and color[5]. In this thesis only two of these approaches will be tested, the Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF).

*1) SIFT:* This approach is used for detection and extraction of local descriptors of images. SIFT is used for object recognition, robotics, 3D image reconstruction, gesture recognition and detect moving object in video[6]. For every object in the image can be extracted interesting points of it, so these points can provide feature description. Usually, these points are placed to the edges of the object where contracts are bigger.
The size of the vector descriptor is 128-dimensional feature

vector. Another important thing during features extraction is the diagnosis matrix. This matrix has to have the same size as the vector descriptor, so possible matches between these two are realizable.A parameter that directly affects SIFT methodis a threshold that limits the amount of exported entities. It is better to have a relative uniformity in the samples. This is the reason why after experiments, threshold is finalized in the value where the amount of descriptors is from 90 to 130 per image.

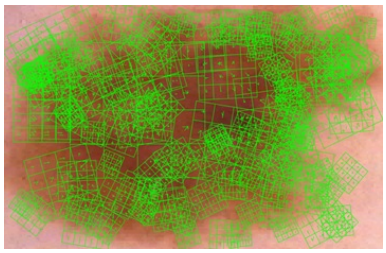


Fig. 2. SIFT descriptors in skin lesion image

*2) SURF:* This method is a fast algorithm for local descriptor-based approach. Points of interest of an image are defined as features from a scale-invariant representation. It is partly inspired by SIFT descriptor, but the basic version of SURF is faster and is considered by the creators to be more resilient to the various image transformations than the SIFT method. SURF is based on Haar 2D wavelet and makes effective use of embedded images[7].
The extracted points via this method called SURF Points and their goal is to provide a unique and robust description of an image. They describe the intensity distribution of the pixels within the neighborhood of the point of interest. Also a matrix with diagnosis is used. For uniformity of this system, only 50 of the strongest points selected for each image.

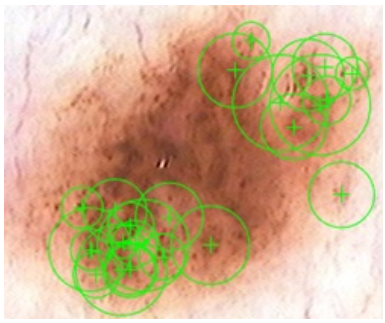


Fig. 3. SURF descriptors in skin lesion image

### *D. Classification*

Classification of an image is a very important step for this kind of projects. When an image is classified according to its visual content, it can provide informations that can effect the final result. There are many classifiers that had been tested, such as Support Vector Machine (SVM) and K-Nearest-Neighbor (KNN).

Deep learning is a new technique to apply machine learning as it is based on learning representations of data which can help in classification. In that it makes use of methods pattern learning methods of data, such an example of input as image, can be mirrored in many different ways, such as a vector of force values per pixel, or in a more abstract way as a set of acnes, districts of a particular shape. Many representations are much better than others at simplifying the learning task.[8]. One of the Deep Learning guarantees, is that the features of different algorithms, will be replaced with unsupervised or semi supervised feature learning and hierarchical feature extraction. A lot of Deep Learning tools such as Deep Neural Networks, Convolutional Neural Networks, Deep Belief Networks and Recurrent Neural Networks have been used in fields like Computer Vision.

*1) Support Vector Machine (SVM):* A support vector machine (SVM) is a classifier and is considered as a very important notion in the field of statistics and computer science for a set of related supervised learning methods, which analyze data and recognize prototypes and used as classification and regression analysis. The principle of SVM is that there are some data points and two or more classes. The goal is to decide in which of these classes each point belongs. This procedure is making the machine a non-probabilistic binary linear classifier. SVM creates a model that separate new samples between the categories. This SVM model is a representation of samples, mapped and separated by a clear gap as wide as possible[9].

*2) K-Nearest-Neighbor (KNN):* K-NN algorithm is a non-parametric method for classification and is a type of instance-based learning, or lazy learning. K-NN is characterized as the simplest of all machine learning algorithms and that because it does not use the training data points to do any generalization. All the training data are used during the testing phase, K-NN makes decision based on the entire training data set and stores all available cases and classifiers new cases based on a similarity measure such as distance functions[10].

*3) Convolutional Neural Networks(CNNs):* Convolutional Neural Networks (CNN) are main tools for deep learning and they are ideal for image recognition because they can learn direct from im- ages. This kind of neural networks has similarities with the neural networks which have been described above: they contain neurons, weights and biases. Each neuron receives an input and gives an output.An important advantage of CNN is that a network can have hundreds of layers that each learn to detect different features of an image. This is the reason why CNNs can be used also for extracting features from an input image and use these features to train a classifier. The output of each layer is used as input in the next layer[11].

CNNs consistence is described by four main types of layers [12]:

**Convolution Layer**: a set of learnable filters which is slided over the image, computing dot products between the entries of the filter and the input image. These filters extend to the full depth of the input image, and they will activate when they see same structure in the images.

**Pooling Layer**: the goal here is to reduce the spatial size of the representation and to reduce the parameters and computation in the network. There are several functions for this purpose, but max pooling is the most common one. For example, a pooling layer of size 2x2 will reduce the input image to one quarter of its original size.

**Non-linear Layer**: In the architecture of a CNN someone can find a variety of functions such as rectified linear units (RELUs) which implements the function $y = max(0, x)$ , so the input and output sizes of this layer are the same.

**Fully-connected Layer**: neurons in this type of layer have fully connections to all activations in the previous layer.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton created a large, deep convolutional neural network named AlexNet, which is trained on more than a million images and can classify images into 1000 object categories and also works as feature extractor. It is working by inserting the available dataset and its ground truth and return the results of the classification. This network comprises of 25 layers total, 5 of those are convolutional layers and 3 are fully connected layers. ReLU is been used for activation function, also, softmax loss is minimized, which means to maximize the corresponding column of the true class in the output vector[13].

## II. Results

For all the experiments, a subset of Vienna dataset was used as described above and the image processing part where the images are thresholded and cropped is the same, so the comparison of results will be using the same data. Also, for all the experiments, the $80\%$ of the data is used for training purpose and the rest $20\%$ for testing purpose.
The summarized classification results on the test images compared with related works, are shown in the tables below. The implementation was made in a laptop with a CUDA-capable NVIDIA$^{TM}$GPU with compute capability 3.0 or higher which is highly recommended for implement deep learning.

| Results/ Method | Sensitivity | Specificity |
|---|---|---|
| Ganster et al. [14] | 87% | **92%** |
| SURF/KNN | **92,48%** | **92%** |

Fig. 4. Ganster et al.[14] compared with results of this research

| Results/ Method | Accuracy |
|---|---|
| Harangi et al. [15] | 86,7% |
| Amirreza et al. [16] | 83,3% |
| Fine-tuned AlexNet | **94,51%** |

Fig. 5. Harangi et al. [15] & Amirreza et al. [16] compared with results of this research

## III. Conclusions and Future Works

### A. Conclusions

According to the importance of an early and quick diagnosis of melanoma, the aim of this master thesis was to develop a system for classifications of skin lesions images. After reviewing the literature on the computer science applications related to melanoma diagnosis, it was really challenging to try these methods and to try to improve them. Analysing images of skin lesions in images, accurate detection of malignant lesions and providing good segmentation of the lesions are important and equivalently challenging steps of automatic melanoma diagnosis. In this research, the images are processed by applying a threshold on them and then define the lesion area using an edge detector.
Experiments revealed that the better way to treat Vienna dataset and classify the images is to use a Convolutional Neural Network (in this case, the pretrained AlexNet) and modify it so it can recognize the required classes for the dataset. Following this procedure, $94,51\%$ accuracy was achieved that outperforms other state-of-art methods.

### B. Future Work

In Computer-Aided Design systems that deal with humans and deadly diseases, it is important to be as more specific as it can be. For that, more tests and analysis using larger datasets are mandatory. There is also room for improvement by analyzing and applying different aspects of software such as balancing, hair removal, etc. To decrease diagnostic time of such a system, an interesting point is the developed software to be applicable in real time.

## References


[1] Bernd Rechel, (2016) , Hospitals in rural or remote areas: An exploratory review of policies in 8 high-income countries
[2] S.Gopinathan, (2016), The Melanoma Skin Cancer Detection and Feature Extraction through Image Processing Techniques.
[3] Omkar Shridha rMurumkar, (2015) , Feature Extraction for Skin Cancer Lesion Detection.
[4] V. Prema, (2016) , Brain cancer feature extraction using OTSU's Thresholding segmentation.
[5] Jigisha M. Patel, (2016) , A review on feature extraction techniques in Content Based Image Retrieval.

[6] Sebastin Castillo-Carrin, (2017) , SIFT optimization and automation for matching images from multiple temporal sources.
[7] Tinne Tuytelaars , SURF: Speeded Up Robust Features.
[8] A. H. Song and Y. S. Lee, (2013), Hierarchical Representation Using NMF.
[9] Nicols Mnera, (2016) , Human features extraction by using anatomical and low level image descriptors from whole body images.
[10] Aman Kataria, (2013) , A Review of Data Classification Using K-Nearest Neighbor Algorithm .
[11] Samer Hijazi , Using Convolutional Neural Networks for Image Recognition.
[12] Doaa A. Shoieb (2016) , Computer-Aided Model for Skin Diagnosis Using Deep Learning.
[13] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, (2012), "Imagenet classification with deep convolutional neural networks."
[14] Ganster, (2001), Automated melanoma recognition.
[15] Balazs Harangi, (2017) , Skin Lesion Detection Based On an Ensemble of Deep Convolution Neural Networks.
[16] Amirreza Mahbod, (2017) , Skin Lesion Classification Using Hybrid Deep Neural Networks.

# Soccer Player(s) video tracking and motion statistics extraction

Charalampaki Eirini

*Abstract*— **One of many important tasks in computer vision applications is real time object detection and tracking. Motion information and visual features, such as shape, color and texture, can be used to extract information from a video, to track and detect objects. Classical tracking algorithms work accurately in perfectly arranged conditions. Although, to efficiently tracking objects is a difficult task due to the variety of the conditions met. The purpose of this paper is to develop a method for effective object tracking and detection in complex scenes, such as sports game.**

## I. INTRODUCTION

In sports, especially soccer, computer vision systems gain tremendous growth due to its numerous uses for sport professionals, trainers and sport game viewers. Soccer video analysis has a wide range of applications, both at a group and individual level. Football players can gain insight into their physiological performance [1], e.g. player's covered distance extraction and trajectory. Coaches can extract information about the quality of tactical analysis and game strategy, strengths or weaknesses evaluation of opponent's team or player and moreover, the verification of referee decisions.

Since tracking objects process is a difficult task due to the variety of the conditions met, different approaches have been proposed. Some researches for distinguishing the athletes from the background, relied on color of pants and shirt. Color-based classification techniques can be divided into template-based player detection and pixel-based detection. In the template based player detection approaches, each window in the image is classified into player or non-player [2]. On the other hand, each pixel is classified by its color component in pixel-based player detection methods [3] The main disadvantage of these methods is that players image might get fragmented into multiple regions. In [4] authors combined players features based particle filter, number detection likelihood model and a motion vector prediction model for tracking multiple players in 3D space. They map 2D images to 3D space in the real world, by using multiple fixed cameras, which allow them to track players in 3D space. To distinguish players with different jersey number, they use the number detection likelihood model and to predict players position when occlusion occurred, they use a motion vector. Authors in [5], proposed an automatic player detection by combining boosting detection and background modeling, unsupervised labeling and efficient player tracking method using Markov Chain Monte Carlo data association applied for broadcast soccer videos. However, whenever the video is blurred, the camera is moving suddenly and unexpectedly or long occlusions occurred, their proposed method leads to failure. In the literature, in videos captured by static cameras, background subtraction method has been extensively used for segmenting moving players based on motion information [6], [7]. Nevertheless, in case of moving camera, the accuracy of this method is difficult. In this paper [8], a blob-guided Particle Swarm Optimization is introduced for efficiently searching multiple players in an image. Authors, based on the number of blobs, divided the swarm into sub-swarms and search for players in each blob. The blob-guided Particle Swarm Optimization detect effectively multiple players, but several open issues need to be investigated. In literature, the discriminative color of the players' uniforms has been widely used, although there are unreliable results using just color information, to distinguish blurred and distant players. Moreover, if the players' uniform colors are similar color with the playfield, advertising billboards and lines, the detection results are uncertain.

The rest of the paper is organized as follows: In Section 2 we present the problem definition, in Section 3, we describe the implementation of the selected tracking algorithms. The algorithm evaluation in the case of the football players is provided in Section 4. In Section 5, the motion statistics extraction and the experimental resuls are presented. Finally, Section 6 conludes the paper.

## II. PROBLEM DEFINITION

Classical tracking approaches work accurately under perfectly arranged conditions. On the other hand, novel tracking algorithms depend upon multiple cameras to extract as many motion information as they can of the object that is tracked. This paper's motivation, is the craving of a straightforward application of tracking football players, without the use of multiple cameras. Our goal is to overcome the challenges of tracking in a complex scene, such as a football match and implement a system that tracks effectively one or/and more football players, with the convention that a single mobile camera is going to be utilized. The following goals has been defined for this paper.

- A single mobile device is used.
- Track one player in the reference sequences.
- Track multiple players in the reference sequences.
- Extract information about the position and the covered distance

Limitations: The following limitations has been set for this paper: According to the initial scenario the software will run to a mobile device and thus only one camera will be available.

This work was not supported by any organization

Charalampaki Eirini submitted this paper for the Degree of Masters in Computer Vision 2017, University of Burgundy, France Eirini_Charalampaki@etu.u-bourgogne.fr

## III. METHODOLOGY

For our implementation, all video sequences are recorded in a live football match, taking place in a mini 5x5 football field with 30 meters x 60 meters size. We choose a steady mobile camera, which only covers a part of the football field. The resolution of the video sequence is high definition 1280x720 size in pixels and the frame rate is 30 frames per second. The C++ interface of open source library OpenCV 3.2 is used inside the Microsoft Visual Studio 2015[9],[10].

### *A. Track a single football player*

In our application, firstly the input video sequence is specified as parameter of the application. Several variables need to be declared. The region of interest (roi) parameter is necessary to record the bounding box of the tracked object and this variable's stored value will be updated using the tracker object. Moreover, a frame variable requires to be set to hold the image data from each frame of the input video. Then, a tracker is created by its name and the selected algorithms to be used are MIL, KCF, TLD and Median Flow. At that point, we initialize the variables to store the motion extraction results, such as the distance, the selected objects center, the previous x position of the object and the previous y position of the object. The previous x and y position of the object to be tracked, is assigned to an original x and y variable respectively. The tracking process begins within a for loop from the first frame of the video sequence until the last one. For finding the new most likely bounding box for the target, we update the tracker and pass the result to the region of interest variable and our application draws the tracked object. At that point, the tracked object's distance is calculated, by using the Euclidean distance formula. In figure 1 below, the frame number on the current frame is obtained and written in the top left of our GUI screen, and the selected football player is tracked throughout the whole video sequence.

### *B. Track multiple football players*

Here, we create a multi tracker object instead of a single tracker object, which is created in the previous implementation. The difference in this approach, is that a container of the multiple tracked objects should be defined.

Moreover, a variable where the center of each selected player will be stored, is declared. In our proposal, the multi tracker object, uses the same tracking algorithm for all the objects to be tracked. For each tracked player, there is, though, the option of using different type of tracking algorithm. To achieve this, the tracking algorithm should be defined, whenever an extra object is added to the multi tracker object, however this option is somewhat out of the scope for this paper. After the creation of the multi tracker object and the setup of the input video sequence, the user is able to select multiple football players to be tracked. Once again, the region of interest function is used, to select multiple football players, with the result saved in the container, that is mentioned above. Afterwards, the tracker is initialized for the first frame of the video and an infinite loop signifies the starting point of the tracking process throughout the whole video sequence. At this point, the tracking result is update, nevertheless a second difference from the single object approach is observed. Considering the fact that the purpose of this approach is to track multiple football players, the drawing process is slightly different. A second for loop is started, drawing the total number of the tracked multiple players. The range of this for loop, starts from 0 to the number of football players, that the user previously selected. Once more, a green circle is defining the center of each tracked object and the motion information as well the position of each tracked player are stored in a text file. To calculate the processing speed, we start a timer

Fig. 1. The result of the tracking process for a single player

Fig. 2. the result of the tracking process for multiple football players

that returns the number of clock ticks since our application started. As soon as our tracker is updated, we subtract the number of clock ticks from that counter and divide the clocks per sec with that timer to calculate the processing speed. The processing speed can be seen in the top left corner of the GUI window.

## IV. ALGORITHM EVALUATION IN THE CASE OF FOOTBALL PLAYERS

For our system, we choose a steady mobile camera, which only covers a part of the football field. The football field is mini 5 x 5 with dimensions 30 x 60 meters. Our application is implemented using OpenCV and C++. The input video sequences resolution is 1280x720 size in pixels and the frame rate is 30 frames per second. The application is tested in two video sequences. The purpose of the first scene is to show that one player can be tracked effectively with the selected algorithms. The second sequence also contains all the players, the players are occluded during some frames. The purpose of the second sequence is to demonstrate that the players can be tracked during occlusion. KCF outperforms all the other tracking algorithms in speed and reliability, throughout the whole video sequence. TLD is much slower than MIL and KCF, but managed to track the selected football player effectively. However, Median Flow tracking algorithm, although is fast, it loses tracking target after some frames. KCF once more is faster in the multi players implementation and it performs sufficiently under partial occlusion. MIL is reasonable efficient under partial occlusion and it keeps tracking the selected targets, however when the players are fully occluding one another, this tracking algorithm fails. Median Flow algorithm achieves better results when no occlusion is occurring and the motion is predictable. TLD algorithm is the slowest of all algorithms tested in the multi players video sequence. TLDs authors claim that in partial occlusion and scale changes this algorithm prevails, however we could not confirm that statement for our experiments, for the reason that it failed after a few frames.

## V. MOTION STATISTICS EXTRACTION AND EXPERIMENTAL RESULTS

The motion statistics and specifically the x and y position of the tracked players, is extracted in an output file. Our goal is to exploit this information to represent the perspective video tracking coordinates to orthographic 3D coordinates in the 3D space, using X3DOM. The focal length of the camera is 29mm. We transform u, v coordinates of the camera to correspond to the world coordinates (X3DOM). Moreover, the position in 3D space is going to be displayed in respect

Fig. 3. The two tracked players are fully occluded

Fig. 4. The KCF tracker does not recover from full occlusion

| | Frame number | KCF # not found | KCF #found | Tracking reliability KCF | occlusion |
|---|---|---|---|---|---|
| Player 1 | 50 | 0 | 50 | 100% | no |
| Player 2 | 50 | 10 | 40 | 80% | Partial |
| Player 3 | 50 | 50 | 0 | 0% | full |

| | Frame number | TLD # not found | TLD #found | Tracking reliability TLD | occlusion |
|---|---|---|---|---|---|
| Player 1 | 50 | 50 | 50 | 100% | no |
| Player 2 | 50 | 50 | 0 | failed | partial |
| Player 3 | 50 | 50 | 0 | failed | full |

| | Frame number | MIL # not found | MIL #found | Tracking reliability MIL | occlusion |
|---|---|---|---|---|---|
| Player 1 | 50 | 0 | 50 | 100% | no |
| Player 2 | 50 | 20 | 30 | 60% | partial |
| Player 3 | 50 | 50 | 0 | failed | full |

| | Frame number | Median Flow # not found | Median Flow #found | Tracking reliability Median Flow | Total occlusion |
|---|---|---|---|---|---|
| Player 1 | 50 | 22 | 28 | 56% | no |
| Player 2 | 50 | 35 | 15 | 30% | partial |
| Player 3 | 50 | 50 | 0 | failed | full |

Fig. 5. Multiple players tracking results

to the upper left corner. The dimension of the object is not mandatory, we only interested in object's position. To determine which screen x-coordinate corresponds to a point at model x coordinate and the subject distance, we need to multiply the point coordinates by the focal length. In our case, this becomes

$$U = f * \frac{x}{|z|}$$

$$V = f * \frac{y}{|z|}$$

for the U, V coordinates (in pixels) respectively. We assume that $y_1 = 1m - H$, and to calculate the unknown parameters of x position and z coordinate (depth) in 3D space, the above equations become

$$x_1 = U_1 * \frac{|z_1|}{f}, \texttt{x coordinate calculation}$$

$$z_1 = y_1 * \frac{f}{v_1}, \texttt{z coordinate calculation}$$

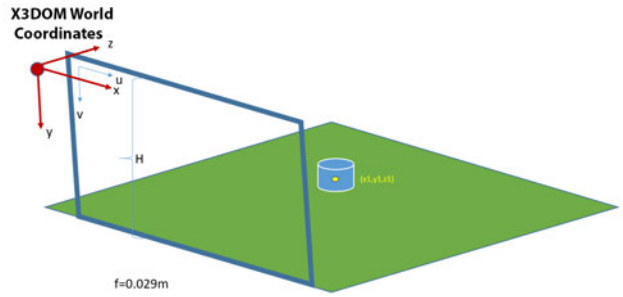



Fig. 6. Simplified Transformation of perspective video tracking coordinates to orthographic 3D coordinates

Fig. 7. the position of 3 players in the first frame

The tracked objects distance is calculated, by using the Euclidean distance formula. In a 3 dimensional plane, the

| player# | u | v | player# | u | v | player# | u | v |
|---|---|---|---|---|---|---|---|---|
| 0 | 480 | 359 | 1 | 642 | 340 | 2 | 957 | 321 |
| 0 | 477 | 363 | 1 | 639 | 344 | 2 | 952 | 325 |
| 0 | 476 | 365 | 1 | 637 | 347 | 2 | 948 | 329 |
| 0 | 472 | 364 | 1 | 634 | 347 | 2 | 942 | 331 |
| 0 | 467 | 363 | 1 | 628 | 347 | 2 | 934 | 331 |
| 0 | 463 | 363 | 1 | 623 | 348 | 2 | 928 | 332 |
| 0 | 460 | 367 | 1 | 619 | 353 | 2 | 922 | 336 |
| 0 | 457 | 370 | 1 | 616 | 355 | 2 | 917 | 339 |
| 0 | 456 | 370 | 1 | 614 | 353 | 2 | 913 | 337 |
| 0 | 454 | 369 | 1 | 611 | 349 | 2 | 909 | 334 |
| 0 | 449 | 370 | 1 | 608 | 348 | 2 | 904 | 334 |

Fig. 8. the position of 3 players in the first frame

| player | U | V | f | X | Y | Z |
|---|---|---|---|---|---|---|
| 0 | 11,2933 | 8,44642 | 29 | 12300,8 | 9200 | 31587,3 |
| 0 | 11,2227 | 8,54053 | 29 | 12089,3 | 9200 | 31239,3 |
| 0 | 11,1992 | 8,58759 | 29 | 11997,8 | 9200 | 31068,1 |
| 0 | 11,105 | 8,56406 | 29 | 11929,7 | 9200 | 31153,4 |
| 0 | 10,9874 | 8,54053 | 29 | 11835,8 | 9200 | 31239,3 |
| 0 | 10,8933 | 8,54053 | 29 | 11734,4 | 9200 | 31239,3 |
| 0 | 10,8227 | 8,63465 | 29 | 11531,3 | 9200 | 30898,8 |
| 0 | 10,7521 | 8,70523 | 29 | 11363,2 | 9200 | 30648,2 |
| 0 | 10,7286 | 8,70523 | 29 | 11338,4 | 9200 | 30648,2 |
| 0 | 10,6816 | 8,6817 | 29 | 11319,2 | 9200 | 30731,3 |
| 0 | 10,5639 | 8,70523 | 29 | 11164,3 | 9200 | 30648,2 |

Fig. 9. First player's coordinates X3DOM

| player | U | V | f | X | Y | Z |
|---|---|---|---|---|---|---|
| 1 | 15,1047 | 7,9994 | 29 | 17371,8 | 9200 | 33352,5 |
| 1 | 15,0342 | 8,09351 | 29 | 17089,5 | 9200 | 32964,7 |
| 1 | 14,9871 | 8,16409 | 29 | 16888,8 | 9200 | 32679,7 |
| 1 | 14,9165 | 8,16409 | 29 | 16809,2 | 9200 | 32679,7 |
| 1 | 14,7754 | 8,16409 | 29 | 16650,1 | 9200 | 32679,7 |
| 1 | 14,6577 | 8,18762 | 29 | 16470,1 | 9200 | 32585,8 |
| 1 | 14,5636 | 8,30526 | 29 | 16132,6 | 9200 | 32124,2 |
| 1 | 14,493 | 8,35231 | 29 | 15963,9 | 9200 | 31943,2 |
| 1 | 14,446 | 8,30526 | 29 | 16002,3 | 9200 | 32124,2 |
| 1 | 14,3754 | 8,21115 | 29 | 16106,6 | 9200 | 32492,4 |
| 1 | 14,3048 | 8,18762 | 29 | 16073,6 | 9200 | 32585,8 |

Fig. 10. Second player's coordinates X3DOM

| player | U | V | f | X | Y | Z |
|---|---|---|---|---|---|---|
| 2 | 22,516 | 7,55237 | 29 | 27428 | 9200 | 35326,6 |
| 2 | 22,3983 | 7,64648 | 29 | 26948,9 | 9200 | 34891,9 |
| 2 | 22,3042 | 7,74059 | 29 | 26509,4 | 9200 | 34467,6 |
| 2 | 22,163 | 7,78765 | 29 | 26182,5 | 9200 | 34259,4 |
| 2 | 21,9748 | 7,78765 | 29 | 25960,1 | 9200 | 34259,4 |
| 2 | 21,8337 | 7,81118 | 29 | 25715,7 | 9200 | 34156,2 |
| 2 | 21,6925 | 7,90529 | 29 | 25245,2 | 9200 | 33749,6 |
| 2 | 21,5748 | 7,97587 | 29 | 24886,1 | 9200 | 33450,9 |
| 2 | 21,4807 | 7,92882 | 29 | 24924,6 | 9200 | 33649,4 |
| 2 | 21,3866 | 7,85823 | 29 | 25038,3 | 9200 | 33951,7 |
| 2 | 21,269 | 7,85823 | 29 | 24900,6 | 9200 | 33951,7 |

Fig. 11. Third player's coordinates X3DOM

distance between points (X1, Y1, Z1) and (X2, Y2, Z2) is given by:

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2}$$

In our case XYZ values are in millimeters. To calculate the speed, we divide the distance traveled by the moving object with the time taken to travel distance d. The average speed is the total distance traveled divided by the total time taken. For example, for some of our distance values, an average speed would be 4268 mm/sec.

| distance | time (1/fps) |
|---|---|
| | |
| 407,33 | 0,033 |
| 194,067 | 0,033 |
| 127,18 | 0,033 |
| 101,37 | 0,033 |
| 396,45 | 0,033 |
| 396,47 | 0,033 |
| 301,758 | 0,033 |
| 24,8 | 0,033 |
| 85,289 | 0,033 |
| 175,78 | 0,033 |

Fig. 12. The calculated distance for the first player

## VI. CONCLUSIONS

The purpose of this paper was to examine if it is possible to develop an application for effective object tracking and detection in complex scenes, such as sports game and specifically in a football game. This paper has shown that this is undoubtedly achievable with an adequate precision and accuracy in the chosen video sequences. The implementation was accomplished with the C++ interface of open source library OpenCV 3.2 inside the Microsoft Visual Studio 2015 and the MIL, KCF, TLD and Median Flow online training tracking algorithms were tested. The presence of noise, scene illumination, object shape variation and total or partial occlusion lead to significant problems in computer vision systems and make efficiently tracking process into a challenging task. Football player tracking and detection is a fundamental procedure in nearly all football video analysis. The results can be utilized for both a group and individual level. Football players can gain insight into their physiological performance, e.g. player's covered distance extraction and trajectory. Coaches can extract information about the quality of tactical analysis and game strategy, strengths or weaknesses evaluation of opponents team or player and moreover, the verification of referee decisions. Furthermore, players' tracking information can be used by the broadcaster of a football match to provide enhanced replays and statistical analysis for the footballs team supporter. The method can also be applied to a wide range of applications such as vision-based human-computer interaction. A huge amount of work required to be done, to show that this implementation is sufficient in a larger scale during a live football game. However, this paper has shown that the effectiveness of OpenCV with the combination of novel tracking algorithms have a huge potential and can bend the arising difficulties.

## REFERENCES


[1] P. G. O Donoghue, M. Boyd, J. Lawlor, and E. W.Bleakley. Time-motion analysis of elite, semi-professional and amateur soccer competition. volume 41, pages 1-12. TEVIOT SCIENTIFIC, 2001.

[2] Sun L, Liu G (2009) Field lines and players detection and recognition in soccer video. In: Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, 1924 April 2009 .

[3] Vandenbroucke N, Macaire L, Postaire J-G (2003) Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis. Computer Vision Image Understanding .

[4] [Xizhou Zhuang, Xina Cheng, Shuyi Huang, Masaaki Honda, Takeshi Ikenaga, Motion Vector and Players Features Based Particle Filter for Volleyball Players Tracking in 3D Space, Advances in Multimedia Information Processing - PCM 2015. Lecture Notes in Computer Science.

[5] Jia Liu, Xiaofeng Tong, Wenlong Li, Tao Wang, Yimin Zhang, Hongqi Wang, Bo Yang, Lifeng Sun, Shiqiang Yang Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video

[6] Choi K, Seo Y (2011) Automatic initialization for 3D soccer player tracking. Pattern Recogn Lett 32(9): 12741282. doi:10.1016/j.patrec.2011.03.009

[7] DOrazio T, Leo M, Spagnolo P, Mazzeo PL,Mosca N, Nitti M, Distante A (2009) An investigation into the feasibility of real-time soccer offside detection from a multiple camera system. IEEE Trans Circ Syst Video 19(12):18041818. doi:10.1109/TCSVT.2009.2026817

[8] M. Manafifard, H. Ebadi, H. Abrishami Moghaddam, Multi-player detection in soccer broadcast videos using a blob-guided particle swarm optimization method, Springer Science+Business Media New York 2016

[9] OpenCV download [Online]. Available: http://opencv.org/ [Accessed February 2017]

[10] MicrosoftVisualStudio[Online].Available: https://www.visualstudio.com [Accessed February 2017]

# A quantitative comparison of deep convolutional neural networks for brain tissue segmentation on magnetic resonance images

Jose Bernal, Sergi Valverde, Oliver Arnau and Xavier Lladó

*Abstract*— **Sensitive medical applications require accurate, reliable, robust and general tools to diagnose and monitor brain lesions and atrophy, for instance. In terms of accuracy, results from grand challenges of relevant international conferences are a reference. However, the case is not the same when considering reliability and robustness since usually the algorithms are specifically tweaked for achieving high scores on one dataset or are complex in practical terms. In this paper, we identify advantages and disadvantages from state-of-the-art algorithms for brain tissue segmentation and, accordingly, propose more robust techniques. The proposal consists of 33 CNN architectures of literature-inspired approaches comprising (i) fusion of single and multiple sources of information, (ii) explicit brain positioning information and (iii) implicit spatial information acquired from triplanar and 3D patches. The different architectures were evaluated using three public datasets comprising images from infants or adults, healthy or unhealthy subjects – with varied atrophy extent – and with different imaging quality and resolution. This allowed us to assess the algorithms in terms of robustness and usefulness in different scenarios. The results of the experiments showed that higher performance, in terms of surface overlap, distance and difference, could be achieved by integrating the mentioned factors in one single approach – if possible. The contributions of this work will be submitted in August 2017 to the 6-month infant tissue segmentation grand challenge at the Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2017 conference.**

## I. Introduction

Nowadays, biomarkers based on measurements on changes in brain tissue are used for diagnosing and monitoring several diseases [1]. Hence, highly accurate and reliable tools performing brain Magnetic Resonance Imaging (MRI) tissue segmentation are required. The Convolutional Neural Networks (CNNs) have been useful in this sense since they are able to identify complex spatial distribution of the tissues and their relationship at local and structural levels. Although there are usually fewer classes in these segmentation tasks compared to large-scale semantic image recognition, there are two major difficulties hindering achieving better accuracy than human raters. First, there is a lack of sufficiently labelled training data. Essentially, generating highly accurate labels and finding sufficient pre-processed and representative data to the underlying problems require large amounts of time. Second, medical image annotation is carried out by experts and is time-consuming, subjective, and error-prone. Learning a model from a less accurate representation of training samples degenerates the algorithm accuracy [2].

Several public brain MR image datasets are available to the community, especially those organised by Medical Image Computing and Computer Assisted Intervention (MICCAI[1]) society, strongly encouraging research and publications in the field. These standard evaluation frameworks have been proposed for quantitatively comparing tissue segmentation algorithms under the same directives and conditions. Although they have certainly carried out their mission successfully, algorithms are, in general, tweaked to obtain high accuracy on those specific datasets. Two negative situations are distinguished in this sense: (i) the algorithm outperforms for a specific dataset, but, with the same configuration, it is not able to achieve great scores on another one; and (ii) the algorithm uses a very complicated set of steps which are impractical in realistic scenarios. These situations in terms of the challenges may seem insignificant since the ranking is based only on the performance and not on other characteristics (e.g. transferability). However, these items are crucial in real-life applications for which reliable, robust and general tools for supporting medical decisions are required.

In this paper, we evaluate several state-of-the-art methods for brain tissue segmentation. Based on the analysis of these models, we identify and analyse potential modifications to increase their robustness under different imaging conditions. We evaluate both the original and the proposed CNN architectures on different public MRI data. First, identified pros and cons from the state-of-the-art methods are described in Section II. Second, the proposed architectures are described in Section III. Third, the proposals are evaluated on two datasets from MICCAI grand challenges (iSeg2017 and MRBrainS13) and one publicly available dataset (IBSR18). Fourth, the results are reported in Section IV. Fifth, the discussion of our work is presented in Section V.

## II. State of the art

Representation learning appears as an alternative to ad-hoc and highly engineered methods since they discover automatically detection- and classification-suitable representations from the input data. CNN, a branch in this research line, identifies compositional hierarchy features which objects from the real world exhibit: low level features (e.g. edges) form patterns and these specific patterns form high level ones (e.g. shapes, textures). Since these strategies have shown record-shattering performances in a variety of computer vision problems, they have been rapidly spread and extended for brain MRI tissue segmentation. The workflow of most of the approaches is threefold. First, the data is prepared for being input to the network by extracting patches from the region of interest. Second, the patches are input to the network to obtain a classification. A single classification is output per patch. Third, the image is reconstructed by

[1] http://www.miccai.org/

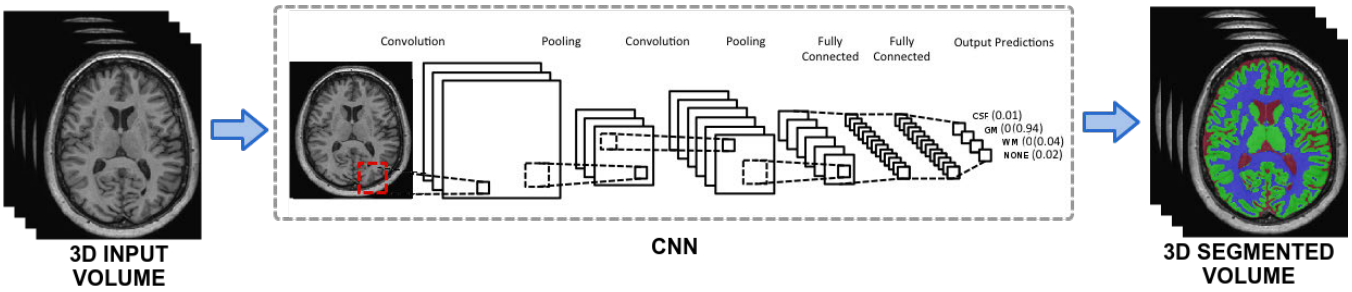


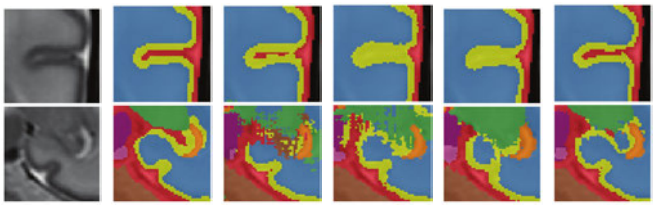

Fig. 1: Brain MRI tissue segmentation using CNNs. In the segmented volume image, the red, the green and the blue labels indicate CSF, GM and WM, respectively. Original image taken from `https://ujjwalkarn.me`.

piling up the obtained labels. This resulting volume contains only three classes grey matter (GM), white matter (WM) and cerebrospinal fluid (CSF). Additional post-processing techniques can be applied on top of the segmented brain, for instance, to remove isolated regions. A diagram of the common segmentation workflow is depicted in Fig. 1.

The most common variations of the architectures are detailed as follow.

The architectures can be grouped according to the input patch into 2D [3], [4], 2.5D [5] and 3D [6], [7], [8]. Two-dimensional CNNs disallow for exploiting of the actual 3D nature of the MRI data. In this sense, 2.5D and 3D CNNs were introduced mainly to overcome this limitation. Although both 2.5D and 3D architectures require more processing than 2D ones, the former type is less computationally expensive than the second one: 2D convolutional layers are used in 2.5D while 3D convolutional are used in 3D. Indeed, the main impediment of working with 3D CNNs consist of the memory requirements, expensive computational costs and slow speed. Moreover, 3D CNNs use a larger number of parameters. Processing the whole 3D volume requires large CPU or GPU memory, and processing will be slow. On the other side, Milletari *et al.* [9] reported that 3D CNNs require smaller number of training samples to surpass the performance obtained by 2D and 2.5D CNNs (13.5 millions patches for 2D and 2.5D versus 1.35 million patches for 3D CNN). This finding shows that 3D CNNs extract more discriminative features, taking advantage of a larger dimension than their 2D and 2.5D counterparts.

Some CNN variants consider integrating different operating modules. Based on the number of independent operative networks integrated into a single model, the networks can be classified into single-path [3], [7], multi-path [5], [4], [6] and cascaded [8] architectures. Multi-path architectures consist in fusing the knowledge from working modules to extract various useful features and, later, aggregated to achieve the desired segmentation. An illustrative example taken from the paper of Moeskops *et al.* [4] is presented in Fig. 2. However, training all the networks arranged in parallel at the same time might demand a considerable amount of memory. Cascaded architectures opt for reprocessing the output of one CNN using another one. In this sense, the classification is refined by reducing the number of misclassified voxels. Nevertheless, this approach requires careful design and training phases to work correctly.



(a) (b) (c) (d) (e) (f)

Fig. 2: Segmentation results using different patch size configurations [4]. The original image is shown in (a), the ground truth in (b) and the corresponding segmentations using (c) only patches of $25 \times 25$, (d) only patches of $51 \times 51$, (e) only patches of $75 \times 75$ and (f) combination of the three sources.

The network can be provided with a single imaging sequence [4], [7] or multiple modalities [3], [5], [8]. According Zhang *et al.* [3], the second option should be addressed since different modalities may bring up valuable contrast information among classes. For instance, multiple sclerosis lesions can be classified easily than in T1-w sequences using the FLAIR modality since lesions appear hyperintense.

The classifier can output a single label [3], [5], [4] or multiple [7], [6], [10] at a time. While the second case corresponds to the Fully CNN (FCNN). The latter approach produces a faster and larger response than the former, but this strategy usually requires a large number of parameters to be set and, consequently, more training samples.

The fact that the architectures process patches separately and not the whole volume has some implications: spatial distribution of brain structures is not directly encoded. Contextual information can be implicitly encoded using 2.5D, 3D and multi-scale architectures but this information depends on the size of the input patch. The bigger the patch size, the more context is given to the network but the more resources to process it are needed. Another option is to consider within-brain positioning features as presented by Wachinger *et al.* [7] in which Cartesian and spectral coordinates are provided to the network. It is important to have in mind that (i) considering XYZ-coordinates on the image plane makes sense only if the volumes are registered and (ii) the distribution of the eigenvectors depends highly on the shape of the brain mask, i.e. these features may not be reliable

when the brain shape differs considerably from one volume to another (e.g infant brains).

## III. Method

According to the literature review, we propose several variants which are built upon modules of the work of Moeskops *et al.* [4] and the three FCNN architectures proposed by Dolz *et al.* [6] as a starting point. In this sense, we work with both CNN and FCNN architectures and evaluate whether there is any additional difference apart from the computational time. The baselines as well as the extensions are described as follows. A diagram of two of the proposed architectures, Left Moeskops$_{2.5D+2.5D+spatial}$ and Dolz multi$_{3D+3D+spatial}$ is displayed in Fig. 3.

### A. Baseline architectures

*1) Moeskops$_{2D+2D+2D}$ and Left Moeskops$_{2D}$:* The Moeskops$_{2D+2D+2D}$ architecture was implemented following the work of Moeskops *et al.* [4], here upon referred as Moeskops' architecture. This network was formed by three branches which inputs shared the same target pixel but have different sizes. This allowed the architecture to obtain multi-scale information – and possibly spatial context as well – to produce the classification label of the target pixel. Each path consisted of three convolutional layers with $24$, $32$ and $48$ kernels, followed by max-pooling layers with stride size of $2 \times 2$. Once the three convolutions took place, the output of the third layer was flattened and input into a dropout layer. The dropout prevents the network from memorising the training samples. Then, the resulting vector is used in a fully-connected layer with $256$ nodes. The resulting vectors from the three branches are merged (e.g. averaged) and subsequently connected to the softmax output layer with three nodes. The differences between branches is that (i) the input sizes of the red, green and blue branches are $75 \times 75$, $51 \times 51$ and $25 \times 25$, respectively; (ii) the kernel sizes of the three convolution layers in the three branches are $9 \times 9$, $7 \times 7$ and $5 \times 5$, $7 \times 7$, $5 \times 5$ and $3 \times 3$ and $5 \times 5$, $3 \times 3$ and $3 \times 3$ (hereupon called Left Moeskops$_{2D}$); (iii) no max-pooling is applied after the third convolutional layer in the Left Moeskops; (iv) feature maps are padded when required to be fully covered by even-sized max-pooling kernels.

### B. Dolz base$_{3D}$, Dolz single$_{3D}$ and Dolz multi$_{3D}$

The first network presented by Dolz *et al.* [6], called base, was composed of three convolutional layers with $25$, $50$ and $75$ kernels all of size $7 \times 7 \times 7$. Each one of these layers was immediately followed by a PReLU layer which rectifies the partial results. Then, the obtained feature maps were input into three consecutive fully connected layers. After that, the prediction took place by using a softmax output layer with four classes (i.e. one additional class for the background).

The second network, referred as single, corresponded to an extension of the base architecture. In this case, each convolutional layer of $7 \times 7 \times 7$ kernels was subdivided into three layers of $3 \times 3 \times 3$. As explained by the authors, they were able to generate a deeper architecture – more complex features could be learnt – while reducing overfitting risk [6].

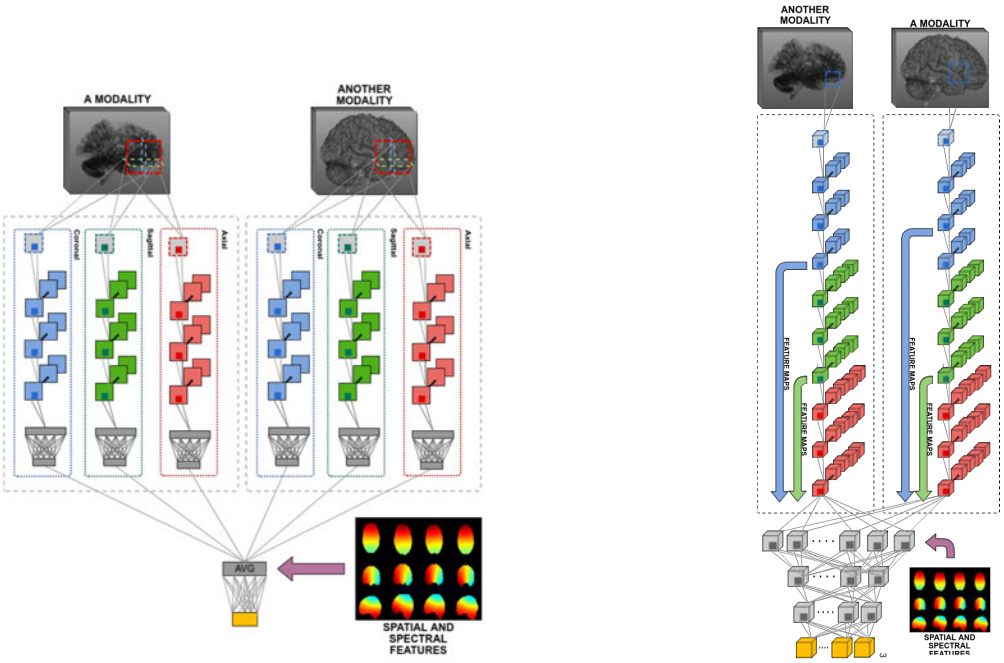
The third FCNN, called multi, took into account feature maps from previous layers to produce the final output. The feature maps coming from shallower layers were cropped to obtain the same dimensions as in the last convolutional layer and subsequently merged. This kind of connections allowed the network to learn from semantic – coming from deeper layers – as well as fine-grained localisation information – coming from shallow layers.

### C. Proposed extensions

Exploiting the 3D nature of MRI data can be beneficial to the networks since a sense of spatial distribution of the different brain structures is supplied. According to the literature, the network becomes 3D aware when considering patches from the three anatomical planes or 3D patches themselves. We considered extending Left Moeskops$_{2D}$ to a 2.5D version, expressed as Left Moeskops$_{2.5D}$, to evaluate this aspect. The merging process takes place immediately before the prediction step following Moeskops *et al.* [4]. In that sense, the merging weights are set up automatically regardless the considered scenario.

As mentioned previously, one of the problems of 3D architectures is that they may require large number of training samples to reach desired segmentation results. Since MRBrainS13 contains a lower number of training volumes than the other two evaluation datasets, it is likely that the 3D architectures exhibit problems. Thus, we consider 2D versions out of the three Dolz architectures, denoted by Dolz base$_{2D}$, Dolz single$_{2D}$ and Dolz multi$_{2D}$, to corroborate this situation. The 2D variants from the three 3D architectures are crafted by placing 2D convolutional layers instead of 3D ones. Accordingly, for each input patch of size $27 \times 27$, a segmented patch of size $9 \times 9$ is produced.

The networks can be equipped with explicit spatial information following the approach of Wachinger *et al.* [7]. We considered modifying our previously defined variants to mine this information. The information is input in the form of $1 \times 6$ (3D spatial and spectral coordinates) vectors and $9 \times 9 \times 9 \times 6$ (3D spatial and spectral coordinates per $9 \times 9 \times 9$ block to classify) matrices in the case of the CNN and the FCNN architectures, respectively. The aspect we intend to evaluate with these architectures is the influence of spatial and spectral coordinates in the performance of 2D CNN, 2.5D CNN and 3D FCNN networks. It has been said that 2.5D and 3D approaches implicitly encode contextual information, but this context highly depends on the size of the patch. The larger the patch, the more information the network can take into account to produce a verdict. However, this is not the case since the input patches for our CNNs are of dimensions $25 \times 25$ and for FCNNs of dimensions $27 \times 27 \times 27$ (and $27 \times 27$ input patches in the 2D FCNN variants). Thus, we should expect improvements regardless the variants discussed so far. The architectures extended with explicit spatial information are identified with the subscript spatial (e.g. the variant of Left Moeskops$_{2.5D}$




(a) (b)

Fig. 3: CNN and FCNN proposed architectures. In (a), Left Moeskops$_{2.5D+2.5D+spatial}$ architecture. The network corresponds to an extension of Left Moeskops$_{2D}$ in which patches from the three anatomical planes are extracted from two modalities. Additionally, explicit spatial information is introduced in the network. In (b), Dolz multi$_{3D+3D+spatial}$ architecture. The network corresponds to an extension of Dolz multi$_{3D}$ in which spatial information is input and 3D patches from two different modalities are extracted.

equipped with spatial information would be denoted by Left Moeskops$_{2.5D+spatial}$).

To gain valuable contrast information among classes, we propose multi-path extensions built upon each one of the previously described architectures. The idea is to replicate the architecture to process in parallel information from two different modalities at the same time. The merging phase differs depending on the baseline. The merging step is carried out before the prediction layer for CNN-based architectures and before the $1 \times 1 \times 1$-kernel convolutional layers for FCNN-based ones ($1 \times 1$ for 2D FCNN). In this particular case, we refer to the multi-path networks by indicating the patch dimension of each of the branches. For instance, if we append two Left Moeskops$_{2.5D}$ branches, each one of them for T1 and T2-FLAIR, the resulting architecture would be denoted by Left Moeskops$_{2.5D+2.5D}$.

## IV. Experiments and results

Public datasets are commonly used for assessing brain MRI tissue segmentation algorithms. Here, we consider two challenges and one publicly available repository: iSeg2017, MRBrainS13 and IBSR18, respectively. The datasets were chosen to evaluate the algorithms since (i) they have been widely used in the literature to compare different methods and also (ii) they give the chance to test the algorithms on healthy and patients data, infants and adults, with different pixel spacing and varied number of samples. We believe that these two factors allow us to see how robust, generalised and useful in different scenarios the algorithms can be. The different algorithms are evaluated using Dice similarity coefficient (DSC), Modified Hausdorff distance (MHD) and absolute volumetric difference (AVD).

The results are presented in the following sections. It is important to note that (i) the results of only ten architectures – out of the proposed 33 – are displayed in the plots to focus the attention of the reader on the relevant observations, and (ii) only DSC results are reported since the performance of the algorithms was similar in terms of MHD and AVD.

### A. Cross-validation results on the MRBrainS13 training set

The results regarding the DSC are shown in Fig. 4. The algorithm performing the best at leave-one-out cross-validation in terms of DSC was Left Moeskops$_{2.5D+2.5D+SC}$ and the worst the same approach using only 2D patches of T1-w. Additionally, it can be seen that including explicitly contextual information either from spectral and spatial coordinates or 2.5D extensions is beneficial for the architectures since the original networks are improved in more than $2\%$. Also, the plot highlights an important drawback of Dolz-based architectures. In contrast to the expectations, a 2D version of Dolz multi outperforms the original version. This is a consequence of the number of parameters to optimise in the 3D FCNN (most of them coming from the $1\times1\times1$-kernel layers). To illustrate the problem, the number of parameters in Dolz multi$_{2D}$ is approximate $574\,053$ while the ones in Dolz multi$_{3D}$ rise to $3\,332\,595$. It is important to highlight

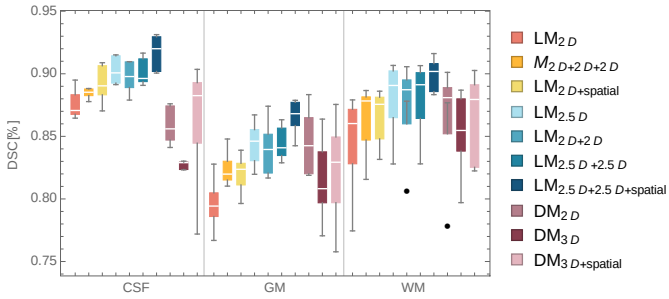



Fig. 4: DSC obtained from leave-one-out cross-validation on the MRBrainS13 training set. In the legends, the abbreviations LM, M and DM stand for Left Moeskops, Moeskops and Dolz multi, in that order.

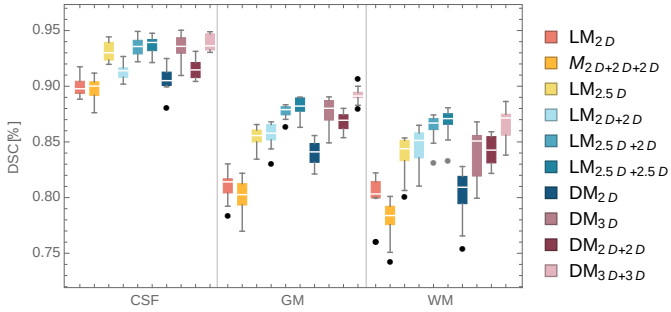



Fig. 5: DSC obtained from leave-one-out cross-validation on the iSeg2017 training set. In the legends, the abbreviations LM, M and DM stand for Left Moeskops, Moeskops and Dolz multi, in that order.

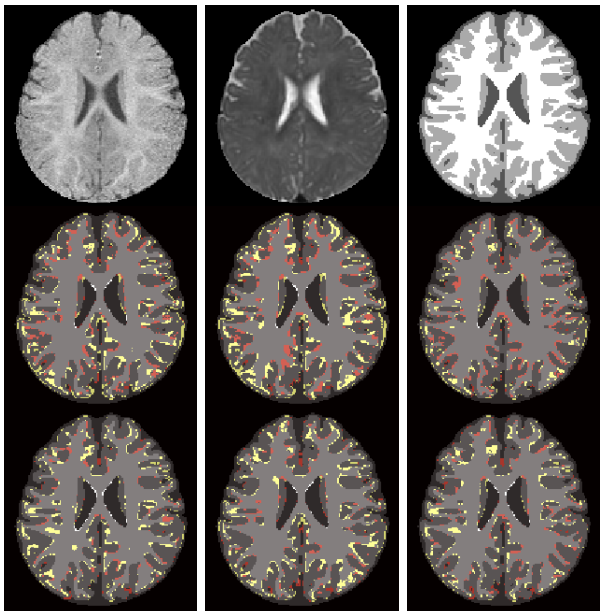


Fig. 6: Segmentation errors for the different methods on an iSeg2017 instance. The images in the first row correspond to T1-w, T2-w and ground truth. From left to right and top to bottom, segmentation errors of $\text{LM}_{2D}$, $\text{M}_{2D+2D+2D}$, $\text{LM}_{2.5D+2.5D}$, $\text{DM}_{3D}$, $\text{DM}_{2D+2D}$, $\text{DM}_{3D+3D}$. The error colour code is the following: yellow, red and white represent misclassified GM, WM and CSF, respectively.

that a much deeper architecture was used by Chen *et al.* [8] in the same challenge. However, two key aspects presumably helped in fully training that architecture: (i) the use of residual connections [11], and (ii) the data augmentation strategy. Additionally, as experienced by Zhang *et al.* [3], combining multiple modalities led to slightly higher performance than single modality approaches.

### *B. Cross-validation results on the iSeg2017 training set*

The leave-one-out cross-validation results obtained for iSeg2017 in terms of the DSC are displayed in Fig. 5. The networks performing the best when considering this similarity measure are Left $\text{Moeskops}_{2.5D+2.5D}$ and Dolz $\text{multi}_{3D+3D}$. Both of them using information from T1-w and T2-w volumes. Even though Dolz $\text{multi}_{3D+3D}$ obtained the highest scores in GM, Left $\text{Moeskops}_{2.5D+2.5D}$ achieved similar values for CSF and WM with lower IQR. Moreover, although Moeskops was expected to yield higher scores than Left $\text{Moeskops}_{2D}$, the case was completely the opposite. This particular situation could mean that the patch dimensions on the former architecture were optimised for working on specific datasets (image resolutions or quality) and not for a general case.

Unlike the evaluation using MRBrainS13 dataset, Dolz-based architectures achieved better performance than Moeskops-inspired ones. This could be a direct consequence of the number of instances (hence, the number of samples) to train the networks – nine and three training instances doing leave-one-out for iSeg2017 and MRBrainS13, respectively.

The use of multiple modalities extensions of an architecture allowed, in terms of DSC, to outperform single modality approaches. The difference is highly noticeable in GM and WM where Left $\text{Moeskops}_{2D+2D}$ obtained a DSC of $(0.86 \pm 0.1)$ and $(0.85 \pm 0.1)$, respectively, while the original Left $\text{Moeskops}_{2D}$ scored overlap was around $0.81$.

The segmentation errors of different algorithms on one slice are presented in Fig. 6. It can be seen that most of the errors are present in WM and GM – more specifically, around sulci and gyri. In accordance to the DSC values, the best response is between $\text{LM}_{2.5D+2.5D}$ and $\text{DM}_{3D+3D}$. While the former strategy achieves less GM misclassified regions than the latter one, the case is the opposite for WM.

### *C. Cross-validation results on the IBSR18 training set*

The proposed architectures were also tested on the well-known IBSR18 dataset through leave-one-out cross-validation. The different approaches were evaluated using DSC and also compared to unsupervised methods using the results reported by Valverde *et al.* [12]. The obtained DSC values are presented in Table I. In the table, the term corrected refers to the IBSR18 dataset in which sulcal CSF has been taken out of the evaluation [12]. It can be seen that the two proposed architectures were able to outperform all of the unsupervised methods in both the original and corrected volumes. In accordance to the expectations,

TABLE I: DSC obtained from leave-one-out cross-validation on IBSR18. The compared methods are listed in the first column. Then, the following six columns correspond to the DSC values for GM, WM and CSF on the original and corrected IBSR18 datasets. The results from FAST to PVC are taken from the research of Valverde *et al.* [12]. The last four rows correspond to the single-modality architectures achieving the best performances in the previous datasets. Left Moeskops$_{2.5D+spatial}$ and Dolz multi$_{3D+spatial}$ are abbreviated to LM and DM, respectively.

| Method | GM | | WM | | CSF | |
|---|---|---|---|---|---|---|
| | Original | Corrected | Original | Corrected | Original | Corrected |
| FAST | $0.74 \pm 0.04$ | $0.88 \pm 0.01$ | $0.89 \pm 0.02$ | $0.89 \pm 0.02$ | $0.12 \pm 0.05$ | $0.47 \pm 0.18$ |
| SPM5 | $0.68 \pm 0.07$ | $0.89 \pm 0.02$ | $0.86 \pm 0.02$ | $0.87 \pm 0.02$ | $0.10 \pm 0.05$ | $0.79 \pm 0.08$ |
| SPM8 | $0.81 \pm 0.07$ | $0.91 \pm 0.01$ | $0.88 \pm 0.02$ | $0.88 \pm 0.01$ | $0.17 \pm 0.08$ | $0.77 \pm 0.08$ |
| FANTASM | $0.71 \pm 0.03$ | $0.88 \pm 0.02$ | $0.88 \pm 0.02$ | $0.88 \pm 0.03$ | $0.11 \pm 0.07$ | $0.53 \pm 0.15$ |
| PVC | $0.70 \pm 0.08$ | $0.83 \pm 0.08$ | $0.83 \pm 0.07$ | $0.84 \pm 0.07$ | $0.13 \pm 0.06$ | $0.52 \pm 0.15$ |
| LM | $0.90 \pm 0.01$ | $\mathbf{0.95 \pm 0.01}$ | $\mathbf{0.92 \pm 0.02}$ | $\mathbf{0.91 \pm 0.02}$ | $0.77 \pm 0.05$ | $\mathbf{0.83 \pm 0.05}$ |
| DM | $\mathbf{0.94 \pm 0.01}$ | $0.92 \pm 0.02$ | $\mathbf{0.92 \pm 0.02}$ | $0.90 \pm 0.02$ | $\mathbf{0.83 \pm 0.03}$ | $0.81 \pm 0.02$ |

CSF was particularly difficult to segment compared to the other two classes. Furthermore, it can be observed that Left Moeskops$_{2.5D+spatial}$ was able to obtain higher scores in the three corrected volumes and one of the original volumes. This outcome is a consequence of the background class that is considered in DM$_{3D+spatial}$. Since the output classified patches for DM$_{3D+spatial}$ are of size $9 \times 9 \times 9$, it is complicated to restrict the samples only to brain areas. Thus, we included the background class on this classifier but the effect in corrected sequences was that the outermost parts of the brain (cortical GM and sulcal CSF) were misclassified.

## V. Discussion

In this paper, we proposed and evaluated 33 different architectures based on advantages and disadvantages from state-of-the-art methods. The improvement opportunities were located around five aspects: the patch dimension, number of sources of information, number of parameters, usage of implicit and explicit contextual information and resources limitations.

We considered well-known datasets from healthy and patient data, infant and adults, with different image qualities. In this sense, we were able to observe how robust the approaches were to data variations. As one of the keys of this research is to assess how versatile the approaches are among datasets, we did not tweak in any sense the algorithms for each specific scenario. According to the leave-one-out cross-validation experiments carried out on the considered datasets, we observed that approaches (i) with implicit – obtained in 2.5D and 3.D architectures – or explicit contextual information – through spatial and spectral coordinates – led to the greatest performances since this location information helps in discriminating one tissue from another, (ii) combining multiple modalities obtained higher performance than single-modality ones possibly due to the contrast information that each of the volumes carries. Additionally, it was observed that equipping 2.5D and 3D architectures with explicit contextual information helped them to achieve a better performance. Presumably, since triplanar and 3D networks acquire some implicit context from the content of the patches, but the patches depict only local instances; while the spatial and spectral coordinates introduce a sense of within-brain positioning.

Although a large number of experiments have been carried during this work, optimising the best parameters for each dataset can be time consuming. In the near future, our goal is to continue improving both architectures specifically for the upcoming MICCAI Grand Challenge iSeg2017 that will take place in Quebec City, Quebec in September 2017. The variants of the two architectures could be driven in terms input patch size, the number of kernels per convolutional layer, the number of fully connected layers, kernel size and merging strategies.

## References


[1] M. Filippi, et al. MRI criteria for the diagnosis of multiple sclerosis: MAGNIMS consensus guidelines. *The Lancet Neurology*, 15(3):292 – 303, 2016.

[2] X. Lladó, et al. Automated detection of multiple sclerosis lesions in serial brain MRI. *Neuroradiology*, 54(8):787–807, 2012.

[3] W. Zhang, et al. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *Neuroimage*, 108:214 – 224, 2015.

[4] P. Moeskops, et al. Automatic segmentation of MR brain images with a convolutional neural network. *IEEE Transactions on Medical Imaging*, 35(5):1252–1261, May 2016.

[5] M. Lyksborg, et al. An ensemble of 2D convolutional neural networks for tumor segmentation. In *Scandinavian Conference on Image Analysis*, pages 201–211. Springer, 2015.

[6] J. Dolz, et al. 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *Neuroimage*, pages –, 2017.

[7] C. Wachinger, et al. DeepNAT: Deep convolutional neural network for segmenting neuroanatomy. *Neuroimage*, pages –, 2017.

[8] H. Chen, et al. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *Neuroimage*, pages –, 2017.

[9] F. Milletari, et al. Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound. *Computer Vision and Image Understanding*, 2017.

[10] O. Ronneberger, et al. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[11] K. He, et al. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] S. Valverde, et al. Comparison of 10 brain tissue segmentation methods using revisited IBSR annotations. *Journal of Magnetic Resonance Imaging*, 41(1):93–101, 2015.

# Bayesian uncertainty quantification in computational imaging

Gourab Ghosh Roy

***Abstract*— Recovering the true unobserved image from noisy incomplete measurements is a challenging task. One popular class of inverse imaging models is that of log-concave models with known parameters, where the log-posterior distribution has a continuously differentiable term. For such models convex optimisation techniques can achieve quite accurate estimates of the original image. However there is inherent uncertainty associated with these high-dimensional ill-posed inverse problems and it is essential to quantify this uncertainty. Building on existing manual approach of Bayesian highest posterior density credible region analysis, this work proposes a fast and coarse uncertainty quantification methodology where the uncertainty quantification task is addressed as a convex optimisation problem. The obtained results demonstrate that such a convex optimisation based approach allows for a better exploration of the highest posterior density region. Hence it can provide better assessment of uncertainty than the manual approach in these imaging problems.**

## I. INTRODUCTION

Image recovery is a very important research area with applications in several fields like medical imaging [1] and radio-interferometry [2]. Recovering the unobserved true image from incomplete and also noisy measurement data is a high-dimensional and ill-posed problem. Many approaches combine data observation and prior knowledge in a Bayesian statistical framework and use Bayesian inference techniques for image recovery [3]. A particular class of problems is the case when the posterior distribution is log-concave with known set of parameters. This allows the maximum-a-posteriori (MAP) estimate to be computed by using high dimensional convex optimisation algorithms [4].

An important aspect to consider in these large-scale inverse imaging problems is the uncertainty associated with the true image [5]. The recovered MAP solution in a particular imaging problem is used for subsequent decision-making, like for instance checking for the existence of a tumor in a MRI image or that of radio-emission in an astronomical image. In the absence of ground truth, it is important to assess the uncertainty associated with features or structures that exist in the recovered image, so that it in turn strengthens subsequent decision-making and reasoning abilities. Convex optimisation based techniques alone cannot support advanced analyses like uncertainty quantification. Proximal Markov chain Monte Carlo (MCMC) methods [6], [7] allow Bayesian analyses of these log-concave imaging models. Uncertainty quantification from Bayesian highest posterior density (or HPD) credible region analysis is done in [7]. But these MCMC based methods in general are quite time consuming.

An efficient computation of approximate HPD credible region using convex optimisation was proposed in [8]. Bayesian uncertainty quantification quantification with the approximate HPD region has also been performed. This is done by removing the structure of interest in the MAP estimate by segmentation-inpainting process, followed by manually checking if the modified image lies within or exits the high confidence level HPD region. In this work we will aim at better exploration of the credible region. The motivation is to investigate solutions in the HPD region that are similar to the image without the structure of interest. This exploration is achieved using the framework of convex optimisation, the same framework that is used to get the recovered MAP image in the first place. The objective is to have an elegant framework for Bayesian uncertainty analysis and to in turn enhance the scientific decision-making abilities.

## II. Bayesian uncertainty quantification

In this section we present tools from the Bayesian framework that allow for the assessment of uncertainty in inverse imaging problems. The class of problems of interest in this work is that of log-concave models with known model parameters, where the log-posterior distribution has a smooth or continuously differentiable term.

Inverse imaging problems involve recovering the unknown true image $x \in \mathbb{R}^n$ from observation $y \in \mathbb{R}^m$. Typically there is a measurement model mapping $x$ to $y$ as

$$y = \Phi x + w, \tag{1}$$

where $\Phi$ denotes the measurement operator and $w$ stands for the noise in the process.

Let $x$ have a prior distribution $p(x)$, and observation $y$ be related to $x$ by the likelihood $p(y|x)$. Using Bayes' theorem, the posterior distribution can be written as [9]

$$p(x|y) = \frac{p(y|x)p(x)}{\int_{\mathbb{R}^n} p(y|x)p(x)\mathrm{d}x},$$

which models the knowledge of image $x$ after observing $y$. Now for log-concave models,

Gourab Ghosh Roy is a VIBOT Master student working in the Biomedical and Astronomical Signal Processing (BASP) group at Institute of Sensors, Signals and Systems, Heriot-Watt University. gg23@hw.ac.uk

$$p(x|y) \propto \exp\left(-g(x)\right), \tag{2}$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is a convex function. In this work, we will be dealing with a particular case where $g$ is of the form

$$g(x) = g_2(x) + g_1(x), \tag{3}$$

where function $g_2$ corresponding to the likelihood term is convex, continuously differentiable and gradient Lipschitz and function $g_1$ corresponding to the prior term is a proper, convex and lower semi continuous function. In sparsity aware models [10] typical examples of $g_1$ and $g_2$ are as follows

$$g_1(x) = f_1(\Psi x), \quad g_2(x) = f_2(\Phi x), \tag{4}$$

where $\Psi$ denotes a sparsifying transform corresponding to the domain where the image has a sparse representation. Function $f_1$ is a norm function promoting sparsity and function $f_2$ is typically a distance function promoting data fidelity. In some cases there may be additional priors on the image $x$ like the positivity condition and $g_1$ can be of the form $g_1(x) = \iota_{\mathbb{R}^n_+}(x) + f_1(\Psi x)$, where the indicator function of a set $\mathcal{C}$ has two values 0 or $+\infty$ respectively based on whether the argument belongs to $\mathcal{C}$ or not.

To capture information about $x$, point estimates are used [9]. A commonly used estimate is the maximum-a-posteriori or MAP estimate obtained by

$$x_{\text{MAP}} = \underset{x\in\mathbb{R}^n}{\operatorname{argmax}}\, \pi(x) = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\, g(x), \tag{5}$$

which can be computed efficiently using convex optimisation techniques [11].

To assess the uncertainty that is inherent to these ill-posed inverse problems, it would be helpful to have posterior credibility sets which denote the region of the $\mathbb{R}^n$ space where most of the posterior probability mass of $x$ lies. In the Bayesian framework this is formalised in the definition of credible regions [9]. A posterior credible region with confidence level $(1-\alpha)$ is defined as a set $\mathcal{C}_\alpha$ if

$$\mathbb{P}\left[x \in \mathcal{C}_\alpha | y\right] = 1 - \alpha \tag{6}$$

where $\mathbb{P}\left[x \in \mathcal{C}_\alpha | y\right] = \int_{\mathcal{C}_\alpha} p(x|y)\mathrm{d}x$. The highest posterior density (or HPD) region is the one with minimum volume [9]

$$\mathcal{C}^\star_\alpha = \{x : g(x) \leq \eta_\alpha\} \tag{7}$$

with $\eta_\alpha \in \mathbb{R}$ such that $\int_{\mathcal{C}^\star_\alpha} p(x|y)\mathrm{d}x = 1 - \alpha$ is satisfied. This can be defined by a single scalar $\eta_\alpha$. Computing the HPD region requires solving very high-dimensional integrals of the form $\int_{\mathcal{C}^\star_\alpha} p(x|y)\mathrm{d}x$. These integrals can be approximated with high accuracy using Monte Carlo integration [12], [13], which is time-consuming. However there is also existing methodology to reduce the high computational cost associated with MCMC methods by approximating HPD regions.

## III. State of the art

### A. Approximating HPD region by convex optimisation

A conservative approximate confidence region $\tilde{\mathcal{C}}_\alpha$ that outer-bounds the true HPD region is proposed in [8] for log-concave models. For any $\alpha \in (4\exp(-n/3), 1)$

$$\tilde{\mathcal{C}}_\alpha = \{x : g(x) \leq \tilde{\eta}_\alpha\}, \quad \tilde{\eta}_\alpha = g(x_{MAP}) + n(\tau_\alpha + 1), \tag{8}$$

where positive constant $\tau_\alpha = \sqrt{\frac{16\log(3/\alpha)}{n}}$.

The true threshold $\eta_\alpha$ in Equation 7 can be approximated by this surrogate threshold $\tilde{\eta}_\alpha$, when $x_{\text{MAP}}$ is available from MAP estimation using convex optimisation methods.

### B. Manual HPD region analysis

Approximating the Bayesian HPD region, [8] also performs uncertainty quantification using HPD region analysis. To quantify the uncertainty of a particular structure of interest in the recovered MAP image, the idea is to remove and replace the structure by a segmentation-inpainting process and then manually check if the new image without that structure still lies within or exits a high confidence level HPD region. If the modified image still lies within the HPD region that is the set of likely solutions to the inverse problem, it is concluded that the uncertainty around that structure is high, and it should not be used for subsequent scientific decision-making. If the new image without the structure does not lie within the defined HPD region anymore, it can be similarly concluded that the uncertainty around the structure is low, and it corresponds to an important feature in the image with high probability.

## IV. Proposed methodology for uncertainty quantification

The proposed approach aims at uncertainty quantification using the Bayesian statistical approach of credible region analysis. Here the motivation is better exploration of the HPD credible region, with the goal of achieving better assessment of uncertainty. Instead of manually checking if the MAP image after inpainting process without the structure of interest still resides in or goes outside the set of likely solutions, the idea is to explore images which lie in the HPD region and are similar to the inpainting output. Such exploration of the HPD region is done using convex optimisation techniques. Here the HPD region for a high confidence level of 0.99 is defined using the efficient approximation methodology outlined in [8] (Subsection III-A). The output of the inpainting (if outside the HPD region) is projected onto this region, which is a convex optimisation task.

On obtaining the projected image it is checked whether the structure exists, that is whether it stands out clearly enough from its immediate background, both by visual inspection and in terms of the energy associated with that structure. In practical scenarios such a decision will be taken by a field expert to confirm or refute the existence of the structure in the HPD region projected image. If the structure of interest does not exist anymore in the projected image, it is concluded that the uncertainty around that structure is high. As such, it is not suitable for subsequent analyses that is based on the existence of that structure. However if the structure still exists in the projected image, it is concluded that the uncertainty for that structure is low and there is sufficient evidence in favor of its existence. With high probability, it is an important feature in the image and can be used for subsequent decision-making and scientific reasoning.

## V. Convex optimisation

The projection of the output of inpainting $x_{\text{inp}}$ onto the 99% HPD region is computed by solving the following optimisation problem

$$x_{\text{proj}} = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\, \iota_{g(x)\leq\tilde{\eta}_{0.01}}(x) + \frac{1}{2}\|x - x_{\text{inp}}\|_2^2. \quad (9)$$

The first term in the optimisation problem dictates the computation of projection onto the HPD region. Typically with $g = g_2 + g_1$, the computation of projection onto the set $g(x) \leq \tilde{\eta}_{0.01}$ does not have a closed form. So block-wise decomposition and auxiliary variables are introduced as in the epigraphical projections computation work [14]. For solving this task, the optimisation problem is rewritten in the following equivalent form

$$\begin{aligned}\min_{x,\epsilon_1,\epsilon_2} \frac{\beta}{2}\|x - x_{\text{inp}}\|_2^2 + \iota_{\epsilon_{11}+\cdots+\epsilon_{1L}+\epsilon_2\leq\tilde{\eta}_{0.01}}(\epsilon_1,\epsilon_2) \\ +\iota_{\forall l, f_{1l}((\Psi x)_l)\leq\epsilon_{1l}}(\Psi x,\epsilon_1) + \iota_{f_2(\Phi x)\leq\epsilon_2}(\Phi x,\epsilon_2).\end{aligned} \quad (10)$$

As $f_1$ is typically a $\ell_1$ or $\ell_{2,1}$ norm function, further block-wise decomposition is required, hence $\epsilon_1$ is a vector with elements represented as $\epsilon_{1l}$. Though scaling of $\beta$ is introduced, the optimisation problem is still the same because the other functions are indicator functions which can only have values 0 or $\infty$. In the case of an additional positivity constraint on the image as mentioned before, the indicator function for positivity is taken outside of $g$ in the HPD region constraint and equivalently incorporated just as a separate additional term $\iota_{\mathbb{R}^n_+}(x)$ in Equation 10.

In this work, the primal-dual algorithm in [15], [16] is used. The algorithm applied to the HPD region projection task is outlined in detail in Algorithm 1. The `for` loop in lines 7−9 of in Algorithm 1 is executed in parallel. If there is an additional positivity constraint on the image $x$, the update of $x_i$ in the primal-dual algorithm (Line 5 in Algorithm 1) would have an additional max operation with respect to 0. Algorithm 1 requires the computation of some epigraphical projections, whose closed forms are provided in [14].

**Algorithm 1** Primal-dual Algorithm

1: $x_0 = x_{\text{MAP}}, \epsilon_{2,0} = f_2(\Phi x_0)$,
2: $\epsilon_{1,0} = (f_{11}((\Psi x_0)_1), \cdots, f_{1L}((\Psi x_0)_L))$
3: $y_{1,0} = \mathbf{0}, y_{2,0} = \mathbf{0}, v_{1,0} = \mathbf{0}, v_{2,0} = 0$
4: **for** $i = 1$ to $N$ **do**
5: $\quad x_i = x_{i-1} - \tau_1\beta(x_{i-1} - x_{\text{inp}}) - \tau_1(\Psi^* y_{1,i-1} + \Phi^* y_{2,i-1})$
6: $\quad (\epsilon_{1,i}, \epsilon_{2,i}) = P_{\epsilon_{11}+\cdots+\epsilon_{1L}+\epsilon_2\leq\tilde{\eta}_{0.01}}(\epsilon_{1,i-1} - \tau_2 v_{1,i-1}, \epsilon_{2,i-1} - \tau_2 v_{2,i-1})$
7: $\quad$ **for** $l = 1$ to $L$ **do**
8: $\quad\quad (y_{1l,i}, v_{1l,i})_l = (y_{1,i-1} + \sigma\Psi(2x_i - x_{i-1}), v_{1,i} + \sigma(2\epsilon_{1,i} - \epsilon_{1,i-1}))_l - \sigma P_{\text{epi}\, f_{1l}}((\frac{y_{1,i-1}}{\sigma} + \Psi(2x_i - x_{i-1}), \frac{v_{1,i-1}}{\sigma} + (2\epsilon_{1,i} - \epsilon_{1,i-1}))_l)$
9: $\quad$ **end for**
10: $\quad (y_{2,i}, v_{2,i}) = (y_{2,i-1} + \sigma\Phi(2x_i - x_{i-1}), v_{2,i} + \sigma(2\epsilon_{2,i} - \epsilon_{2,i-1})) - \sigma P_{\text{epi}\, f_2}(\frac{y_{2,i-1}}{\sigma} + \Phi(2x_i - x_{i-1}), \frac{v_{2,i-1}}{\sigma} + (2\epsilon_{2,i} - \epsilon_{2,i-1}))$
11: $\quad$ **if** $(i > 10)$ & $(g(x_i) \leq \tilde{\eta}_{0.01})$ **then**
12: $\quad\quad$ Exit for loop
13: $\quad$ **end if**
14: **end for**

An important aspect of the primal-dual minimisation algorithm is the selection of the proximal parameters $\tau_1$, $\tau_2$ and $\sigma$ and the weight $\beta$. The proximal parameters control the step sizes in the primal and dual domain. The weight parameter balances the minimisation of the difference with $x_{\text{inp}}$ image against satisfying all the involved constraints. In this approach, the values of $\beta$, $\tau_1$ and $\tau_2$ are set by the user (such that $\tau_1\beta < 1.9$), and to guarantee convergence the value of $\sigma$ is computed from [17]

$$\sigma = \frac{0.95 - \frac{\tau_1\beta}{2}}{\tau_1(||\Psi||^2 + ||\Phi||^2) + 2\tau_2}, \qquad \tau_1\beta < 1.9 \quad (11)$$

The optimisation problem at hand is found from the experiments to be complex and also sensitive to the choice of parameters. As such, finding the solution which satisfies all the constraints and also minimises the $\ell_2$ distance term in a practically feasible amount of time is quite difficult. So in this work a particular stopping criteria is incorporated. A good choice of the initial estimate of the solution $x_0$ is found to be the MAP image $x_{\text{MAP}}$, with the other primal and dual variables initialised as in Algorithm 1. With the selection of a large enough weight $\beta$ (satisfying Equation 11), the primal-dual algorithm update initially puts the solution closer to $x_{\text{inp}}$ but out of the HPD region. The algorithm waits for that to happen in at least 10 (user defined value) iterations. Once the updates in the primal-dual algorithm bring the solution $x_i$ back in the HPD region after $i$

iterations, the algorithm stops. At this point, not all the constraints involving the auxiliary variables in Equation 10 may be satisfied, only $g(x_i) \leq \tilde{\eta}_{0.01}$ has to hold. Now this is not the global optimum, but this estimate can be used to coarsely and efficiently assess the uncertainty in the imaging problem.

## VI. Simulations and results

In this section we present the results of applying our proposed uncertainty quantification approach to one radio-interferometric imaging problem [18]. Radio-interferometric (RI) imaging is used to observe radio emissions from a given area of the sky at great sensitivity and angular resolution. With the arrival of next generation radio telescopes RI images would be provided at unprecedented resolution and sensitivity. Array of antenna pairs that measure radio emissions produce visibilities or radio-interferometric data. The projected baseline components in units of wavelength of observation are denoted as $(u, v, w)$ where $w$ is the component in the line of sight whereas $(u, v)$ denotes the components in the orthogonal plane. The measurement model [18] relating visibility measurements $y$ to the intensity image $x$, with the measurement operator $\Phi$ that maps from the image space to the visibility space is given as

$$y = \Phi x + w, \quad \Phi = GFZ, \tag{12}$$

where $F$ is the discrete Fourier transform operator, the $n_0$-oversampling and scaling is accounted for by $Z$ to pre-compensate for interpolation imperfections, and $G$ allows for the computation of continuous Fourier samples from discrete Fourier coefficients in the nonuniform fast Fourier transform method proposed in [19]. The additive noise in the process is denoted by $w \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_m)$. Along with the positivity prior, the sparsity prior used is the SARA collection of wavelets [20] which is a concatenation of a Dirac basis with first eight Daubechies wavelet bases. The log-concave posterior distribution is given by

$$p(x|y) \propto \exp\left(-\frac{\|y - \Phi x\|_2^2}{2\sigma^2} - \iota_{\mathbb{R}_+^n}(x) - \lambda|\Psi x|_1\right) \tag{13}$$

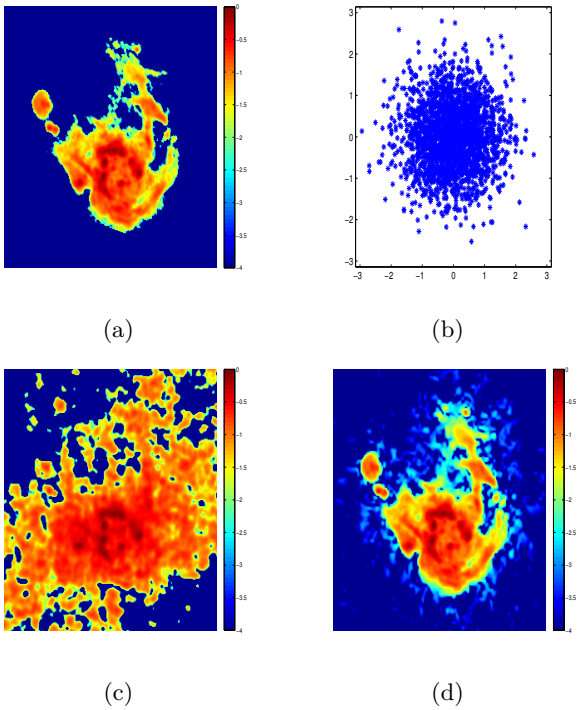
where $\lambda$ is the regularisation parameter.

There are several choices of optimisation algorithms that can be used to solve the MAP estimation problem like fast iterative shrinkage-thresholding algorithm (FISTA) [21], alternating direction method of multipliers (ADMM) [22], the primal-dual algorithm in [15], [16] etc. After obtaining the MAP estimate image, the structure of interest for uncertainty quantification is selected. That is represented in a binary mask $M$, with which particular pixels are chosen in the image. Based on the variational methods proposed in [23], the image interpolation or inpainting over the chosen structure is performed as

$$x_{\text{inp}} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}}\, \iota_{\mathbb{R}_+^n}(x) + \lambda|\Psi x|_1 + \iota_{M^c x = M^c x_{\text{MAP}}}(x), \tag{14}$$

where $M^c$ is the binary complement of the mask $M$. There are also many algorithms which can be used to solve this optimisation problem.

The test is with a $256 \times 256$ image of the HII region in the M31 galaxy (Figure 1a). A random $u - v$ coverage is generated using Gaussian sampling, with zero mean and variance of 0.25 of the maximum frequency. This concentration of visibility data for low frequencies makes the problem an ill-posed one. The oversampling is $n_0 = 4$. The ratio of the number of measurements to the number of pixels in the image is just 0.035. After generating Fourier measurements, noise is added to them to simulate real RI visibility data. The $\sigma$ of the additive noise used is $\frac{0.6}{\sqrt{2}}$. The regularisation parameter $\lambda$ assumed known here is $10^2$. The MAP estimate (Figure 1d) obtained by convex optimisation gives a reconstruction signal-to-noise ratio (RSNR) value of about 28.81 dB.



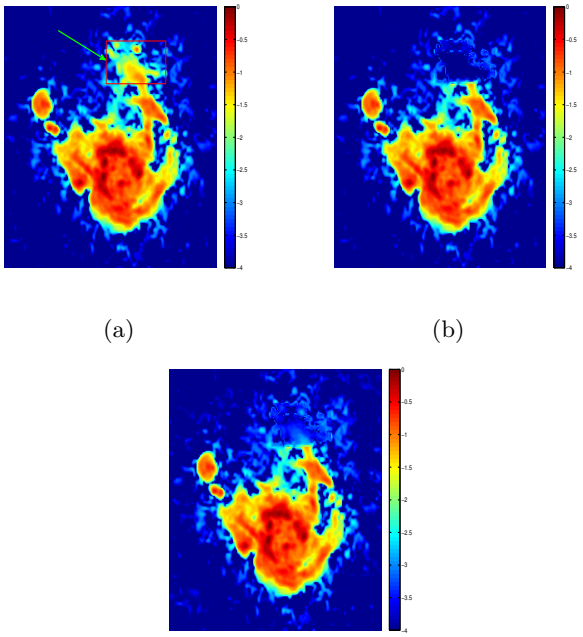
(a) (b) (c) (d)

Fig. 1: MAP Estimation. (a) True Image (Log scale). (b) Measurement locations in Fourier domain (frequency normalized between $[-\pi, \pi]$). (c) Image recovered from only measurements (Log scale). (d) Image recovered (MAP) with average sparsity and positivity prior (Log scale).

### A. Experiment 1 - High uncertainty

In Figure 2a, the structure of interest is denoted by an arrow in the MAP image. Any pixel with intensity value greater than $10^{-3}$ in the chosen rectangular region is part of the structure. In the output of the inpainting process (Figure 2b), the structure disappears. This image without the structure however lies outside the 99% HPD region and so the projection onto the HPD region is computed. The parameter values used are $\tau_1 = 1e-5$,

$\tau_2 = 1e2$ and $\beta = 2e4$. The run time is approximately 3 seconds. Looking at the HPD region projected image in Figure 2c, it can be said that the structure of interest does not exist. A small part of it does come back (better visualised in log scale) with projection, but the difference with its immediate background is not as pronounced as in the MAP image and also the structure energy value (0.0034) is similar to that (0.0027) in the output of inpainting. This implies that there is not enough evidence to support the existence of that structure. It may be concluded that this structure should not be used for subsequent scientific analysis because of high uncertainty around it at the current experimental settings which would include factors like number of measurements, noise level, etc. An important point to note is that the manual approach would assess the uncertainty to be low, so this method does achieve improved uncertainty quantification.



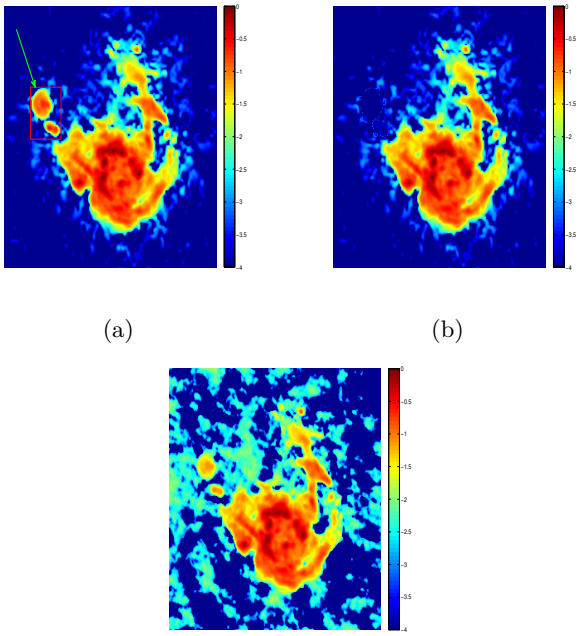
(a) (b)

(c)

Fig. 2: Experiment 1 - High Uncertainty. (a) MAP Estimate with structure 1 (Log scale). Structure Energy = 1.172. (b) Image after inpainting (Log scale). Structure Energy = 0.0027. (c) 99% HPD region projected image (Log scale). Structure Energy = 0.0034.

### *B. Experiment 2 - Low uncertainty*

In Figure 3a, the selected structure of interest is marked with an arrow in the MAP image. It is chosen with the same intensity threshold as in the previous experiment. The output of the inpainting process does not have that structure but the image lies outside the HPD region so its projection onto the HPD region is computed. The parameter values used are same as before and the run time is approximately 157 seconds.The structure of interest is observed to still exist in the HPD region projected image (Figure 3c). Projecting onto the HPD region brings the structure back. Visually the structure still is clearly discernible, and the energy value of 1.091 in the projected image compared to those in the MAP image (6.541) and the output of inpainting ($1.95e-05$) also shows that the structure exists in the projected image. Again this is open to interpretation and as mentioned previously in practical scenarios whether this still qualifies as a structure in the projected image should be judged by experts. Here the observation implies that the uncertainty around that structure is low under these experimental settings, and there is sufficient evidence in favor of its existence. With high probability it is an important feature in the image, as such it may be used for subsequent scientific analyses.



(a) (b)

(c)

Fig. 3: Experiment 2 - Low Uncertainty. (a) MAP Estimate with structure 2 (Log scale). Structure Energy = 6.541. (b) Image after inpainting (Log scale). Structure Energy = $1.95e-05$. (c) 99% HPD region projected image (Log scale). Structure Energy = 1.091.

## VII. Discussion

In this work aimed at coarse but fast uncertainty analysis, the HPD region is approximated using convex optimisation. For applications which are sensitive but also can allow for larger processing times, proximal MCMC methods can be used for more accurate computation of the HPD region. The idea of analysing the projection onto the defined HPD region would still apply. Now an important point to note is that for uncertainty analysis the HPD region needs to be defined for a high enough confidence level, so that the projected image with

or without the structure of interest still qualifies as a likely solution of the inverse problem. In all experiments the value of the confidence level $1-\alpha$ is taken to be 0.99. As outlined in [8], the posterior probability mass in a $\mathbb{R}^n$ space is highly concentrated around a $n-1$ dimensional shell, the result being the difference between $\eta_{0.01}$ and $\eta_{0.99}$ is very small. Another important aspect is that using a higher-dimensional credible region defined in terms of $x \in \mathbb{R}^n$ to explore properties of a specific region or structure in the image whose dimensionality is lesser than $n$, can lead to overestimation of uncertainty [8]. However the marginal posterior for such a defined region of interest in the image is typically computationally intractable. The projection onto the high-dimensional HPD region is a complex and challenging problem. For sensitive uncertainty quantification applications, the stopping criteria outlined in the proposed method cannot be used as we will need more accurate and precise estimates of the projection onto the HPD region. In this work not focusing on convergence to global optimum, proper selection of parameters and appropriate stopping criteria has aimed at fast but approximate uncertainty quantification, which can over or under estimate uncertainty.

## VIII. CONCLUSIONS AND FUTURE WORKS

In this work a fast and coarse uncertainty quantification approach for ill-posed high-dimensional inverse imaging problems is proposed. The imaging problems investigated are log-concave models with known parameters, where the log-posterior distribution has a smooth term. The proposed methodology addresses the highest posterior density credible region exploration as a convex optimisation task. The experiments on radioastronomical imaging problem demonstrate that this efficient approach better quantifies uncertainty than state of the art manual HPD region analysis.

Future work will involve quantification of uncertainty in more advanced imaging models. One such category of problems will have unknown model parameters, so there is uncertainty about them as well. A more complex class of inverse imaging problems would be models which are not log-concave. Possible approaches would be to use transformations to change the geometry of the space and make the problem a convex one, or to use results from compressibility theory.

## IX. ACKNOWLEDGMENTS

The author gratefully acknowledges the contributions of Prof. Yves Wiaux, Dr. Marcelo Pereyra and Dr. Audrey Repetti in this work. He also thanks everyone involved in the VIBOT program.

## References

[1] M. Lustig, D. Donoho, J. M. Pauly *et al.*, "The application of compressed sensing for rapid mr imaging," *Magn. Reson. Med*, vol. 58, no. 6, pp. 1182–1195, 2007.

[2] A. Ferrari, J. Deguignet, C. Ferrari, D. Mary, A. Schutz, and O. Smirnov, "Multi-frequency image reconstruction for radio interferometry. A regularized inverse problem approach," *ArXiv e-prints*, Apr. 2015.

[3] J. Kaipio and E. Somersalo, *Statistical and computational inverse problems*. Springer Science & Business Media, 2006, vol. 160.

[4] P. J. Green, K. Łatuszyński, M. Pereyra, and C. P. Robert, "Bayesian computation: a summary of the current state, and samples backwards and forwards," *Statistics and Computing*, vol. 25, no. 4, pp. 835–862, 2015.

[5] L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, L. Tenorio, B. van Bloemen Waanders, K. Willcox, and Y. Marzouk, *Large-scale inverse problems and quantification of uncertainty*. John Wiley & Sons, 2011, vol. 712.

[6] M. Pereyra, "Proximal markov chain monte carlo algorithms," *Statistics and Computing*, vol. 26, no. 4, pp. 745–760, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11222-015-9567-4

[7] A. Durmus, E. Moulines, and M. Pereyra, "Efficient bayesian computation by proximal markov chain monte carlo: when langevin meets moreau," *arXiv preprint arXiv:1612.07471*, 2016.

[8] M. Pereyra, "Maximum-a-posteriori estimation with bayesian confidence regions," *SIAM Journal on Imaging Sciences*, vol. 10, no. 1, pp. 285–302, 2017.

[9] C. Robert, *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[10] E. J. Candès *et al.*, "Compressive sampling," in *Proceedings of the international congress of mathematicians*, vol. 3. Madrid, Spain, 2006, pp. 1433–1452.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] C. Robert and G. Casella, "Monte carlo statistical methods springer," *New York*, 2004.

[13] M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. O. Hero, and S. McLaughlin, "A survey of stochastic simulation and optimization methods in signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 224–241, 2016.

[14] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "Epigraphical projection and proximal tools for solving constrained convex optimization problems: Part i," 2012.

[15] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013.

[16] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Advances in Computational Mathematics*, vol. 38, no. 3, pp. 667–681, 2013.

[17] J.-C. Pesquet and A. Repetti, "A class of randomized primal-dual algorithms for distributed optimization," *arXiv preprint arXiv:1406.6404*, 2014.

[18] A. Onose, R. E. Carrillo, A. Repetti, J. D. McEwen, J.-P. Thiran, J.-C. Pesquet, and Y. Wiaux, "Scalable splitting algorithms for big-data interferometric imaging in the ska era," *Monthly Notices of the Royal Astronomical Society*, vol. 462, no. 4, pp. 4314–4335, 2016.

[19] J. A. Fessler and B. P. Sutton, "Nonuniform fast fourier transforms using min-max interpolation," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 560–574, 2003.

[20] R. E. Carrillo, J. McEwen, and Y. Wiaux, "Sparsity averaging reweighted analysis (sara): a novel algorithm for radio-interferometric imaging," *Monthly Notices of the Royal Astronomical Society*, vol. 426, no. 2, pp. 1223–1234, 2012.

[21] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[23] T. F. Chan and J. Shen, "Variational image inpainting," DTIC Document, Tech. Rep., 2005.

# Automatic Seed Identification from CT-scan in Prostate Brachytherapy

Kibrom Berihu Girum[1], Alain Lalande[1], CREHANGE Gilles[1,2]
[1]Le2i, CNRS FRE-2005, Pole Imagerie Médicale et santé
Universite de Bourgogne Franche-Comté, Dijon, France
[2] Département de radiothérapie, CGFL, Dijon, France
kibrom2b@gmail.com

*Abstract*— **Post-implant dosimetry evaluation of permanent prostate brachytherapy must be performed for optimal curing of localized prostate cancer. The dosimetry analysis is often performed from computed tomography (CT) images. However, in post-implant CT examination of salvage prostate brachytherapy cumulated seeds appear from primary and salvage permanent brachytherapy. Indeed considerable amount of radioactive seeds are overlapped. In this paper, we presented a new method for post-implant dose distribution analysis of radioactive seeds implanted during salvage permanent prostate brachytherapy. The proposed method involves detection, separation and orientation estimation of radioactive seeds in 3D volume of CT images. Furthermore, the detected radioactive seeds in salvage permanent brachytherapy are identified based on the time of implantation using the two CT examinations acquired from primary and salvage post-implant. Dosimetry analysis is performed in salvage permanent prostate brachytherapy before and after the primary implanted seeds are removed. Computer vision techniques such as thresholding, connected component leveling, principal component analysis (PCA), geometry based connected component pattern analysis of closely spaced seeds in 3D and 2D images are implemented. Moreover, detected and separated radioactive seeds from primary and salvage permanent prostate brachytherapy are registered using manually selected landmarks. Five patient data set with 2 CT images each time were used to evaluate the proposed method. The proposed algorithm is evaluated both qualitatively and quantitatively and gave a promised result. The results of detection and separation compared with the number of seeds implanted demonstrate that our algorithm can achieve 99.4% and 97.4% in primary and salvage brachytherapy respectively. More importantly, dosimetry analysis of radioactive seeds implanted only during salvage permanent prostate brachytherapy is possible thanks to an initial manual based registration and an algorithm based on computer vision techniques.**

## I. INTRODUCTION

The burden of cancer is increasing throughout the globe as a result of population aging and growths as well as increasing adoption of cancer-causing lifestyle, particularly smoking and physical inactivity [1]. Prostate is one of the organs that has been affected in this outbreak. Recent studies show that it appears as 12 new cases and 23 deaths per 100,000 men worldwide per year from 2006-2010 [2] [3]. The most common and highly effective technique for curing such cancer is radiotherapy which involves exposition of the cancerous cells to ionizing radiation either through external beams or by implanting a permanent radioactive seeds (low-dose rate (LDR) brachytherapy). In order to cure the prostate cancer effectively the dose distribution (dosimetry) of implanted seeds should be optimal in such a way that it maximizes the dose on cancerous cells while minimizes on the healthy organs. The post-operative dose distribution (dosimetry) analysis is performed mostly using CT-based images [5]. If local relapse of prostate cancer is observed after performing primary brachytherapy, a salvage brachytherapy is required.

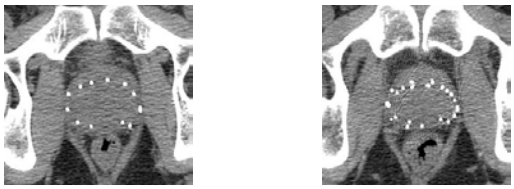
Salvage prostate LDR brachytherapy involves implantation of new radioactive seeds while the radioactive seeds implanted during the primary prostate brachytherapy are in place [6]. It is important to note that the radioactive seeds have a certain lifetime of radiation. Specifically, iodine-125 radioactive seeds are characterized by half-life time of about 60 days. Thus, the post-operative dosimetric analysis of salvage prostate brachytherapy should be performed only on the newly added radioactive seeds because there is no dose contributed (radiation) from the primary implanted seeds.

However, since the radioactive seeds implanted on both primary and salvage prostate brachytherapy are generally the same, it is impossible for radiologists to differentiate them in CT-based images acquired from the post-operative salvage brachytherapy. Thus, radiologists could use the CT examinations obtained from post-implant primary and salvage prostate LDR brachytherapy to compare and identify visually the position of primary implanted radioactive seeds. But, the position of radioactive seeds might not be the same on both CT examinations.Furthermore, two CT examinations in general are acquired with considerable interval of time as well as potentially difference in CT imaging setup. Figure 1 depicts radioactive seeds appearing from primary and salvage prostate brachytherapy.

Moreover, the prostate volume changes over time [6] in which the position and orientation of implanted radioactive seeds might change according to the momentum of prostate volume change. Pinkawa et al. [7] also presented that the radioactive seeds drift independently with considerable distance between the time of implantation and post-implant CT examination. Additionally, the radioactive seeds are also possibly to migrate to other tissues or drop out through urination [8] [9] [10]. Thus manual dosimetry analysis is almost impossible.

Theoretically, the seeds are supposed to align in the needle insertion direction, i.e. align with the CT axis. However,

Kibrom Berihu Girum is a student in the MSc Erasmus Mundus in VIsion and ROBotics (VIBOT). kibrom2b@gmail.com



(a) (b)

Fig. 1: Post-implant CT scans at the same position of a patient: (a) prostate CT scan from primary prostate LDR brachytherapy, (b) prostate CT scan from salvage prostate LDR brachytherapy.

practically considerable seeds are observed with different orientations and overlapping of closely spaced seeds. Indeed, the seed placement depends on many biomechanical factors in addition to the radiologist experience.

Thus seed detection, separation of overlapped seeds and orientation estimation have been studied for primary permanent prostate brachytherapy. Liu et al. [11] used binary threshold to segment the seeds then geometry-based connection search algorithm is applied in 2D to separate overlapped seeds. Template based separation of overlapped seeds is also developed by modeling the connection type of seeds in 2D [12]. Nguyen et al. [13] developed an automatic image processing solutions for the separation, localization and 3D orientation estimation of prostate seeds.

In this paper we presented a new algorithm for identification of radioactive seeds on post-implant CT-based images obtained from salvage permanent prostate brachytherapy. The main aim is to analysis the dose distribution contributed only by radioactive seeds implanted during salvage brachytherapy. It involves non-rigid registration of seeds detected and separated from the primary and salvage post-implant CT-images. The rest of this paper is organized as follows. Section II explains the developed algorithm and materials used. In Section III we presented and discussed the experimental obtained results. Conclusion and future work is summarized in section V.

## II. METHODOLOGY

### A. Experimental data

The data set used to investigate the performance of this developed algorithm consists of CT-based images obtained from post-implant primary and salvage permanent prostate brachytherapy. The data set used was obtained from CGFL radiotherapy department (Dijon, France) and Institute Curie (Paris, France). These data sets were acquired at different time intervals and potentially adapting imaging setup. For all patients iodine-125 were implanted. The iodine-125 emits photons up to 35kev and has a half-life of 59.46 days. The outer physical dimensions of the seed are $l = 4.5mm$ length and $r = 0.8mm$ external radius. The resolution of the CT examinations were different ( specifically, in $mm^3$ $0.4883 \times 0.4883 \times 2.5, 0.3906 \times 0.3906 \times 2.5, 0.5078 \times 0.5078 \times 2.5, 0.4375 \times 0.4375 \times 2.5$, and $1.2695 \times 1.2695 \times 1.25$). Moreover, the VariSeed planning software (Varian Medical Systems, Inc, USA) is used to calculate the dose distribution of radioactive seeds before and after the primary seeds are removed from the salvage prostate LDR brachytherapy CT-based images.

### B. Proposed method

The overall proposed algorithm is shown in figure 2 which is categorized into four main parts:

*1) System input:* The two CT examinations of a specific patient are taken as input, i.e. the primary and salvage permanent prostate brachytherapy post-implant CT-based images of a patient. 3D volume image is created from the 2D CT-based DICOM slices.

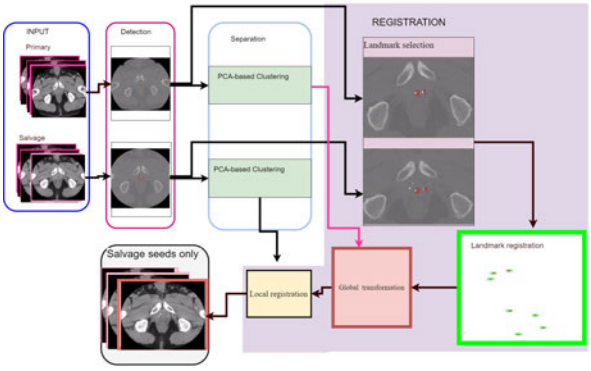



Fig. 2: The overall developed algorithm in block diagram.

*2) Detection of radioactive seeds:* The first and important step of registering the radioactive seeds between the two CT examinations is detecting the radioactive seeds. The radioactive seeds are metallic objects which possess a very high intensity level in CT-based images in addition to the pelvis. Thus, thresholding [19] is used to segment the radioactive seeds and pelvis from the CT volume. Furthermore, the pelvis and radioactive seeds are separated using connected component leveling [20], where the pelvis possess at much higher volume than the radioactive seeds. The proposed methodology is shown in figure 3. Numerous studies [13]-

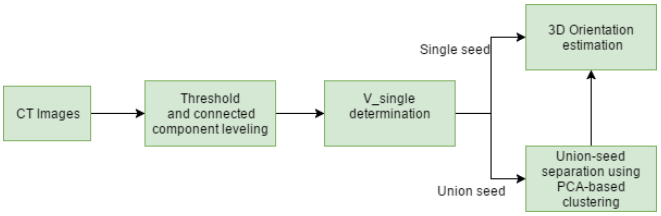



Fig. 3: Block diagram of seed detection, separation and orientation estimation. $V_{single}$: volume of single seed

[15] have been proposed that volume of single seed can be determined by computing the mean plus three times the standard deviation of all detected seeds' volume. These methods were with the assumption of the number of single seeds is greater than the number of overlapped seeds and were performed either in primary CT images or phantom

data set. However, the number of overlapped seeds could be greater than the number of single seeds, particularly in salvage brachytherapy.

Therefore, we proposed a new methodology for determining the volume of single seed and the size of single seed by exploiting the shape characteristics of the radioactive seeds in addition to the statistical method. The radioactive seeds used for low-dose radioactive seeds are generally cylindrical shapes [16]. Thus, the volume of a single radioactive seed is $V = \pi r^2 h$, where V the volume of the cylinder, r the radius and h the length of the cylinder. If we consider slice by slice the size of a seed would be $A = \pi r^2$, where A is area of the base of a cylinder. Therefore, volume of a single seed, $V_{single}$, is determined by comparing the number of voxels obtained for each detected seed and corresponding calculated volume of the seed. Then the radioactive seed with minimum error between the detected volume and calculated volume using PCA is used as $V_{single}$. This $V_{single}$ is used to partition detected overlapped (union) seeds. The detail of computing $V_{single}$ is explained in algorithm 1.

**Algorithm 1** Seed detection and classification, $V_{single}$ determination

1: Input DICOM series
2: Normalize and threshold the DICOM series
3: Connected component leveling
4: Remove largest volume connected component (pelvis)
5: Store number of voxels detected for each seed on vector $V_{detected}$
6: Compute volume of each connected component according to cylinder volume formula and store on vector $V_{pca}$
7: $error = V_{pca} - V_{detected}$
8: Compute standard deviation $S$ of all detected seeds' volume
9: Select $V_{selected}$ from the voxels which scores minimum error and occurs more than five times
10: **while** not classified all detected seeds **do**
11: **if** $V_{detected} > V_{single} + 3 \times S$
$V_{detected}$ is $Union$ seed
**else if** $V_{detected} < V_{single} - 3 \times S$
$V_{detected}$ is $noise$
**else**
$V_{detected}$ is $single$ seed
**end if**
12: **end while**

*3) Separation of overlapped seeds:* After the determination of the volume of a single seed is determined, the next step is to partition the overlapped seeds into single components. The block diagram for partitioning of overlapped seeds is depicted in figure 4. Then closely spaced (union) seeds

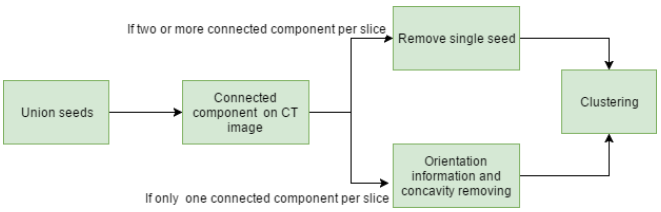


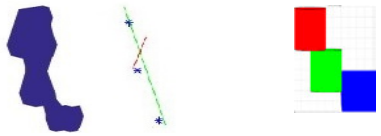

Fig. 4: Block diagram for partitioning of overlapped seeds.

are separated by combining the geometrical information of radioactive seed, and connected component pattern analysis of union seeds in 2D and 3D. Further the overlapped seeds are detected in 2D, once connected component leveling is performed in 3D. If the detected union seed appears more than once in any of the 2D images, the part of the seed which appears as one in any of the two 2D images is the point of connection and hence removed form the whole volume of the seed. This algorithm was proposed after we observed the connection types of overlapped seeds over 16 different CT examinations. Most of connection types are characterized in such pattern analysis except for some connection types, such as parallel connected seeds, T shaped connection. Thus for such connection types PCA is used to compute the longest and shortest axis of the connected component. First the number of connected seeds $k$ is determined as $k = \frac{V_{unionseed}}{V_{single}}$, where $V_{unionseed}$ is the volume of detected union seeds to separate. Then, $k$ centers are computed along the longest axis obtained using PCA [23]. The PCA is used to reduce the 3D coordinate of union seed into two component where the first component is the height. The second component is radius. The concave region of the union seed is detected and removed [21]. Then, the $k$ computed centers are given to k-means clustering to cluster [21] into $k$ components. The concavity detection is performed in 3D at the boundaries of the detected union seeds as shown in equation 1.

$$f_c(P(i,j,m)) = \sum_{x=i-W/2}^{i+W/2} \sum_{y=j-W/2}^{j+W/2} \sum_{z=m-W/2}^{i+W/2} BI \cap M(x,y,z), \tag{1}$$

where $BI$ is the union seed, and $P(i,j,m)$ is a point runs on the contour of the seed and $M$ is a $W \times W \times W$ mask centered on $P(i,j,m)$ [21]. Then the concaveness value of a point is $F_c(P(i,j,m))$ is the number of voxels of $M$ that intersects the background of $BI$. Hence, these voxels are removed before clustering, which enhances the clustering algorithm.

After the all overlapped radioactive seeds are clustered into single components algorithm 1 starting form step 5 is performed for all seeds. Figure 5 shows example of separating three overlapped seeds.



(a) (b)

Fig. 5: Detection of overlapped seeds and separation: (a) Detected union seeds, computed principal axis and centers using PCA, (b) Approximated seeds in cylindrical form after partitioning.

*4) Registration:* Classification of the detected and separated radioactive seeds in the CT-based images obtained from salvage permanent prostate brachytherapy into primary and salvage seeds is performed. To do so, we used manually selected corresponding landmarks between the two examination of a specific patient to initialize the registration. Let say the $P'$ and $P$ are the selected voxels from salvage and primary detected seeds respectively. The transformation matrix $M$ is computed so that $P' - MP \approx 0$. This computation is shown in equation 2.

$$\sum_{j=1,\ldots,N} (x'_j - \frac{M_{11}x_j + M_{12}y_j + M_{13}z_j + M_{14}}{M_{41}x_j + M_{42}y_j + M_{43}z_j + 1})^2 + (y'_j - \frac{M_{21}x_j + M_{22}y_j + M_{23}z_j + M_{24}}{h_{41}x_j + M_{42}y_j + M_{43}z_j + 1})^2 + (z'_j - \frac{M_{31}x_j + M_{32}y_j + M_{33}z_j + M_{34}}{M_{41}x_j + M_{42}y_j + M_{43}z_j + 1})^2, \quad (2)$$

where $N$ is the number of correspondence voxels, and $P'$ is composed of $(x'_j, y'_j, z'_j)$ approximated coordinates for $(x_j, y_j, z_j)$ of point $P$. Finally, this minimization problem is considered as least squares problem and Levenberg-Marquardt algorithm [18] is used. The transformation matrix $M$ is arranged as in equation 3.

$$M = \begin{pmatrix} M_{11} & M_{12} & M_{13} & tx \\ M_{21} & M_{22} & M_{23} & ty \\ M_{31} & M_{32} & M_{33} & tz \\ M_{41} & M_{42} & M_{43} & 1 \end{pmatrix} \quad (3)$$

This global transformation matrix is used to transform the seeds detected from primary space into the salvage space. However, since the radioactive seeds' movement is independent [7] [9] this global transformation model could not model the whole transformation of individual seeds. Thus, local 3D shape rigid registration [22] is necessary to refine the global non-rigid registration. To reduce the computation of 3D rigid registration between the primary and salvage detected seeds, only seeds within a $7 \times 7 \times 7$ cube (in pixels) are considered as a candidate seeds to register. Then a $n \times m$ association matrix is created, where the $n$ and $m$ are the number of detected seeds in primary and salvage CT images respectively. The association matrix is then filled with the distance and orientation difference obtained between registered primary and salvage seeds from the 3D shape rigid registration transformation matrix. The matrix is then normalized to have a probability of association. Finally, the seed identification is performed with the highest probability between corresponding seeds.

The registered seeds are removed from the salvage 3D volume of images by replacing their intensity with the background intensity values. Moreover, the dosimetry analysis in CT-based salvage images before and after the primary radioactive seeds are removed is performed using VariSeed planning software.

## III. EXPERIMENTAL RESULTS

Here, we present the experimental results obtained for ten different CT-examination of five patients.

### A. Seed detection and separation

Table I shows the detection and separation result for primary post-implant CT images. Note that number of implanted seeds were recorded during implantation. Also detecting of radioactive seeds in salvage brachytherapy is done for the corresponding patients,i.e. $p0, p01, p02, p03, p04, p05$. Note that the total number of implanted seed in salvage CT images is the sum of radioactive seeds implanted during primary and salvage brachytherapy. Table II depicts the result obtained for salvage CT images. From these two tables I and II we can observe that the probability of overlapping of radioactive seeds is high in salvage permanent prostate brachytherapy. When the number of detected is greater than the implanted seeds this could be

| Patient | No. implanted | No. detected | No. after separated | No. union |
|---|---|---|---|---|
| p01 | 64 | 52 | 64 | 12 |
| p02 | 65 | 56 | 65 | 9 |
| p03 | 61 | 44 | 60 | 16 |
| p04 | 70 | 49 | 71 | 22 |
| p05 | 50 | 34 | 49 | 15 |

TABLE I: Seed detection and separation results for primary brachytherapy. No.: "number of seeds ".

| Patient | No. implanted | Total | No. detected | No. after separated | No. union |
|---|---|---|---|---|---|
| p01 | 51 | 115 | 69 | 113 | 44 |
| p02 | 45 | 110 | 63 | 107 | 44 |
| p03 | 18 | 79 | 50 | 80 | 30 |
| p04 | 23 | 93 | 59 | 90 | 31 |
| p05 | 26 | 76 | 47 | 72 | 25 |

TABLE II: Seed detection and separation results for salvage brachytherapy. No.: "number of seeds ".

due to calcifications. The accuracy of detection is 99.4 % and 97.4% in primary and salvage brachytherapy respectively.

### *B. 3D seed orientation estimation*

The estimated 3D orientation of detected radioactive seeds is shown in figure 6 for patient one from primary and salvage brachytherapy respectively. The comparison is also made with the VariSeed planning software, where it assumes radioactive seeds are aligned in the direction of CT-axis. The size of seeds is not equal, especially in salvage brachytherapy 8c , because there is overlapping of closely spaced seeds.

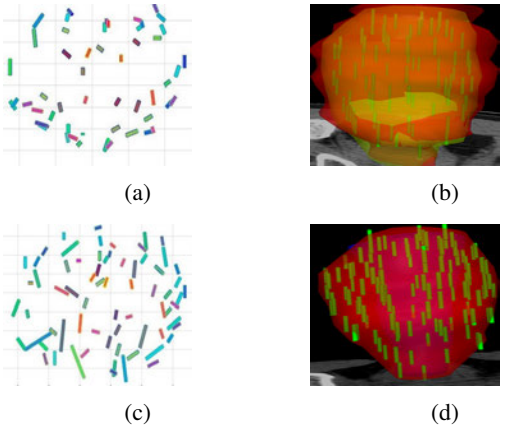


Fig. 6: Orientation estimation of detected seed. Estimated 3D orientation of seeds obtained from, (a) primary brachytherapy, (c) salvage brachytherapy using the proposed methodology. Estimated 3D orientation of seeds obtained from, (b) primary brachytherapy, (d) salvage brachytherapy using VariSeed planning software.

### *C. Seed identification*

The primary implanted radioactive seeds are identified and removed from the salvage prostate LDR brachytherapy post-implant CT-based images. Figure 7 displays seed identification and removing of primary seeds from patient one.

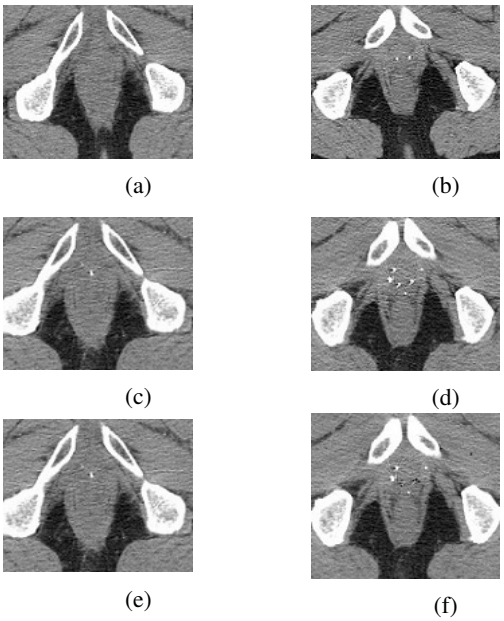


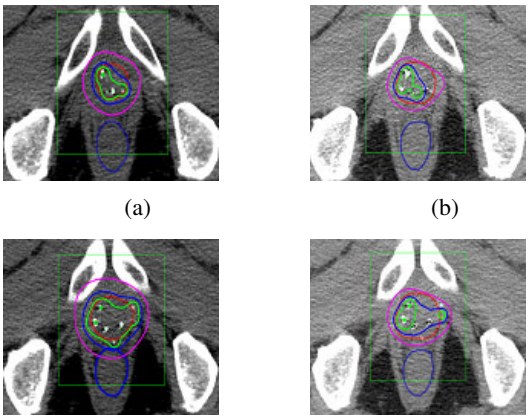
Fig. 7: Seed identification. CT scan acquired from: Primary brachytherpy, (a) and (b); Salvage brachytherapy, (c) and (d). The CT images of the salvage brachytherapy after the primary seeds are removed, (e) and (f). Note that the correspondence is column wise. The associated color and its dose percentage is: Magenta at dose=240 Gy; Blue at dose=180 Gy; Green at dose=120 Gy.

### *D. Dosimetry analysis*

The experimental results for dose distribution analysis before and after the primary radioactive seeds are removed is depicted in figure 8. As clearly indicated in figure 8, the dose distribution is quite different before and after the primary radioactive seeds are removed. This dos distribution was obtained with the VariSeed planning software.

## IV. Conclusion

In this project, the automatic detection, separation and orientation estimation of iodine-125 radioactive seeds in 3D CT images of both primary and salvage LDR prostate brachytherapy is implemented. Moreover, radioactive seeds identification on CT-based images obtained from salvage permanent prostate brachytherapy is implemented with an ergonomic graphical user interface to handle all necessary steps. More importantly, dosimetry analysis of radioactive seeds implanted during salvage prostate brachytherapy is possible thanks to the computer vision techniques. Post-implant evaluation of permanent prostate brachytherapy is



(a) (b) (c) (d)

Fig. 8: Dosimetry analysis for CT images of salvage prostate brachytherapy shown at the same position: (a) and (c) dose distribution before the primary seeds are removed, (b) and (d) dose distribution after the primary seeds are removed.

extremely important to assess permanent prostate brachytherapy procedures by computing the real dose distribution to the prostate cancer and other organs instead of assuming the alignment of seeds along the theoretical insertion direction of the brachytherapy needles.

As a future work, it is recommended to analyses the potential difference of computed tomography imaging setup. A complete automatic identification of radioactive seeds could be developed using a non-rigid registration methods considering the whole CT images globally followed by a local non-rigid registration of the radioactive seeds. Most importantly a dose-volume histogram should be developed for both at individual and cumulated level of detected radioactive seeds. Finally, using multi-modal imaging, adding of the tumor area on the 3D modeling of the pelvis and the seeds using information from PET or MRI imaging would be helpful for better analysis of the tumor and future outcomes.

## REFERENCES


[1] A. Jemal, F. Bray, M. M. Center, J. Ferlay, E. Ward, and D. Forman. Global cancer statistics. CA: a cancer journal for clinicians, 61(2):6990, 2011.

[2] CR-UK, cancer stats incidence UK. Available from: www.cancerresearchuk.org and[cited May 30 2017]. Cancer research uk 2011, 2014.

[3] M. Horner, L. Ries, M. Krapcho, N. Neyman, R. Aminou, N. Howlader, S. Altekruse, E. Feuer, L. Huang, A. Mariotto, et al. Seer cancer statistics review, 1975-2006, national cancer institute. bethesda, md, 2009.

[4] G. Créhange, I. Hsu, A. J. Chang, and M. Roach III. Salvage prostate brachytherapy for postradiation local failure. In Management of Prostate Cancer, pages 287302. Springer, 2017.

[5] S. Nag, W. Bice, K. DeWyngaert, B. Prestidge, R. Stock, and Y. Yu. The american brachytherapy society recommendations for permanent prostate brachytherapy postimplant dosimetric analysis. International Journal of Radiation Oncology Biology Physics, 46(1):221230, 2000.

[6] G. Créhange, D. Krishnamurthy, J. A. Cunha, B. Pickett, J. Kurhanewicz, I. Hsu, A. R Gottschalk, K. Shinohara, M. Roach, and J. Pouliot. Cold spot mapping inferred from mri at time of failure predicts biopsy-proven local failure after permanent seed brachytherapy in prostate cancer patients: Implications for focal salvage brachytherapy. Radiotherapy and Oncology, 109(2):246250, 2013.

[7] M. Pinkawa, B. Asadpour, B. Gagel, M. D. Piroth, H. Borchers, G. Jakse, and M. J. Eble. Evaluation of source displacement and dosevolume changes after permanent prostate brachytherapy with stranded seeds. Radiotherapy and Oncology, 84(2):190196, 2007

[8] S. Nag, D. D. Scaperoth, R. Badalament, S. A. Hall, and J. Burgers. Transperineal palladium 103 prostate brachytherapy: analysis of morbidity and seed migration. Urology,45(1):8792, 1995.

[9] A. D Steinfeld, B. R. Donahue, and L. Plaine. Pulmonary embolization of iodine-125 seeds following prostate implantation. Urology, 37(2):149150, 1991.

[10] E. M. Tapen, J. C. Blasko, P. D. Grimm, H. Ragde, R. Luse, S. Clifford, J. Sylvester, and T. W. Griffin. Reduction of radioactive seed embolization to the lung following prostate brachytherapy. International Journal of Radiation Oncology* Biology* Physics, 42(5):10631067, 1998

[11] H. Liu, G. Cheng, Y. Yu, R. Brasacchio, D. Rubens, J. Strang, L. Liao, and E. Messing. Automatic localization of implanted seeds from post-implant ct images. Physics in medicine and biology, 48(9):1191, 2003.

[12] M. Yazdi, S. GhadarGhadr, and L. Beaulieu. A template based approach for automatic seed detection in post-implant ct images for prostate brachytherapy. In Nuclear Science Symposium Conference Record, 2006. IEEE, volume 6, pages 32053208. IEEE, 2006.

[13] H. Nguyen, C. Fouard, F. Meneu, J. Giraud, and J. Troccaz. Automatic 3d seed location and orientation detection in ct image for prostate brachytherapy. In Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on, pages 13201323. IEEE, 2014.

[14] E. Holupka, P. Meskell, E. Burdette, and I. Kaplan. An automatic seed nder for brachytherapy CT postplans based on the hough transform. Medical physics, 31(9):26722679, 2004.

[15] Z. Lu and W. Chen. Fast and robust 3-d image registration algorithm based on principal component analysis. In Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on, pages 872875. IEEE, 2007.

[16] P. Karaiskos, P. Papagiannis, L. Sakelliou, G. Anagnostopoulos, and D. Baltas. Monte carlo dosimetry of the selectseed 125i interstitial brachytherapy seed. Medical physics, 28(8):17531760, 2001.

[17] J. H. Han. Detection of convex and concave discontinuous points in a plane curve. In Computer Vision, 1990. Proceedings, Third International Conference on, pages 7174.IEEE, 1990.

[18] T. K. Moon. The expectation-maximization algorithm. IEEE Signal processing maga-zine, 13(6):4760, 1996.

[19] R. M. Haralick, and L. G.Shapiro. Image segmentation techniques. Computer vision, graphics, and image processing, Elsevier, 29(1): 100(132, 1985).

[20] C. R. Gonzalez and E. R. Woods. Image processing. Digital image processing, 2, 2007.

[21] J. H. Han. Detection of convex and concave discontinuous points in a plane curve. In Computer Vision, 1990. Proceedings, Third International Conference on, pages 7174. IEEE, 1990.

[22] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. Robotics-DL. International Society for Optics and Photonics tentative,586606, 1992.

[23] B. T. Joshua, S. D. Vin, and C. L. John. A global geometric framework for nonlinear dimensionality reduction. science, 290(5500):23192323, 2000.