On patient see-through Augmented Reality based on visual SLAM

Nader Mahmoud Elshahat Elsayed Ali, Oscar Garcia Grasa, Stéphane Nicolau, Christophe Doignon, Luc Soler, Jacques Marescaux, Jose Maria Martinez Montiel

Noname manuscript No.
(will be inserted by the editor)

On-Patient See-through Augmented Reality based on Visual SLAM

Nader Mahmoud · Óscar G. Grasa · Stéphane A. Nicolau · Christophe Doignon · Luc Soler · Jacques Marescaux · J.M.M. Montiel

the date of receipt and acceptance should be inserted later

Abstract *Purpose* An augmented reality system to visualize a 3D pre-operative anatomic model on intra-operative patient is proposed. The hardware requirement is commercial tablet-PC equipped with a camera. Thus, no external tracking device nor artificial landmarks on the patient are required.
Methods We resort to Visual SLAM to provide markerless real-time tablet-PC camera location with respect to the patient. The pre-operative model is registered with respect to the patient through 4-6 anchor points. The anchors correspond to anatomical references selected on the tablet-PC screen at the beginning of the procedure.
Results Accurate and real-time pre-operative model alignment (approximately 5 mm mean FRE and TRE) was achieved, even when anchors were not visible in the current field of view. The system has been experimentally validated on human volunteers, in-vivo pigs and a phantom.
Conclusions The proposed system can be smoothly integrated into the surgical workflow because it: 1) operates in real-time, 2) requires minimal additional hardware only a tablet-PC with camera 3) is robust to occlusion, 4) requires minimal interaction from the medical staff.

Keywords Augmented Reality · Visual SLAM · Registration · Operating room · Surface meshes

Nader Mahmoud
IRCAD (Institut de Recherche contre les Cancers de l'Appareil Digestif), France
ICube (UMR 7357 CNRS), Université de Strasbourg, France
E-mail: nader-mahmoud.ali@etu.unistra.fr

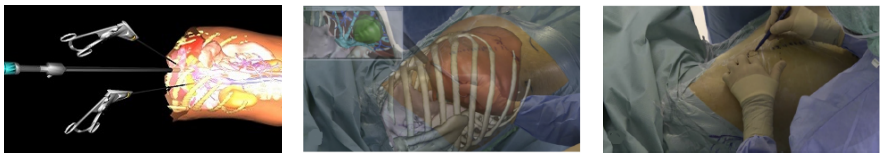
Óscar G. Grasa · Stéphane A. Nicolau · Luc Soler · Jacques Marescaux
IRCAD (Institut de Recherche contre les Cancers de l'Appareil Digestif), France

Christophe Doignon
ICube (UMR 7357 CNRS), Université de Strasbourg, France

J.M.M. Montiel
Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain
E-mail: josemari@unizar.es

1 Introduction

Patient pre-operative 3D model (P3DM) is readily available through various imaging modalities such as computed tomography (CT) or magnetic resonance (MRI). The P3DM is typically displayed on a desktop monitor, laptop or tablet-PC. However, the practitioner has to mentally project that information onto the patient. An augmented reality (AR) superimposition of the P3DM onto the patient can provide the practitioner with a kind of "X-ray vision", easing the information transfer from the P3DM to the actual patient for a surgical procedure. Such AR technology can overcome some of minimally invasive surgery (MIS) limitations such as trocar/instrument placement in thoracic surgery[1]. Indeed, the trocar placement can then be decided before the surgery on the P3DM (cf. Fig. 1[a]) and this location is superimposed intra-operatively on a static view provided by a fixed camera. This AR view allows fast, safe and optimal trocar set-up to provide the best surgical approach to reach the target (cf. Fig. 1[b-c]). However, this technique suffers from two important drawbacks. First, the P3DM registration is performed manually and needs to be recomputed after every change of the relative position of the camera with respect to the patient. Second, this kind of relative motion is difficult to avoid even using bulky fixing methods for both the camera and the patient.



(a) (b) (c)

Fig. 1: Port positioning with AR guidance during trans-thoracic minimally invasive hepatectomy [1]. (a) Pre-operative trocar placement planning. (b) and (c) Marking of the chosen port site.

The accuracy of the image augmentation with the P3DM depends on two factors: 3D camera tracking and P3DM registration. 3D camera tracking implies the computation, in real-time, of the 3D camera position with respect to the patient body. P3DM registration implies anchoring, also in real-time, of the P3DM to the patient body. In this paper, we propose a Visual Simultaneous localization And Mapping (VSLAM)-based approach for on-patient AR visualization, while the tablet is moved by the practitioner around the patient. Our system satisfies the following constraints: seamless integration in the operating room (OR), real-time performance, minimal interaction from the medical staff and robustness to occlusion and failure.

The reminder of the paper is organized as follows. Section 2 provides a review of the related work on on-patient AR visualization and VSLAM for camera tracking and mapping. Section 3 gives an overview of the proposed on-patient AR visualization system, followed by a detailed system description in Section 4. In section 5, the experimental results and evaluations are discussed. Finally, a conclusion summarizes our achievements and future work are presented in section 6.

2 Related work

2.1 On-Patient AR visualization

Various approaches for on-patient AR have been proposed in recent years, based on two different techniques to track 3D camera location: surface-based registration and 2D/3D point correspondences. In surface-based registration techniques [2–6], a tablet-PC is mounted with a range camera, RGB-D sensor or stereo-vision to continuously capture the depth and color information, from which the skin surface is automatically extracted. This surface is then registered with the P3DM acquired from CT images, typically ICP (Iterative Closest Point) is used. This process is repeated for every frame at 5-10Hz [4]. The major drawbacks of this type of techniques are the computation cost of depth image segmentation and ICP. Secondly, a good initialization for the ICP registration is required and has to be provided manually. Depending on the interface, it is not clear whether practitioners/surgeons can accept this task. Thirdly, this kind of methods are not robust to partial occlusions of the skin. To achieve real-time performance, either parallel processing [4] or client/server architecture [6] or both [2] are used: a powerful server PC is necessary to process data and the tablet-PC is used as a display tool only.

In 2D/3D point correspondences techniques [7–10], markers need to be visible in the CT image and can be either natural landmarks or artificial ones placed on the patient before scanning. Their 2D positions in each frame are used to solve a 2D/3D geometrical relationship. In this case, the obvious drawback is the use of markers which should be visible in CT/MRI. Moreover, a minimum number of markers must be visible in every frame to register the P3DM [7,10], which impedes the surgeon movements. Indeed, markers are likely to be occluded by surgeon hand or a surgical instruments. Moreover, current on-patient visualization techniques typically evaluate their accuracy by measuring the registration errors of skin fiducials [2,3,5,6], the discrepancy in pixels in the image [8,10], and/or the processing time [4]. To address these drawbacks, a VSLAM-based method for on-patient visualization is proposed. Our system is rigorously evaluated in terms of: processing time and robustness on human data, registration accuracy on pigs during both breath-hold and respiration phases using fiducials and registration accuracy on a liver phantom using fiducials.

2.2 VSLAM-based camera tracking

VSLAM is a popular topic in robotics and computer vision, as it aims at building a 3D map of an unknown environment while simultaneously tracking camera location. Davison [11] proposed the first real-time VSLAM, based on an Extended Kalman Filter (EKF). Further improvements over the EKF SLAM have been proposed [12,13]. EKF approaches initialize robustly, but have a poor scaling given its limitation of the map to a few hundreds points. EKF approaches have been successfully applied to MIS notably for abdominal surgery to track endoscope motion, to provide 3D scene structure [14] and to live AR annotations [15].

A significant leap with respect to EKF VSLAM was Klein and Murray's PTAM (Parallel Tracking and Mapping) [16]. PTAM algorithm performs in real-time all the steps of the classic photogrammetric 3D reconstruction [17]: matching, view

selection, initialization and non-linear optimization termed *Bundle Adjustment* (BA). Recently, ORB-SLAM [18] builds on PTAM and extends its performance to large scale mapping including loop closure, large scale relocation and position graph optimization. In this system, ORB image features play a key role in data association and relocation. ORB [19] is a newly developed image point descriptor that is able to handle rotation and scale changes with a comparable performance to SIFT [20] but with only a fraction of the computational cost. Our VSLAM combines some of these systems in order to balance good performance and reasonable hardware requirements in the medical environment. We use EKF for automatic initialization, PTAM approach for operation in room size environments where loop closure is not needed, and ORB features are used for relocation.

Our contributions are: 1) A VSLAM for on-patient AR visualization, which only requires a tablet-PC. 2) A usage strategy that fits the clinical constraints and is easy to setup and use inside the OR. 3) Interactions from the surgeon are reduced to the identification of 4 to 6 anatomical references at the beginning of the procedure. 4) The system is validated providing geometrical accuracy and computing cycle time.

3 System overview

The workflow of our system is shown in Fig. 2. It firstly consists of an offline stage. A CT volume of the patient is acquired and segmented to generate the P3DM. The P3DM is composed of surface meshes corresponding to the skin surface and to the selected body structure surfaces. The practitioner/surgeon selects at least 4 (typically between 4 and 6) anatomical landmarks ($L_i, i \in \{1..6\}$) on the skin mesh (called *anchor points*). The anchors should be easily identifiable on the skin of the patient during the procedure.

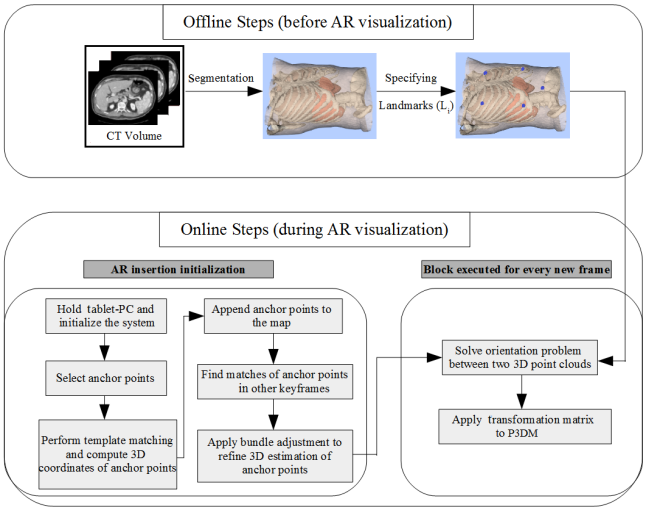


Fig. 2: Overall system workflow

In the OR, the practitioner directs the tablet-PC camera at the patient and performs a translational motion to bootstrap the VSLAM, then he/she identifies the anchor points (L_i) by clicking on the tablet-PC live video stream. Those anchor

points provide the 3D/3D registration of the P3DM within the VSLAM map. Afterwards, a synthetic image of the P3DM can be overlaid on live video stream. This interactive procedure of anchor points identification is required only once at the beginning of the procedure, then, the practitioner can move the tablet-PC around the patient and experience the AR visualization (cf. Sec. 4.3), even if none of the anchors remains visible in the tablet-PC camera field of view.

4 System description

4.1 VSLAM architecture

Our VSLAM aims at running on mobile devices and is intended for small scenes. It is a PTAM-like algorithm that automatically and sequentially computes a 3D photogrammetric reconstruction from the live video stream (cf. Fig. 3). To do so, a set of interest points has to be matched along the sequence; we use sparse features detected in the image by the Features from Accelerated Segment Test (FAST) detector [21]. In order to reduce number of outliers, we keep only the most salient features, whose Shi-Tomasi score [22] is over 100. The ORB descriptor [19] is then used to describe the detected features (block A in Fig. 3). A set of frames –named keyframes– has to be selected because the complexity scales cubically with the number of frames. Then a non-linear BA optimization yields 3D location for the map points and positions for the keyframes accurately. The BA is run in a thread termed *Mapping* described below. In parallel, another thread –termed *tracking*– estimates the position of the frames that are not keyframes.

Camera tracking This task operates sequentially on all frames of the live video (block C in Fig. 3). The 3D locations of the map points are assumed to be available. Then the position of each frame is computed by non-linear optimization of the reprojection error for the matched points. To avoid the influence of spurious matches, a two-stage optimization is applied. In the first stage the Huber influence function is used as it is less sensitive to outliers. While in the second stage the optimization is switched to the Tukey kernel to achieve a robust optimization. An initial guess is needed for the camera position and is computed from the camera position and velocity estimated for the previous frame.

The difference among the various VSLAM methods is how the matches between the map points and the current frame are computed. We estimate a region where the map points are expected to be found by reprojecting the 3D map points onto the predicted camera position. The ORB descriptor of each map point is compared with those of all the features detected inside the predicted region, using the ratio between closest to second-closest neighbors as a score [23]. If no matching is found, a correspondence is searched by patch correlation in the prediction region. If the number of matches is below 20 (empirically defined), the camera is assumed to be lost and the relocation process is started. Additionally, the tracking thread chooses a keyframe among the processed frames using the standard VSLAM criteria of minimal parallax distance with respect to all map keyframes. For each of the keyframes, we compute the median parallax with respect to the current frame. If the smallest parallax angle is over 2 degrees, the current frame becomes a new keyframe. This parallax angle threshold is chosen to be small to increase the number of keyframes, map points and to avoid rapid tracking loss.

The median parallax is estimated using the median of the XYZ coordinates of the map points detected in the current frame.

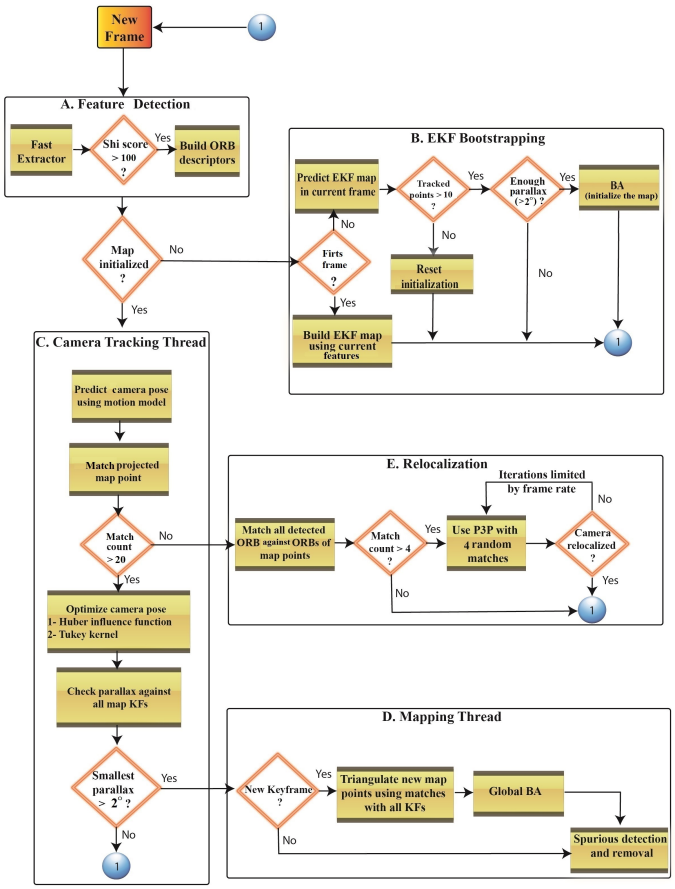


Fig. 3: VSLAM architecture.

Mapping The mapping thread runs in parallel with the tracking thread but at a lower frequency, continuously improving the map points estimation (block D in Fig. 3). The BA minimizes total reprojection error with respect to the keyframe positions $\mathbf{X}_{WC_i}$ and the 3D map point locations $\mathbf{X}_{Wj}$:

$$\underset{\mathbf{X}_{Wj},\mathbf{X}_{WC_i}}{\arg\min} \sum_{i,j} \rho\left(||\mathbf{u}_{ij} - \text{CamProj}(\mathbf{X}_{Wj}, \mathbf{X}_{WC_i})||^2\right) \tag{1}$$

where $\mathbf{u}_{ij}$ is the matched observation of the j-th map point by the i-th keyframe. The CamProj codes the projection function including perspective and radial distortion. ρ denotes the robust Huber influence function. The BA non-linear optimization has been implemented using Google Ceres [24]. After each BA iteration, spurious map points are removed if: 1) the reprojection error of the point on the keyframe used for its creation is over a threshold (the median of reprojection errors of all measured map points). 2) the point is detected only in two keyframes, although it is visible in more than two. 3) The ratio between number of times

point is measured and number of times point is predicted is smaller than 0.3 (empirically defined). The mapping thread is also responsible for the initialization of new map points. Once a new keyframe is added to the map, matches between the new keyframe ORB points and all other keyframes in the map are sought. We use standard patch correlation guided by epipolar geometry.

Bootstrapping Previous mapping and tracking processes assume that there is a map. Next we describe how the map is initialized from scratch (block B in Fig. 3). For bootstrapping, the system has to select two keyframes that render enough parallax, this selection has proven an issue in VSLAM. We use a simple EKF VSLAM with all features encoded in inverse depth [12]. This approach can handle low parallax geometries, being able to exploit every single image in the sequence to estimate the map and the camera position. We process images until most of the map points are detected with enough parallax. Then we consider the first and the last processed images as the two initial keyframes for BA. Given these two keyframes and their relative locations, robust new point matches are computed by epipolar search, as in the mapping thread. Afterwards, an initial guess for the map points and two keyframe positions is fed to BA. The proposed EKF VLSAM only initializes points in the first frame. If those points fail to be tracked, or go out of the field of view before rendering enough parallax, the map is discarded and a new initialization is launched automatically.

Camera relocation Tacking can be lost because of camera occlusion, feature deletion due to fast camera motion or failure to track enough map points. Then the camera has to be located with respect to the map from scratch. Our system detects all the ORB points in the current image (block E in Fig. 3). They are matched with respect to the ORB descriptors of all the map points using as score the ratio between closest to second-closest neighbors to compute the putative matches. Then a perspective-three-point (P3P) [25] from random samples of size 4 is executed. The number of Random Sample Consensus (RANSAC) iterations are limited by the frame rate. To validate the relocation, the tracking algorithm has to produce a coherent position for the next frame in the sequence, otherwise relocation is re-attempted with the new frame.

4.2 Registration of pre-operative model with VSLAM map

Registration is initialized interactively by the practitioner once the VSLAM has been bootstrapped. The practitioner selects the 2D anchor points over the live video stream by tapping on the tactile screen of the tablet-PC. The 3D coordinates of the 2D anchor points are computed and appended to the map following the procedure described in Algorithm 1 (a variant of [26]). The P3DM is then translated, rotated and scaled to align the landmarks in the model with the anchors in the map. Initialized anchor points store a correlation patch to match in other keyframes. The correlation matching is guided by the epipolar geometry. Once two keyframes with proper matches are found, the 3D coordinates of the anchor point are triangulated, then the matches are propagated among previous keyframes. The BA is iterated to refine the 3D geometry of the map anchors. These anchor points are never removed from the map. On the arrival of a new keyframe,

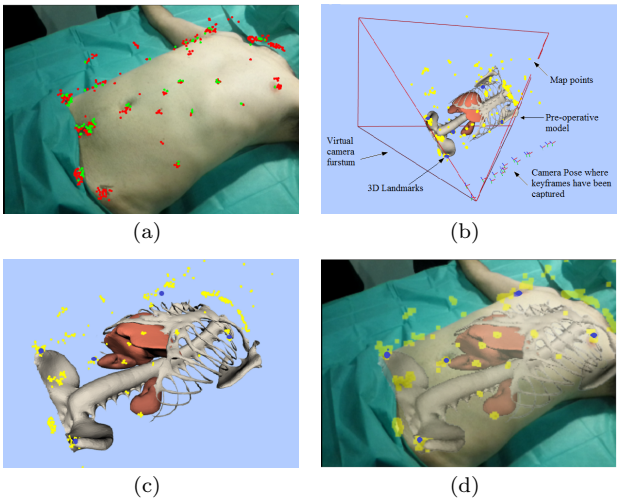
geometry-guided search for correlation matches is performed and T_{pm}(rotation, translation and scaling) is recomputed using [27] by minimizing the alignment error $err = \sum_{i=1}^{n} \|l_i - T_{pm} \cdot L_i\|$ (cf. Fig. 4 [b]). After each mapping iteration of the mapping thread, T_{pm} is refined with the newly available l_i estimations.

Input : List of keyframes with their estimated positions
Input : Query image in which anchor points are selected
Input : 3D coordinates of landmarks from P3DM (L_i)
Output: Transformation T_{pm} from P3DM to map
foreach *selected 2D anchor point* **do**
  Extract a square patch around the point
  Perform epipolar-guided cross correlation with all the map keyframes
  Estimate 3D coordinates (l_i) using triangulation
  Find more matches in previous keyframes
  Append l_i to the map
end
Apply BA to refine (l_i) estimations
Compute T_{pm} from L_i and l_i [27]

Algorithm 1: P3DM registration using anchor points from a real image

4.3 See-through AR

To provide AR overlay on the live video: 1) The VSLAM tracking thread provides a position estimate for each frame of the live stream, then a virtual camera with the same intrinsic parameters of the tablet-PC camera is located at the estimated position in the virtual scene (cf. Fig. 4[b]). 2) The image acquired by the virtual camera, taking into account the tablet-PC camera distortion, is rendered (cf. Fig. 4[c]). 3) The fusion is performed (cf. Fig. 4[d]) between the real camera image (cf. Fig. 4[a]) and the rendered one (cf. Fig. 4[c]).



(a) (b)

(c) (d)

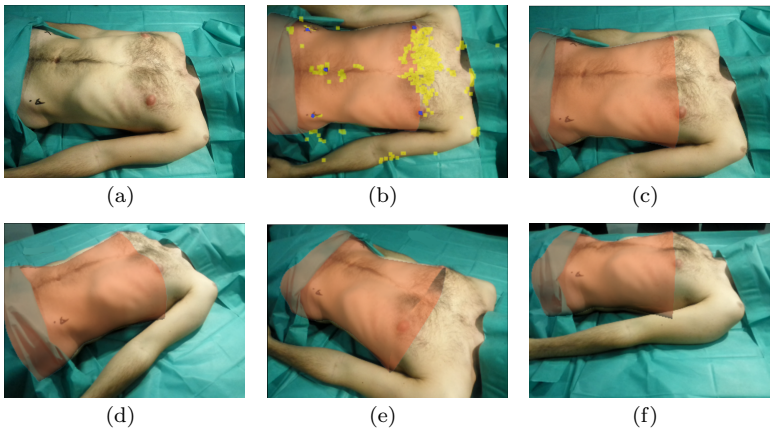
Fig. 4: AR insertion. (a) Tablet-PC camera frame with projected (red) and matched (green) map points. (b) Virtual 3D scene including the registered P3DM. (c) Virtual camera image. (d) Fused AR image.

5 Results

The proposed system has been implemented in C++ with OpenCV and VTK libraries and executed on a Sony VAIO Duo 13 tablet-PC with Intel(R) Core i7 (1.8 GHz), 8 GB RAM, with a camera of 640x480 at 30fps. Firstly, the system performance was evaluated on in-vivo data in terms of computation time with two volunteers, each of them laying on the table while the practitioner holds the tablet-PC and moved around them. The CT scans of these volunteers were performed several years ago. Secondly, the system accuracy was assessed by means of several experiments with fiducials; the first experiments were on four in-vivo pigs, the second on a phantom. All computations were performed exclusively on the tablet-PC. Beforehand, the focal and distortion of the tablet-PC camera were calibrated using [28]. Every P3DM in our experiments was segmented using our own software but can be obtained, for clinical applications, using a commercial service like Visible Patient[29]. More details can be appreciated in the accompanying video.

5.1 Volunteers experiments: computation time evaluation.

In this experiment, the time required for each step of the system was evaluated. For both volunteers, the five anatomical landmarks chosen as anchors for registering the P3DM were: right nipple, left nipple, umbilicus, right iliac crest and left iliac crest. The left and right iliac crests were marked with a pen on the skin of both volunteers, to easily identify them in the 2D images (cf. Fig. 5[a] and Fig. 6[a]). Fig. 5 and Fig. 6 show AR annotated frames for both volunteers from different points of view.



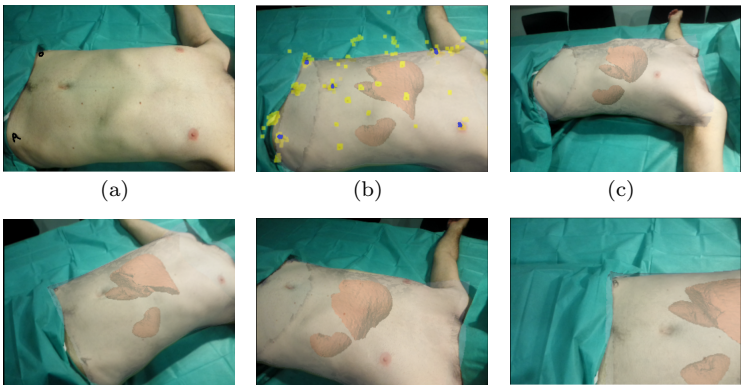
(a) (b) (c)

(d) (e) (f)

Fig. 5: Experiment on first volunteer. (a) Real tablet-PC image. (b) Skin registration with anchor points (in blue) and automatic map points. (c-f) Skin AR overlays over frames from different points of view.

VSLAM initialization: the VSLAM bootstrapping did not fail in any of the experiments. It was initialized using on average less than 20 frames. If failed, the initialization was automatically relaunched, and eventually succeeded.

Camera tracking and VTK rendering: Average tracking time was approximately 32 ms per frame for a map size that ranged between 180-200 points and 30-40 keyframes for video sequences composed of 750-900 frames. The average VTK rendering time was approximately 33 ms per frame, including the ideal projective imaging, the distortion and the fusion with the real frame. After each mapping step, the anchors 3D locations were updated, hence the AR insertion location in the map had to be recomputed, which took less than 1.2 ms. The time of initial insertion of the P3DM into the map can take up to 3 seconds depending on the sequence, due to searching for the anchor matches in all the keyframes. Therefore, total average time was 66.2 ms per frame.



(a) (b) (c)

(d) (e) (f)

Fig. 6: Registration of transparent skin, liver, left kidney and right kidney on the body of the second volunteer from different points of view.

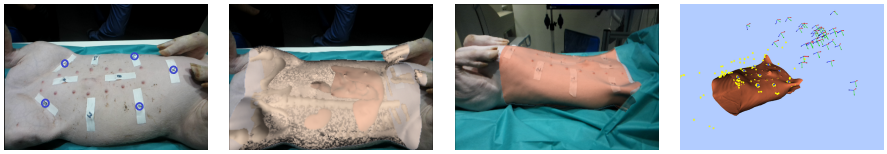
Loss of tracking and relocalization performance: In case of lateral (cf. Fig. 5[f] and Fig. 6[c].) or close up (cf. Fig. 6[f]) tablet-PC movements, the P3DM can still be registered even if most of the anchor points are not visible. Camera tracking is robust to partial scene occlusion since few map points are needed for VSLAM to estimate the tablet-PC position (cf. Fig. 9[b] and [c]). In case of full scene occlusion or severe camera motion, the relocalization module in VSLAM always relocated the tablet-PC position once a few map points were visible again, which required approximately 15 ms. As a result of this module, re-initialization of the whole system in the case of tracking loss is not necessary.

5.2 Accuracy evaluation

To assess the registration accuracy of the proposed system, experiments on four pigs were performed and the surface fiducial registration error (FRE) as well as the target registration error (TRE) were reported. Additionally, a plastic phantom was used to evaluate the registration accuracy on internal body structures that are far from the anchor points used for registration.

5.2.1 Data acquisition

Each pig was placed on the CT table, and nine radio-opaque markers were stuck on its skin before acquisition (cf. Fig. 7[a]). The CT scan was performed with breath-hold via a mechanical ventilation system. For each pig, two videos were recorded, one with breath-hold and another during respiration. The P3DM and the 3D coordinates of the markers were extracted from the CT images.



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

Fig. 7: Experiments on pigs. (a) Nine radio-opaque markers were attached to the surface of the pig. (b) P3DM composed of skin, bones, liver, left kidney and right kidney overlaid on an image of the first pig. (c) Lateral view of skin registration on the second pig. (d) Keyframe locations during camera motion around a pig.

5.2.2 Registration accuracy on pigs

The 3D coordinates of the markers extracted from CT were considered as ground truth. All markers were clicked on 2D image and their 3D coordinates computed and appended to the map. Only five markers were used as anchors to compute the 3D/3D registration those are displayed in blue in Fig. 7[a]. Fig. 7[b] and [c] shows the registration results on two pigs from different directions. The keyframe locations during camera motion around one of the pigs are displayed in Fig. 7[c] and represented by axes. The averaged $\overline{FRE}$ over all the frames in the sequence was calculated from the five markers used in the registration. The averaged $\overline{TRE}$ over the sequence was computed from the remaining four markers. $\overline{FRE}$ and $\overline{TRE}$ are defined in eq. (2):

$$\overline{FRE} = \frac{1}{F}\sum_{f=1}^{F}\frac{1}{5}\sum_{i=1}^{5}||l_i - T_{pm}\cdot L_i|| \qquad \overline{TRE} = \frac{1}{F}\sum_{f=1}^{F}\frac{1}{4}\sum_{i=6}^{9}||l_i - T_{pm}\cdot L_i|| \quad (2)$$

where F refers to the number of processed frames. As defined in eq. (2), the distance between the two point clouds was computed for every frame. In the Inner summation of eq. (2), the average distances of the five markers used for the registration and average distances of the remaining four markers were computed. Then $\overline{FRE}$ and $\overline{TRE}$ over all frames in the sequence were defined from the outer summation in eq. (2). The length of all video sequences ranged between 600 and 800 frames with 30 to 40 keyframes and map sizes between 176 and 279 points.

Each video was processed five times, each time the same frame was used to select the anchors. For the five registration trials on each pig sequence, the minimum, maximum and mean values of $\overline{FRE}$ and $\overline{TRE}$ are reported in Table 1 during breath-hold. Table 2 shows the influence of the breathing.

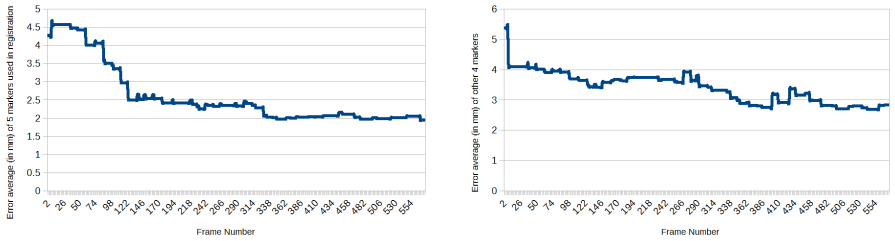
Table 1: $\overline{FRE}$ and $\overline{TRE}$ (in mm) of the four pigs sequences recorded during breath-hold.

	Pig 1			Pig 2			Pig 3			Pig 4		
	min	max	mean	min	max	mean	min	max	mean	min	max	mean
$\overline{FRE}$	2.72	3.07	2.94	2.41	2.61	2.52	3.42	4.27	3.82	1.01	2.66	1.55
$\overline{TRE}$	3.36	3.99	3.74	3.38	3.98	3.69	3.75	4.28	4.07	1.5	2.88	2.2

Table 2: $\overline{FRE}$ and $\overline{TRE}$ (in mm) of the four pigs sequences recorded during breathing

	Pig 1			Pig 2			Pig 3			Pig 4		
	min	max	mean	min	max	mean	min	max	mean	min	max	mean
$\overline{FRE}$	2.53	3.64	3.11	2.76	3.7	3.1	4.62	6.09	5.32	2.62	4.22	3.1
$\overline{TRE}$	3.49	4.56	3.92	3.65	4.08	3.88	4.95	6.24	5.6	2.62	4.62	3.44

After the initial insertion into the map, the P3DM is affected by a small jittering, due to the low number of keyframes and the poor geometrical conditioning. On the arrival of keyframes with a wider baseline, thus rendering bigger parallax, this jittering disappears within a few seconds, according to our experiments. Afterwards, the estimation of the anchor points 3D coordinates becomes accurate and so does the 3D/3D registration (cf. Fig. 8).



(a) (b)

Fig. 8: Evolution of the average distances between l_i and L_i (in mm) over all frames of one video sequence in case of breath-hold. (a) The average distances between the 5 markers used in the registration. (b) The average distances of the remaining 4 markers.

5.2.3 Registration accuracy on phantom

A phantom with a plastic liver was used to evaluate the system accuracy for points far from the body surface. 13 markers were attached on the external surface, 2 markers on the plastic liver and 4 markers on the phantom base (cf. Fig. 9[a]). The phantom sequences and CT were obtained following the same steps as those of the pig experiments in Sec. 5.2.1. Five markers on phantom surface were used to compute the registration (cf. Fig. 9[a]). Table 3 shows $\overline{FRE}$ of the five markers used for the registration, $\overline{TRE}$ of the two liver markers and $\overline{TRE}$ of the four markers at the phantom base. All were computed after processing the full

sequence and averaging the results of five registration trials.

(a) (b) (c)

Fig. 9: Experiments on phantom. (a) Markers in blue were used for the registration. (b) and (c) Liver AR overlay with partial scene occlusion.

Table 3: $\overline{FRE}$ and $\overline{TRE}$ (in mm) after processing the whole phantom sequence.

$\overline{FRE}$			$\overline{TRE}$ (two liver markers)			$\overline{TRE}$ (four markers at the base)		
min	max	mean	min	max	mean	min	max	mean
2.28	7.74	5.1	5.16	9.7	6.61	9.9	13.7	11.8

As shown in Table 3, the closer the target to the skin, the better the registration accuracy. The four markers on the phantom base represent the worst target position, i.e. close to the skin of the back. Therefore, 11.8mm can be considered the worst system accuracy. For the sake of completeness, the registration accuracy using all the 13 markers stuck on the surface has also been computed and provide a reduction between 1.5-2.0 mm on both $\overline{FRE}$ and $\overline{TRE}$. It is worth noting that the $\overline{FRE}$ is larger than in case of pigs due to utilization of different markers. The markers used with pigs were covered and a pen was used to mark their center of mass to be easily identifiable in the images (cf. Fig. 7[a]). In the phantom the markers were not covered and hence there were some inaccuracies in clicking the center of the cross shape of each markers.

6 Conclusion and Future work

A VSLAM-based on-patient AR visualization system is presented, which can be seamlessly integrated into the OR as the only external device is a commercial tablet-PC computer. The proposed system provides real-time performance, robustness to occlusion and detection failure. It requires minimal interaction with medical staff, i.e. the definition of the anchors by clicking on the live video. This is interaction is considered non-disruptive by most surgeons. In contrast to marker-based AR, our system is able to provide AR overlays even if none of the anchors used for the registration remains visible. Experimental results show the applicability of the proposed system, both in terms of computation time and accuracy. Although the system can already provide a great assistance, it can be further improved. Extension from rigid to non-rigid registration would allow taking breathing motion into account.

Acknowledgment: This work is supported by the Direccíon General de Investigacíon Científica y Téctinca of Spain under Project DPI2015-67275-P

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical approval: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. In addition to All applicable international, national, and/or institutional guidelines for the care and use of animals were followed.

Informed consent: Informed consent was obtained from all individual participants included in the study.

References

1. Hallet J, Soler L, Diana M, Mutter D, Baumert TF, Habersetzer F, Marescaux J, Pessaux P(2015) Trans-Thoracic Minimally Invasive Liver Resection Guided by Augmented Reality. Journal of the American College of Surgeons 220(5):e55e60.
2. Kilgus T, Heim E, Haase S, Prufer S, Muller M, Seitel A, Fangerau M, Wiebe T, Iszatt J, Schlemmer HP, Hornegger J, Yen K, Maier-Hein L(2015) Mobile markerless augmented reality and its application in forensic medicine. IJCARS 10(5):573-586.
3. dos Santos T, Seitel A, Kilgus T, Suwelack S, Wekerle AL, Kenngott H, Speidel S, Schlemmer HP, Meinzer HP, Heimann T, Maier-Hein L (2014) Pose-independent surface matching for intra-operative soft-tissue marker-less registration. Med Image Anal 18(7):1101-1114.
4. Macedo M, Souza A, Giraldi G (2014) High-Quality On-Patient Medical Data Visualization in a Markerless Augmented Reality Environment. SBC Journal on Interactive Systems 5(3):41-52.
5. Lee J, Huang C, Huang T, Hsieh H, Lee S (2012) Medical augment reality using a markerless registration framework. Int J in Expert Systems with Applications 39(5):5286-5294.
6. Chen X, Xu L, Wang Y, Wang H, Wang F, Zeng X, Wangb Q, Egger J (2015) Development of a surgical navigation system based on augmented reality using an optical see-through head-mounted display. J Biomed Inform 55:124-131.
7. Rassweiler JJ, Mller M, Fangerau M, Klein J, Goezen AS, Pereirac P, Meinzerb HP, Teberd D (2012) iPad-Assisted Percutaneous Access to the Kidney Using Marker-Based Navigation: Initial Clinical Experience. European Urology 61(3):628-631.
8. Muller M, Rassweiler M, Klein J, Seitel A, Gondan M, Baumhauer M, Teber G, Rassweiler JJ, Meinzer HP, Maier-Hein L(2013) Mobile augmented reality for computer-assisted percutaneous nephrolithotomy. IJCARS 8(4):663-675.
9. Sun Y, Luebbers H, Agbaje J, Schepers S, Vrielinck L, Lambrichts I, Politis C (2013) Validation of anatomical landmarks-based registration for image-guided surgery: an in-vitro study. J Cranio Maxill Surg 41(6):522-526.
10. Schneider A, Baumberger C, Griessen M, Pezold S, Beinemann J, Philipp Jurgens P, Cattin PC (2014) Landmark-Based Surgical Navigation. Clinical Image-Based Procedures. Translational Research in Medical Imaging 8361:57-64.
11. Davison AJ (2003) Real-Time Simultaneous Localisation and Mapping with a Single Camera, IEEE Int Conf on Computer Vision 2:1403-1410.
12. Civera J, Davison AJ, Montiel JMM (2008) Inverse Depth Parametrization for Monocular SLAM, IEEE Trans in Robotics 24(5):932-945.
13. Civera J, Grasa OG, Davison AJ, Montiel JMM (2010) 1-Point RANSAC for extended Kalman filtering. Application to realtime structure from motion and visual odometry. J Field Robot 27(5):609-631.
14. Grasa OG, Bernal E, Casado S, Gil I, Montiel JMM (2014) "Visual SLAM for Handheld Monocular Endoscope", IEEE Transactions on Medical Imaging 33(1):135-146.
15. Grasa OG, Civera J, Montiel JMM (2009) EKF Monocular SLAM 3D Modeling, Measuring and Augmented Reality from Endoscope Image Sequences. In MICCAI, vol. 2.
16. Klein G, Murray D (2007) Parallel Tracking and Mapping for small AR Workspace. IEEE and ACM Int Symposium on Mixed and Augmented Reality(ISMAR), pp 1-10.

17. Mikhail EM, Bethel JS, McGlone JC (2001) Introduction to modern photogrammetry. New York: John Wiley & Sons.
18. Mur-Artal R, Montiel JMM, Tards JD (2015) ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans on Robotics 31(5):1147-1163.
19. Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: An efficient alternative to SIFT or SURF. IEEE Int Conf Comp Vis (ICCV), pp 2564-2571.
20. Lowe DG (1999) Object recognition from local scale-invariant features. In Proc. of IEEE Int Conference on computer vision, 2:1150-1157.
21. Rosten E, Drummond T (2006) Machine learning for high-speed corner detection. In Proc of 9th European Conference on Computer Vision, pp 430-443.
22. Shi J, Tomasi C (1994) Good features to track, IEEE Computer Society Conference in Computer Vision and Pattern Recognition, pp 593-600.
23. Lowe DG (2004) Distinctive Image Features from Scale-Invariant Keypoints, Int J of Computer Vision 60(2):91-110.
24. http://ceres-solver.org/. Accessed 4 April 2016
25. Gao X, Hou X, Tang J, Cheng H (2003) Complete solution classification for the perspective-three-point problem, IEEE Trans Pattern Anal 25(8):930-943.
26. Gálvez-López D, Salas M, Tards JD, Montiel JMM (2015) Real-time Monocular Object SLAM. J of Robots and Autonomous Systems. doi: 10.1016/j.robot.2015.08.009
27. Horn BK (1987) Closed-form solution of absolute orientation using unit quaternions. J OPT SOC AM A 4(4):629-642.
28. Zhang Z (2000) A Flexible New Technique for Camera Calibration. IEEE Trans Pattern Anal 22(11):1330-1334.
29. https://www.visiblepatient.com/en/. Accessed 4 April 2016