Joint Scene and Object Tracking for Cost-Effective Augmented Reality Guided Patient Positioning in Radiation Therapy

Hamid Sarmadi[2], Rafael Muñoz-Salinas[1,2,*], M.Álvaro Berbís[3], Antonio Luna[4], R. Medina-Carnicer[1,2]

Abstract

Background and Objective
The research done in the field of Augmented Reality (AR) for patient positioning in radiation therapy is scarce. We propose an efficient and cost-effective algorithm for tracking the scene and the patient to interactively assist the patient's positioning process by providing visual feedback to the operator. Up to our knowledge, this is the first framework that can be employed for mobile interactive AR to guide patient positioning.

Methods
We propose a point cloud processing method that combined with a fiducial marker-mapper algorithm and the generalized ICP algorithm tracks the patient and the camera precisely and efficiently only using the CPU unit. The alignment between the 3D reference model and body marker map is calculated employing an efficient body reconstruction algorithm.

Results
Our quantitative evaluation shows that the proposed method achieves a translational and rotational error of $4.17\,\mathrm{mm}/0.82°$ at 9 fps. Furthermore, the qualitative results demonstrate the usefulness of our algorithm in patient positioning on different human subjects.

Conclusion
Since our algorithm achieves a relatively high frame rate and accuracy employing a regular laptop (without the usage of a dedicated GPU), it is a very cost-effective AR-based patient positioning method. It also opens the way for other researchers by introducing a framework that could be improved upon for better mobile interactive AR patient positioning solutions in the future.

Keywords: Patient Positioning, Augmented Reality, ArUco Markers, Generalized ICP, Marker Mapper, Surface Guided Radiation Therapy (SGRT)

1. Introduction

The traditional process of radiation therapy is usually separated into two phases: the planning phase and the treatment phase. The treatment phase itself normally consists of multiple treatment sessions where the malignant tissue is radiated. In the planning phase, a CT scan is performed on the patient and, based on the result, the area to be radiated is planned for the subsequent radiation sessions in the treatment phase. It is therefore crucial that in the radiation session the position of the patient is the same as the position in the planning phase.

With the introduction of head-mounted displays such as the Microsoft Hololens[5], Augmented Reality (AR) has gained traction in medical research, specifically in surgical applications such as training [12, 16] and intervention [19, 4, 20]. Medical experts questioned about these systems have shown overwhelmingly positive opinions [12, 30]. However, the application of AR to guide patient positioning is so scarce that there is no related works in the recent literature reviews [9, 23, 39]. AR for patient positioning could have the potential benefit of assisting the operators by the interactive real-time visualization of the actual patient's position compared to the desired patient position (Fig. 1).

In the past decade, there has also been a growth of works that take advantage of consumer-level depth or RGB-Depth (RGB-D) cameras (e.g. Microsoft Kinect)

*Corresponding author

Email addresses: hamid.sarmadi@imibic.org (Hamid Sarmadi), rmsalinas@uco.es (Rafael Muñoz-Salinas), a.berbis@htime.org (M.Álvaro Berbís), aluna70@htime.org (Antonio Luna), rmedina@uco.es (R. Medina-Carnicer)

[1]Computing and Numerical Analysis Department, Edificio Einstein. Campus de Rabanales, Córdoba University, 14071, Córdoba, Spain, Tlfn:(+34)957212255

[2]Instituto Maimónides de Investigación en Biomedicina (IMIBIC). Avenida Menéndez Pidal s/n, 14004, Córdoba, Spain, Tlfn:(+34)957213861

[3]HT Médica, Hospital San Juan de Dios. Avda Brillante 106, 14012, Córdoba, Spain

[4]HT Médica, Clínica las Nieves, Carmelo Torres 2,23007, Jaén, Spain

[5]https://www.microsoft.com/en-us/hololens

Figure 1: Setup employed for capturing data (a), and a frame from one of our testing sequences on a subject without (b) and with (c) augmented reality produced by our program. ArUco markers are attached to the body for tracking, while another set of ArUco markers are placed on the environment to track the camera pose in the environment. In c, the red model mesh represents the tracked body of the patient, and the blue model mesh shows its desired pose. Our method also provides numerical feedback on the corner of the AR image in the form of rotational and translational error.

[42, 22, 21] which are very useful in AR applications. These sensors are affordable and also provide a real-time depth map of the scene and a corresponding color image. In other words they can give a dense real-time geometrical image of the scene rather than the sparse pose estimation that is possible using e.g. fiducial markers. Simultaneously, research on fiducial planar markers has proposed fast, robust and cheap methods for precise camera pose tracking. They do not need special equipment except for a color camera and a set of printed markers. These give fiducial planar marker detectors such as ArUco [13, 31] many advantages with respect to the traditional infrared-based markers.

Taking advantage of these two recent technologies, this paper proposes a novel method for assisted patient positioning which is able to simultaneously track the patient and the treatment environment. Our system is able to render a virtual overlay of the patient's current pose and its desired pose using a freely moving RGB-D camera. Hence it can be employed for mobile interactive AR to guide pa-

tient positioning. To our knowledge this is the first method capable of such operation. Additionally, it is possible to take advantage of our approach for patient monitoring. This is possible by fixing the RGB-D camera pointing towards the patient. However this would not take advantage of the full potential of our method which lets the RGB-D camera move freely.

Our novel RGB-D based, model-based, object tracking algorithm is accurate and fast at the same time without the need of general purpose GPU computing on a dedicated GPU, using only the CPU unit. This makes our method usable on a wide range of hardware which makes it more accessible and cost-effective. We also believe this algorithm can have general applications beyond patient positioning or medicine. Our approach only requires an over the counter RGB-D camera and an average consumer laptop without the need for a powerful dedicated GPU. Although we have not seen any other similar work to our approach even in the industry, it is far more affordable than other industry level non-invasive surface-guided pa-

tient positioning methods such as AlignRT, Catalyst, and IDENTIFY [14] that do not even have our AR capabilities.

The rest of this paper is structured as follows. Section 2 explains the related works, then in Section 3 the proposed approach is introduced. Section 4 describes the experimental results for validating our approach, after that those results are discussed in Section 5, and finally Section 6 draws some conclusions and future works.

2. Related Work

2.1. Patient Positioning in AR and Computer Vision

From the few methods that investigate AR for patient positioning the early works do not perform any patient tracking and leave it to the user to detect when the pose is correct only with overlaying the desired pose of the patient on the video [37, 38, 11]. These approaches need a calibration method to calibrate the fixed cameras with respect to the linear accelerator (linac). Another more recent method gives the possibility of using moving cameras however it still does not perform patient tracking and for that relies on the user's eyes [8]. The most recent method we found that claims it has application in AR for patient positioning is [10]. They use a time of flight camera fixed to the linac and apply an advanced registration method to align the current patient's geometry to the reference model obtained in the planning phase. The mentioned methods either assume that the camera is fixed in the environment or they do not track the patient. A fixed camera limits the view in which the patient can be seen. Especially it makes the algorithm unsuitable to be used in conjunction with head-mounted displays (such as HoloLens) for AR. Another disadvantage is that the camera cannot show different parts of the patient on demand and it has to be fixed from before just for a specific point of view. Furthermore, the camera needs to be calibrated with respect to the linac and regularly checked if the calibration is still correct. Tracking the patient on the other hand, which is absent from most of the methods we mentioned, is necessary to give the user correct numerical or visual indications of the amount of the error in positioning the patient.

A new novel approach to computer-assisted patient positioning is presented in [33]. They take advantage of a heightmap data structure reconstructed using their Global ICP algorithm and an RGB-D sensor. They use it to compare the pose of the patient in the planning phase and the treatment phase in radiation therapy. They also take advantage of ArUco markers [13, 31] to align the reconstructed scenes. However, this approach is not capable of tracking the patient and giving visual feedback to the operator in real-time.

2.2. Joint Scene and Object Tracking in Computer Vision

Simultaneous tracking of the camera (scene) and the object in the scene is not a new idea. One early work is [24] where they apply self-localization and tracking of dynamic objects at the same time. The data is captured using a LIDAR in a self-driving car scenario. Here the scene is treated as a separate object. Another joint scene/object tracking algorithm has been employed for tracking people using data from a moving monocular camera [6].

One more recent work [32] is a SLAM algorithm that recognizes moving object by segmenting them and treats the scene (background) just as another object. In this research, an RGB-D sensor is utilized for the detection and reconstruction of multiple rigid moving objects. Nevertheless, they need two dedicated GPUs one for performing the SLAM algorithm and another one just for the segmentation of objects using a convolutional neural network (CNN). Another similar approach is presented in [36], however, in this approach they can additionally remove non-rigid moving objects from the background using a probabilistic framework for robust camera tracking. Notwithstanding they still need dedicated GPUs for object detection and reconstruction.

2.3. Model-based Object Tracking In Computer Vision

An early example of 3D model-based tracking is presented in [7]. The authors present an algorithm that tracks a 3D object by tracking the contours of its projection on the image plane on using data from a monocular camera.

In [5] a particle-filter based tracking approach is presented that tracks the object using edge features. Keypoint features are employed for the initialization of tracking. They do not reach real-time performance and only suggest it could be possible with an implementation that takes advantage of GPU computing.

A Gaussian filter based tracking method using depth map input is put forward in [15]. Higher accuracy is achieved by robustification of the Gaussian filter and real-time performance is obtained by reducing the complexity of the filter. Despite that, no solution is provided for robust initialization or re-initialization of tracking which could be expected since they only use depth input.

A more recent approach is introduced in [41]. In this work CAD models of the object are employed in conjunction with reference pictures taken of them, however, their objects have simple shapes with planar surfaces. They use image feature matching to initialize a template matching algorithm and then deduce the 3D pose from that. Their algorithm is real-time however they require a discrete GPU.

In [40] an approach for 3D tracking is presented that uses three different types of constraints: texture-based points, edges, and point-to-plane distance. Although their algorithm does not need any GPU acceleration it needs to use very simple geometries containing few surfaces for point-to-plane geometry-based registration which we think is not very appropriate for tracking complex shapes such as the human body.

2.4. Comparison with other approaches

This section compares, in Table 1, our proposal with three of the most common commercial solutions (AlignRT, Catalyst and Sentinel) and other experimental solutions for patient positioning.

The table indicates, amongst other aspect, the type of sensor employed, the registration speed and precision. We would like to note that none of the methods are designed

Method	Sensor	Registration Speed (fps)	Precision (mm)	Static Camera	Regular Calibration	Mobile AR
AlignRT	Proprietary	≤ 5[28]	<1 [43]	Yes	Yes	No
C-Rad Catalyst	Proprietary	Not Reported	1 [43]	Yes	Yes	No
C-Rad Sentinel	Proprietary	<1 [35]	2 [35]	Yes	Yes	No
Ehsani et. al. [10]	Kinect V2	Not reported	2	Yes	Yes	No
Bauer et. al. [3]	Kinect V1	Not reported	12	Yes	Yes	No
Sarmadi et. al. [33]	Xtion Pro Live	<1	11	No	No	No
Ours	Realsense L515	9	4	No	No	Yes

Table 1: Comparison of different patient positioning approaches.

to be usable in mobile AR (except ours). In addition, our approach has a high registration speed, specially if we consider that no GPU is required.

Regarding the precision, AlignRT is the most precise method, while ours obtains a precision of 4mm. It must be considered, though, that our method is designed to work with a moving camera, which is a more challenging problem since its position must be recalculated in every frame. Nevertheless, as we explain in the experimental section, there is room for improvement in the future as the depth camera technology evolves.

One advantage of our method is that it does no need for regular recalibration like the others. There is only one other method that employs a moving camera (Sarmadi et. al. [33]). However, they need to scan the person by moving the sensor to create only one frame of reconstruction which takes several second. Hence their method is not suitable for live feedback needed in AR.

Finally, we should mention an important disadvantage of the commercial methods which is their high price. As an example, the price of an AlignRT system starts from £150,000 plus £20,000 for annual service [1]. Other noncommercial proposed methods however normally use over the counter sensors, hence they have a low price. For example, the Intel Realsense L515 is priced at only $349.

3. Methodology

3.1. Overview

Our proposed approach is designed to give real-time feedback for patient positioning in radiation therapy through AR to the person who performs the positioning. We assume that the non-rigid deformation of the patient's body matches that of the patient's desired pose. Then, our method enables viewing a virtual model of the patient overlaid on top of his/her body, and also a virtual model where the patient needs to move to. Furthermore, real-time quantitative feedback of the positioning error is shown to the operator in the augmented image. The mentioned information can be used for visualization in head-mounted displays or a tablet connected to the camera.

Figure 2 provides a summary of our approach which is explained in detail in this section.

Our method requires the following inputs obtained from the planning phase:

- A map of the treatment room. This is obtained using the UcoSLAM [26] algorithm, which employs 2D image features (keypoints) and ArUco planar markers [13, 31] using an RGB-D camera.

- A 3D reference model of the patient's body surface that can be obtained from a 3D laser scanner or from the CT scan performed in the planning phase.

- The correct pose of the 3D reference model w.r.t. the created environment map which indicates the desired pose of the patient in the treatment room. This could be done by a calibration method such as in [38].

We assume that the steps corresponding to the planning phase have been carried out according to the mentioned methods.

In the treatment phase, for every session, the patient needs to wear tight-fitting clothes with small ArUco planar markers printed (similar to [2]), or using stickers directly attached to the body with markers printed on them. We should mention that attaching the sticky markers is very easy and takes no more than a couple of minutes. One only needs to randomly place several of them in the body to make sure they are properly visible from the camera's view point. While the markers do not need to be in the same position on the body from one treatment session to another, they must remain fixed within the session. In the first step of our algorithm, we create a temporary 3D body model of the patient including the 3D geometry and the positions of the markers on the patient's body. In the second step, we register the position of the markers (body marker map) to the accurate reference 3D model created in the planning phase. The third step is the only real-time step and happens while the operator is positioning the patient. In this stage, both the body of the patient and the environment are tracked with respect to the camera. The body is tracked by a combination of body marker map tracking and geometrical alignment of the reference 3D model on the depth map captured with the camera. The camera is tracked with respect to the environment with the UcoSLAM algorithm using the environment map created in the planning phase. For visualization, the tracked camera pose is employed to overlay the desired body pose (shown in blue in Figure 1c) and the current body pose (shown in red). In addition, the difference between the desired body pose and the current patient's pose is calculated to show rotational and translational errors in patient position to the operator.

UcoSLAM is a new tracking algorithm that ourperforms other state-of-the-art SLAM algorithms in accuracy and

Planning Phase

UcoSLAM
Environment Map
Creation

Reference 3D Model
Acquisition

Determining Correct
Model Pose w.r.t
UcoSLAM Map

Treatment Phase

Preparation Stage

Attach Markers on
Patient's Body

Create Temporary
3D Body Model

Align Reference Model to
Temporary Model

Tracking Stage (Real-Time)

Patient Pose Tracking by
Marker Map & Generalized ICP

AR Visualization and
Error Feedback

Environment Tracking
by UcoSLAM

Figure 2: A high-level overview of our approach is presented. First, in the planning phase an UcoSLAM environment map of the treatment room, a 3D model of the body part, and the desired pose of the model with respect to the environment map are obtained. Then at each treatment session, in the preparation stage ArUco markers are attached to the patient's body, a temporary 3D body model is created and aligned to the reference model. In the tracking stage, which happens in real-time, the environment tracking by UcoSLAM is used to demonstrate the desired pose and patient pose tracking by marker map and generalized ICP is employed to show the patient's current pose. The desired pose is shown in blue color and the current pose is shown in red color in AR visualization. Finally numerical error feedback is added to the AR visualization.

speed as has been proved in [26]. An advantage of this method is that it is the unique one that can map and track ArUco planar markers as well as image features (keypoints) as part of its design. Since markers can stay in the same place for long term, they are good reference points specially for our application where we want to track the environment consistently in different treatment sessions. Hence we have chosen the UcoSLAM algorithm to map and track the environment in the treatment room.

The result of our system can be seen in Figure 1. The color image from the camera is shown with (Figure 1c) and without (Figure 1b) the augmented visualization. While the red transparent model represents the current patient's pose, the blue opaque model represents the desired one. The rotational and translation error of the positioning are also reported in the augmented image.

Since the main contribution of this paper is the process of simultaneously tracking the camera and the patient, we are not explaining in more details how to obtain the 3D reference model (e.g. from CT scan), create a UcoSLAM map

of the treatment environment, and determine the target pose of the reference model with respect to the UcoSLAM map. These steps, which are done in the planning phase, are out of the scope of this paper and do not need any novel algorithm for their implementation.

The rest of this section provides a detailed explanation of the different steps involved in the proposed method. Creation of a temporary 3D body model of the patient is explained in Subsection 3.2, alignment of the reference model to the temporary model is described in Subsection 3.3, and finally the tracking stage is detailed in Subsection 3.4.

3.2. Temporary Body Model Creation

An RGB-D video sequence of the patient laying on the treatment bed is recorded to create a temporary 3D model of their body at the beginning of each treatment session. We refer to it as *temporary* since it will only be used during the current treatment session. We propose a fast reconstruction method by combining UcoSLAM [26] and the generalized ICP [34] algorithm. Generalized ICP is a fast approximation of the point-to-plane ICP algorithm which we exploit to refine the result of UcoSLAM pose estimation. Generalized ICP is one of the fastest CPU based registration algorithms with a good accuracy compared to the state of the art [29]. The captured sequence is also used to create a three-dimensional map of the markers attached to the patient (*body marker map*) [25] that is employed for tracking the patient with respect to the camera.

Since the body marker map and temporary 3D model are obtained from the same video sequence, it is possible to establish the relationship between their reference system to align them. This is important since we will later need the alignment from the reference 3D model (CT scan) to the body marker map's coordinate system for correct pose estimation and visualization of the 3D model on top of the patient's body.

Formally speaking, let us assume:

$$F_i = (D_i, I_i), \quad i = 1 \ldots n \tag{1}$$

represents the data in the i-th frame of the RGB-D sequence, where D_i is the depth map and I_i is the RGB image.

We feed the marker mapper algorithm [25] with the sequence of the images $\{I_i\}_{i=1}^n$ to create the marker map, $\mathcal{M}$. Since the marker mapper does not need time-coherent input, only a subset of the frames is employed to speed-up computation.

$$\mathcal{M} = \text{MarkerMapper}(\{I_i | i \in \mathfrak{L}\}) \tag{2}$$

where

$$\mathfrak{L} = \{i \in Z^+ | 1 \leq i \leq n \ \wedge \ i/1 = \lfloor i/1 \rfloor\} \tag{3}$$

Here, $1 \in Z^+$ is a constant that enables processing only every l-th frame, and $\mathfrak{L}$ is the set of all valid indices.

The temporary geometrical reconstruction of the body and its alignment to the created body marker map are explained below.

3.2.1. Geometrical Body Reconstruction

Our geometrical body reconstruction algorithm takes advantage of both pose estimation from the UcoSLAM algorithm [26] and the generalized ICP registration algorithm [34]. An example of this geometrical reconstruction can be seen in Figure 3.

UcoSLAM is fed with the frame set $\{F_i\}_{i=1}^n$, generating as a result the set of 3D poses of the body w.r.t. the camera tracked for all frames, $\{P_i^U\}_{i=1}^n$ where P_i^U is the 4×4, 3D transformation matrix corresponding to the i-th.

We keep the body reconstruction in the form of a point cloud $\mathfrak{C}$ and go through the frame indices $i \in \mathfrak{L}$ sequentially converting each depth map D_i to a point cloud C_i:

$$C_i = \text{DepthToPointcloud}(D_i), \quad i \in \mathfrak{L} \tag{4}$$

$$\mathcal{C}_i = \text{Downsample}(C_i, d_i), \quad i \in \mathfrak{L} \tag{5}$$

Here the depth map D_i is converted to a point cloud, C_i, using the known camera parameters and then downsmapled to $\mathcal{C}_i$ by voxel downsampling using a size of d_i, which is calculated dynamically for each frame as:

$$d_i = \max(\sqrt[3]{V_i/N}, d_0) \tag{6}$$

where V_i is the volume of the bounding cube of $\mathcal{C}_i$, $N \in Z^+$ is a constant, and $d_0 \in R^+$ is the minimum possible voxel size, also a constant.

To perform the reconstruction we transform each point cloud $\mathcal{C}_i$ to its corresponding reconstruction pose, P_i^R, and add it to $\mathfrak{C}$.

In order to determine P_i^R we take P_i^U and refine it by the generalized ICP algorithm:

$$P_i^R = \begin{cases} \text{GeneralizedICP}(\mathcal{C}_i, \mathfrak{C}, P_i^U) & i \in \mathfrak{L} \wedge i > 1 \\ P_1^U & i = 1 \end{cases} \tag{7}$$

Again, not all frames are required for the reconstruction, thus, only a subset of them is employed to speed-up the computation. Also, since at the first frame $\mathfrak{C}$ is empty, there is no pose refinement done for the first frame and $P_1^R = P_1^U$.

To keep the size of the point cloud manageable, we also downsample $\mathfrak{C}$ after each addition:

$$\mathfrak{C} \leftarrow \text{Downsample}(\mathfrak{C}, d_{\mathfrak{C}}) \tag{8}$$

where $d_{\mathfrak{C}} \in R^+$ is a constant for the voxel size.

3.2.2. Alignment of Body Marker Map to Geometrical Body Reconstruction

When processing the reconstruction sequence, we obtain the body marker map $\mathcal{M}$ but also its pose P_i^M at each frame w.r.t. the camera:

$$(P_i^M, S_i) = \text{MarkermapPose}(\mathcal{M}, I_i), \quad i \in \mathfrak{L} \tag{9}$$

where $S_i \in \{true, false\}$ indicates whether marker map pose estimation was performed successfully.

Figure 3: Visualization of temporary 3D reconstruction of the patient (represented by a mannequin) and its alignment to the given 3D model. First a rough alignment is given by the user and then it is refined furthermore by the point-to-plane ICP algorithm. Please note that only every l-th frame from the input RGB-D frame sequence is used for 3D reconstruction.

Now we need to determine the transformation, T, that relates the coordinates systems of $\mathcal{M}$ and $\mathfrak{C}$. Let us define the set of all indices with successful marker map tracking by:

$$\mathfrak{S} = \{i \in \mathfrak{L} | S_i = true\} \qquad (10)$$

Since we have corresponding poses for each frame where the marker map is successfully tracked we can write:

$$P_i^R T = P_i^M, \quad i \in \mathfrak{S} \qquad (11)$$

Now it is possible to determine T using the least square method:

$$\left(\sum_{i \in \mathfrak{S}} (P_i^R)^\top P_i^R\right) T = \sum_{i \in \mathfrak{S}} (P_i^R)^\top P_i^M, \qquad (12)$$

which is a system that can be solved for all columns of T at the same time:

$$T = \left(\sum_{i \in \mathfrak{S}} (P_i^R)^\top P_i^R\right)^{-1} \sum_{i \in \mathfrak{S}} (P_i^R)^\top P_i^M. \qquad (13)$$

To make sure that the rotation component in T belongs to the SO(3) Lie group, we convert it to the axis angle representation and back:

$$T = \begin{bmatrix} R & \vec{t} \\ t_{4,1} & t_{4,2} & t_{4,3} & t_{4,4} \end{bmatrix} \qquad (14)$$

$$(r, \theta) \leftarrow \mathrm{AngleAxis}(R) \qquad (15)$$

$$\hat{R} \leftarrow \mathrm{RotationMatrix}(r, \theta) \qquad (16)$$

Here, R is the 3×3 rotation component of T, $\vec{t}$ is its translation component, r and θ are the rotation axis and angle obtained from R, and $\hat{R}$ is the rotation matrix created from r and θ.

We also fix the numbers on the last row of T to create a transformation belonging to the SE(3) Lie group. Finally,

we can write:

$$\hat{T} = \begin{bmatrix} \hat{R} & \vec{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (17)$$

where $\hat{T}$ is the final obtained transformation from the marker map coordinate system to the reconstruction coordinate system.

3.3. Reference Model to Temporary Model Alignment

The alignment of the 3D reference model to the temporary body reconstruction is done semi-automatically. First, the user manually gives a rough alignment and then we use point-to-plane ICP to refine it. For visualization in the manual input we create a mesh out of our temporary body model's point cloud using Poisson surface reconstruction [18]:

$$\mathfrak{M}_{\mathrm{rec}} \leftarrow \mathrm{PoissonSurface}(\mathfrak{C}) \qquad (18)$$

where $\mathfrak{M}_{\mathrm{rec}}$ is the triangle mesh surface created from the temporary model point cloud, $\mathfrak{C}$. It should be mentioned that we prune the resulting mesh so that it does not have vertices that are too far away from any point in $\mathfrak{C}$.

We denote the manual alignment given by the user by T_{manual} and refine this transformation by:

$$T_{\mathrm{refined}} = \mathrm{PointToPlaneICP}(\mathcal{V}_{\mathrm{rec}}, \mathcal{V}_{\mathrm{model}}, T_{\mathrm{manual}}) \qquad (19)$$

Here $\mathcal{V}_{\mathrm{model}}$ is the set of vertices from the patient's reference model mesh, $\mathcal{V}_{\mathrm{rec}}$ is the set of vertices from the patient's temporary reconstruction mesh, and T_{refined} is the result of the point-to-plane ICP alignment of $\mathcal{V}_{\mathrm{rec}}$ to $\mathcal{V}_{\mathrm{model}}$, initialized by T_{manual}. A visual example of 3D reconstruction to model alignment from our implementation can be seen Figure 3 in the first two images from right.

Finally to obtain the alignment from the marker map $\mathcal{M}$ to the reference model we can write:

$$\mathfrak{T} = T_{\mathrm{refined}} \hat{T} \qquad (20)$$

where $\hat{T}$ was defined in Eq. 17, and $\mathfrak{T}$ is a transformation that takes a point from the body marker map coordinate system to the reference model coordinates system. Now it is possible to perform our hybrid model based patient tracking taking advantage of $\mathfrak{T}$.

3.4. Tracking

Our tracking algorithm consists of two main parts: scene tracking and patient tracking. Scene tracking is done by the UcoSLAM algorithm in tracking mode. This approach needs a reconstruction of the scene in the form of a UcoSLAM map with the patient not being present. This reconstruction should be done in the planning phase of radiation therapy so that it is consistent between treatment sessions. The reason the patient should not be present while doing scene reconstruction is that it could confuse scene tracking later when the patient is present and moving in the scene.

Let us assume:

$$F_i^* = (D_i^*, I_i^*), \quad i = 1, \ldots, m \qquad (21)$$

represents the data in the i-th frame of the tracking sequence where D_i^* is the depth map and I_i^* is the RGB image. At each frame, first, we track the pose of the scene using the UcoSLAM algorithm:

$$P_i^{U*} \leftarrow \text{UcoSLAM}(F_i^*) \qquad (22)$$

where P_i^{U*} is the scene pose in real-time at frame i which is used to visualize the patient in its desired pose. This is useful for error calculation and visual feedback while performing the positioning. We also need to calculate the pose of the patient for proper visual feedback to the user.

3.4.1. Patient Tracking

Pose estimation of the patient is first done by ArUco marker map tracking [13, 31] of the created body marker map [25]. Then the pose is further refined employing the generalized ICP [34] algorithm, registering the 3D geometry of the reference model to the depth map in the current frame. Before this registration, some preparations are done on the 3D data to improve the result which are discussed below and are also visualized in Figure 4.

We denote the body marker map pose estimation by:

$$(P_i^{M*}, S_i^*) = \text{MarkerMapPose}(\mathcal{M}, I_i^*) \qquad (23)$$

Here, again $S_i^* \in \{true, false\}$ determines if the pose estimation is performed successfully and P_i^{M*} is the estimated pose.

Marker map pose estimation is prone to errors that can happen due to motion blur or imperfect marker map to model alignment. We compensate this by refining the pose estimation using the data from the depth map.

First of all, we need to create a point cloud from the depth map and then downsample it, similar to temporary body reconstruction:

$$C_i^* = \text{DepthToPointcloud}(D_i^*) \qquad (24)$$

$$\mathcal{C}_i^* = \text{Downsample}(C_i^*, d_*) \qquad (25)$$

where C_i^* is the point cloud created from D_i^*, $\mathcal{C}_i^*$ is the downsampled point cloud and $d_* \in R^+$ is the voxel size for voxel downsmapling which is a constant.

Similar to temporary model reconstruction, we define the set of frame indices where the marker map performs pose estimation successfully:

$$\mathfrak{S}^* = \{1 \leq i \leq m | S_i^* = true\} \qquad (26)$$

Let us assume that k is the first frame number where the marker map pose estimation is successfully performed:

$$\text{k} = \min\{i \in \mathfrak{S}^*\} \qquad (27)$$

Then we define the primary pose of the patient in the current frame:

$$P_i^* = \begin{cases} (P_i^{M*})\mathfrak{T}^{-1} & i \in \mathfrak{S}^* \\ P_{i-1}^* & i \notin \mathfrak{S}^* \wedge i > \text{k} \end{cases} \qquad (28)$$

To increase the speed of our algorithm, we reduce the number of vertices in our input model mesh:

$$\mathfrak{M}_{\text{model}}^* = \text{MergeCloseVertices}(\mathfrak{M}_{\text{model}}, d_*) \qquad (29)$$

where we merge neighboring vertices in model mesh, $\mathfrak{M}_{\text{model}}$, that are closer than the constant d_*. The merged vertices are replaced with a single vertex with the average of their position. The advantage of merging vertices instead of subsampling them by voxels is that the surface structure can be better preserved since merging is applied to neighboring vertices on the mesh. We define the vertices in $\mathfrak{M}_{\text{model}}^*$ as $\mathcal{V}_{\text{model}}^*$ and use them to refine the alignment of the pointcloud from depthmap, $\mathcal{C}_i^*$.

In order to increase the speed and accuracy of pose refinement, we remove the points in $\mathcal{V}_{\text{model}}^*$ which are not visible according to the primary pose P_i^* with respect to the camera. We use the algorithm in [17] for this purpose:

$$\mathcal{V}_i^{\text{visible}} = \text{ExtractVsibilePoints}(\mathcal{V}_{\text{model}}^*, t_i^{\text{cam}}) \qquad (30)$$

where t_i^{cam} is the relative position of the camera with respect to the model:

$$\begin{bmatrix} R_i^{\text{cam}} & t_i^{\text{cam}} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}_{4 \times 4} = T_i^* \qquad (31)$$

$$T_i^* = (P_i^*)^{-1} \qquad (32)$$

To increase the registration speed we also remove the parts of the pointcloud $\mathcal{C}_i^*$ that are not close to the model according to the primary pose:

$$\mathcal{C}_i^{\text{nei}} =$$
$$\left\{ p \in \mathcal{C}_i^* \left| \exists p' \in \mathcal{V}_{\text{model}}^* : \left\| T_i^* \begin{bmatrix} p \\ 1 \end{bmatrix} - \begin{bmatrix} p' \\ 1 \end{bmatrix} \right\|_2 < d_{\text{nei}} \right. \right\} \qquad (33)$$

Here $\mathcal{C}_i^{\text{nei}}$ is the cloud that contains the neighborhood of the patient's body in the primary pose in $\mathcal{C}_i^*$ and $d_{\text{nei}} \in R^+$ is the distance we use to determine it.

Finally, we can refine the pose of the patient using the

8

Figure 4: Steps of geometric alignment in our tracking algorithm. Please note that merging model vertices happens only one time before tracking starts. The rest of the steps need to be repeated for individual frames.

Parameter	Value	Description
l	10	Constant for skipping frames in marker mapper and temporary model reconstruction.
N	5×10^4	Constant for dynamic subsampling
d_0	1 cm	Minimum voxel size for dynamic subsampling
$d_{\mathfrak{C}}$	1 cm	Downsampling voxel size for body reconstruction
d_*	2 cm[1], 1 cm[2]	Downsampling voxel size for depth point cloud in patient tracking
d_{nei}	10 cm[1], 4 cm[2]	Neighborhood extraction maximum distance in patient tracking

[1] Used with the Xtion Pro Live dataset
[2] Used with the Realsense L515 dataset

Table 2: List of values we used for different constants in our algorithm for quantitative evaluation.

point cloud $\mathcal{C}_i^{\mathrm{nei}}$:

$$\hat{T}_i^* = \mathrm{GeneralizedICP}(\mathcal{C}_i^{\mathrm{nei}}, \mathcal{V}_i^{\mathrm{visible}}, T_i^*) \qquad (34)$$

Now we can assign the final pose to be used for visualization by:

$$\hat{P}_i^* = (\hat{T}_i^*)^{-1} \qquad (35)$$

At the end to calculate the final pose error we can write:

$$P_i^{\mathrm{adj}} = \hat{P}_i^* \left(P_i^{U*} \ P_U^{\mathrm{ref}} \right)^{-1} \qquad (36)$$

where P_U^{ref} is the reference pose of the patient with respect to the UcoSLAM map that we assume is given by the user through an interface, and P_i^{adj} is the transformation that can be used to give feedback to the person about the error in positioning. In the scenario of patient positioning P_U^{ref} is determined in the planning phase of radiation therapy.

4. Experimental Results

This section aims at evaluating the validity of the proposed method, both qualitatively and quantitatively, for patient positioning.

The quantitative evaluation has been done using a mannequin as the patient. The tracking accuracy of the proposed method was measured with a motion capture system along with the running speed of our implementation. In the qualitative evaluation, human subjects are employed and the output of our algorithm is demonstrated for visual inspection.

This section presents first the implementation details of our algorithm, including the hardware and software libraries employed, as well as the values of the employed parameters. Then, we demonstrate numerical results related to our quantitative evaluation. Finally, the qualitative results are presented using snapshots of our program's video output.

4.1. Implementation Details

We tested our implementation on a laptop with Intel® Core™ i7-4700HQ running the Ubuntu 18.4 operating system. To capture RGB-D images we used the Asus Xtion Pro Live and the Intel Realsense L515 sensors. Both the depth and RGB images of the Xtion sensor were set to the resolution of 640×480 pixels. The depth image of

L515 was set to the 640×480 resolution however the color had the resolution of 1280×720. The depth cameras were manually calibrated with respect to the color cameras and depth images were registered to the color camera coordinate system.

Our algorithm was implemented in C++ and the 3D visualization was developed using the Qt3D[6] V5.9 library. For general image processing, we took advantage of OpenCV[7] V3. To perform point cloud and mesh processing, including the point-to-plane ICP we employed the Open3D[8] V0.9.0 library. We also took advantage of the original implementation of Generalized ICP which is publicly available online[9].

We also employed publicly available implementations of UcoSLAM[10] V1.0.8, marker mapper[11] V1.0.15, and ArUCO[12] V3.1.11. Finally, the KinectFusion algorithm [27] has been employed to create the 3D model reference of the mannequin employed in our tests.

Along with the paper, we have employed several parameters for our algorithm. Their concrete values employed in our quantitative experimentation are indicated in Table 2.

4.2. Quantitative Evaluation

The quantitative evaluation has been carried out using a mannequin (see Fig. 5(a)) with infrared reflective dots attached. The dots are tracked using an OptiTrack[13] motion capture system comprised by a total of six infrared synchronized cameras that achieves sub-millimeter precision in the estimation of the dot positions. The reflective markers are used to determine the ground truth poses (rotation and translation) of the mannequin along the test sequences recorded. We also placed ArUco markers on the floor for accurate scene tracking using UcoSLAM, which combines ArUco markers, image features, and depth to track the camera pose in the environment.

We captured two datasets, one with the Xtion Pro Live sensor and another one using the Realsense L515 sensor. For each dataset, we recorded a first RGB-D video sequence of the mannequin to obtain the reference 3D model using the KinectFusion algorithm, and another sequence of the environment without the mannequin to create the UcoSLAM environment map. These sequences are the equivalent of our planning phase.

Then, 6 sequences were recorded for evaluation purposes, where the mannequin is moved and rotated in different poses around the target position. Each sequence lasts several seconds and the total amount of frames from all sequences is 3006 for each type of sensor that was employed. To simulate a real scenario and to properly analyze the system accuracy, the sequences were recorded with the camera and the mannequin in different relative poses and moving them locally during the video sequences. We tried

---

[6]https://wiki.qt.io/Qt3D
[7]https://opencv.org/
[8]http://www.open3d.org/
[9]https://github.com/avsegal/gicp
[10]https://sourceforge.net/projects/ucoslam/
[11]https://sourceforge.net/projects/markermapper/
[12]https://sourceforge.net/projects/aruco/
[13]https://www.optitrack.com/

Sequence #	Asus Xtion Pro Live		Intel Realsense L515	
	MRE	MTE	MRE	MTE
1	1.77°	6.87 mm	0.82°	4.59 mm
2	1.45°	5.27 mm	0.78°	3.45 mm
3	2.19°	8.67 mm	1.05°	4.89 mm
4	1.81°	7.51 mm	0.55°	4.02 mm
5	1.65°	6.94 mm	0.87°	3.23 mm
6	1.73°	8.41 mm	0.86°	4.64 mm
All Evaluation Frames	1.77°	7.28 mm	0.82°	4.17 mm

Table 3: Mean rotational error (MRE) and mean translational error (MTE) of our algorithm applied on the datasets captured by the Asus Xtion Pro Live and Intel Realsense L515 sensors. The evaluation is done by comparing the output to the ground truth from the motion capture system.

to keep enough amount of the background (scene) in the captured images to make sure the UcoSLAM scene tracking is performed correctly.

In order to estimate the relationship between the motion capture system reference system and ours, we split each tracking sequence into two halves. The first half was used to estimate the essential transformations needed for evaluation and the second half was employed for calculating the errors with respect to the ground truth.

We computed the error both in translation and rotation, and the results can be seen in Table 3 for the datasets related to each type of sensor. We evaluated the mean error for each of our 6 tracking sequences. We also calculated these error values on the collection of all evaluation frames from all sequences (1503 frames for each sensor), the result of which can be observed in the last row of the table. Furthermore, we calculated the overall median errors for both of the datasets as $6.71 \, \text{mm}/1.53°$ for the Xtion Pro Live sensor and $3.83 \, \text{mm}/0.77°$ for the Realsense L515 sensor.

We have also evaluated the running speed of our implementation. We calculated the average running time in the form of frames-per-second (fps) for all frames of all tracking sequences used for evaluation. To do so, for each frame, we measured the time lapsed since the previous pose estimation until the current pose estimation. The mean running speed of the patient tracking was 19 fps for the Xtion sensor and 9 fps for the Realsense L515 sensor

Before being able to run the program for patient tracking it is needed to create a marker map of the patient and align it to the 3D model. This requires a few steps that we call the preparation steps. We have summarized the running time for those steps that take a significant time, in Table 4 using our reconstruction sequences.

4.3. Qualitative Evaluation

For qualitative results, we have applied our algorithm to several sequences on human subjects using the Xtion Pro Live sensor and also on the mannequin we have used in our quantitative evaluation.

We employed 3 different human subjects, one woman and two men presented in Figure 6. One man (middle two rows) has no apparel on his torso and the ArUco markers

Figure 5: A demonstration of the setup for our quantitative experiment is presented (a). Infrared reflective markers were tracked by the motion capture system for estimating the ground truth pose. The RGB-D camera was employed to capture the test video sequence. We took advantage of the Realsense L515 (b) and the Xtion Pro Live (c) RGB-D cameras in our evaluation.

Preparation Step	Asus Xtion Pro Live	Intel Realsense L515
Map Creation: UcoSLAM + Marker Map	37 s	62 s
Geometrical Reconstruction	37 s	68 s
Mesh Creation	2 s	2 s
Interactive Reference Model Alignment	10 s	31 s
Total	86 s	163 s

Table 4: The time spent by each part of the algorithm to create the marker map and UcoSLAM map in addition to finding the transformation form the marker map to the 3D model coordinate system.

are attached on his skin. The second man (last two rows) has tight-fitting clothing on his torso and the markers are attached on the clothing. For the woman (first two rows), the upper torso is clothed and the lower half of the torso is exposed with most of the markers attached to this part. In the figure, the red transparent model shows their tracked pose and the blue solid model shows their desired pose.

Finally, the snapshots related to the mannequin's qualitative evaluation are presented in Figure 7. The color codes are the same as the ones in Figure 6 and similarly there is a numerical feedback of the positioning error.

5. Discussion

5.1. Quantitative Results

As it can be viewed in Table 3 we where able to reach the average error value of $7.28\,\text{mm}/1.77°$ using the Xtion Pro Live sensor and $4.17\,\text{mm}/0.82°$ employing the Realsense L515 sensor. We suspect the main reason that we get a higher accuracy using the Realsense sensor is that it has a higher color image resolution. The higher resolution leads to a better ArUco marker detection and pose estimation which are employed both for scene tracking and patient tracking. It also helps with more robust detection of image features which is used by UcoSLAM for scene tracking.

Additionally the overall median errors ($6.71\,\text{mm}/1.53°$ for the Xtion Pro Live sensor and $3.83\,\text{mm}/0.77°$ for the

Realsense L515 sensor) are significantly smaller than the average error which means that most of the errors come from the minority of frames. This suggests the general robustness of our method in pose estimation.

In regards to the tracking speed, the average frame rate of our implementation was 19 fps for the Xtion Pro Live sensor and 9 fps for the Realsense L515 sensor. We would like to suggest that the reason the slower frame rate for the Realsense sensor is because of its higher resolution color image (1280×720). The higher resolution causes slower marker and image feature detection that affects both the scene tracking and the patient tracking speed. It should be reminded that this frame-rate was measured on a relatively old laptop. Hence we believe that our algorithm has the potential for real-time operation using up-to-date hardware.

Regarding the running time of the preparation steps, as can be observed in Table 4, we were able to prepare the program to start tracking in under 2 minutes for the Xtion Pro live sensor and under 3 minutes using the Realsense L515 sensor. Furthermore, it can be seen that the semi-automatic reference model alignment can be done relatively fast and the majority of the time needed by the preparation steps is spent on body model's map creation and its geometrical reconstruction.

11

Figure 6: An example of the result from our tracking algorithm for patient positioning for multiple frames in different sequences for three human subjects. The red mesh shows the currently tracked patient pose, the blue mesh shows the pose where the patient needs to be positioned. There is a feedback of the pose error in rotation and translation in the corner of each frame.

Figure 7: Qualitative results using the same mannequin employed in quantitative evaluation. The red mesh shows the current pose of the mannequin, and the blue mesh shows the desired pose. Rotational and translation error feedback is visualized in the corner of the images.

5.2. Qualitative Results

As you can see in Figure 6 there are reliable pose estimations for all of the subjects. In some cases, you might notice small mismatches in the tracked torso (red model) and the actual current pose of the patient. We believe this is because of the non-rigid deformations of the upper body. Since we are not using any apparatus to fix the upper body it can slightly deform and cause a small error in registration. As said before, this can be improved by employing a fixing apparatus.

In Figure 6 you can also view our two ways of visual feedback. First, the model meshes related to the current pose and the target pose of the patient. Second, the numerical feedback on the bottom corner of the image showing the positioning error in rotation and translation. As you can see the intersection of the two model meshes can clearly show the operator how close the current pose of the patient matches that of the target pose. Also, when the patient pose is close enough to the target pose the operator can correct it furthermore by looking at the numerical position error visualized on the corner of the image.

Finally regarding the qualitative results of the mannequin, as can be observed in Figure 7, tracking is done with high accuracy and the tracked model matches almost perfectly with the mannequin. We believe that since there are no non-rigid movements here unlike the human subject and there is a smaller room for error.

6. Conclusion and future work

This paper has proposed a cost-effective and efficient method for interactive AR in patient positioning that can be used on mobile devices (such as Head Mounted Displays) requiring a only a RGBD camera and a regular CPU. Our method combines 3D information with texture information (keypoints) and artificial markers in order to create a map of the environment and a 3D model of the patient. In the planning phase, a map of the treatment room is created and the 3D model of the patient is obtained. Then, for each treatment session, a temporary body model is created at the beginning of the session, which is employed for tracking. Our system is able to simultaneously track the camera pose in the treatment room as well as the subject's body, providing visual information about the target position required for treatment.

The conducted experiments prove that our proposal achieves a high accuracy with the mean error of $4.17\,\text{mm}/0.82°$ and the median error of $3.83\,\text{mm}/0.77°$. Also, the proposed method has proved to obtain a relatively high frame rate (9 fps) without the need to use a dedicated GPU nor a very powerful computer. It is then a cost-effective method that can be even employed in hospitals with a limited budget. Our qualitative results also showed the usefulness of the algorithm to be employed with the AR interface and how it can help the operator for patient positioning.

We still think that there is room for improvement. Our algorithm only performs rigid tracking similar to some other industrial level solutions. Adding non-rigid tracking the algorithm can make our approach even more capable for example in tracking the respiration movements of the patient. Another interesting future work is to accounting for local deviation of body parts when registering the model, since it could improve the system precision. We also think the application of our algorithm in other medical areas such as medical education can be explored further.

We also consider that it will be necessary in the future to conduct a study to obtain feedback from medical doctors on how our solution can be improved to be applied in real life situation.

Finally, we would like to mention that we have opened the way for other researchers to work on mobile interactive AR to assist patient positioning. We have laid down a

framework that could be improved or build upon by others to have even better affordable patient positioning solutions that take advantage of AR.

Conflict of interest

The authors do not have financial and personal relationships with other people or organizations that could inappropriately influence (bias) their work.

Acknowledgment

References

[1] NICE Develops Medtech Innovation Briefing on AlignRT in Breast Cancer Radiotherapy. *ESMO.org*, September 2018.

[2] Inés Barbero-García, José Luis Lerma, and Gaspar Mora-Navarro. Fully automatic smartphone-based photogrammetric 3D modelling of infant's heads for cranial deformation analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:268–277, August 2020.

[3] S. Bauer, J. Wasza, S. Haase, N. Marosi, and J. Hornegger. Multi-modal surface registration for markerless initial patient setup in radiation therapy using microsoft's Kinect sensor. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1175–1181, November 2011.

[4] João Cartucho, David Shapira, Hutan Ashrafian, and Stamatia Giannarou. Multimodal mixed reality visualisation for intraoperative surgical guidance. *International Journal of Computer Assisted Radiology and Surgery*, 15(5):819–826, May 2020.

[5] Changhyun Choi and Henrik I Christensen. Robust 3D visual tracking using particle filtering on the special Euclidean group: A combined approach of keypoint and edge features. *The International Journal of Robotics Research*, 31(4):498–519, April 2012. Publisher: SAGE Publications Ltd STM.

[6] Wongun Choi, Caroline Pantofaru, and Silvio Savarese. A General Framework for Tracking Multiple People from a Moving Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1577–1591, July 2013. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[7] Andrew Comport, Eric Marchand, Muriel Pressigout, and François Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):615–628, 2006.

[8] Francesco Cosentino, Nigel W. John, and Jaap Vaarkamp. RAD-AR: RADiotherapy - Augmented Reality. In *2017 International Conference on Cyberworlds (CW)*, pages 226–228, September 2017.

[9] Martin Eckert, Julia S. Volmerg, and Christoph M. Friedrich. Augmented Reality in Medicine: Systematic and Bibliographic Review. *JMIR mHealth and uHealth*, 7(4):e10967, 2019. Company: JMIR mHealth and uHealth Distributor: JMIR mHealth and uHealth Institution: JMIR mHealth and uHealth Label: JMIR mHealth and uHealth Publisher: JMIR Publications Inc., Toronto, Canada.

[10] Omid Ehsani, M. Pouladian, S. Toosizadeh, and A. Aledavood. Registration and fusion of 3D surface data from CT and ToF camera for position verification in radiotherapy. *SN Applied Sciences*, 1(11):1347, October 2019.

[11] Samuel Bednar French. *An augmented reality based patient positioning system with applications to breast radiotherapy*. 2014. Accepted: 2016-04-05T19:35:04Z Publication Title: Dissertations & Theses @ SUNY Buffalo,ProQuest Dissertations & Theses Global.

[12] C. Freschi, S. Parrini, N. Dinelli, M. Ferrari, and V. Ferrari. Hybrid simulation using mixed reality for interventional ultrasound imaging training. *International Journal of Computer Assisted Radiology and Surgery*, 10(7):1109–1115, July 2015.

[13] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer. Generation of fiducial marker dictionaries using Mixed Integer Linear Programming. *Pattern Recognition*, 51:481–491, March 2016.

[14] Jeremy D. P. Hoisak and Todd Pawlicki. The Role of Optical Surface Imaging Systems in Radiation Therapy. *Seminars in Radiation Oncology*, 28(3):185–193, July 2018.

[15] Jan Issac, Manuel Wüthrich, Cristina Garcia Cifuentes, Jeannette Bohg, Sebastian Trimpe, and Stefan Schaal. Depth-based object tracking using a Robust Gaussian Filter. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 608–615, May 2016.

[16] Allan Javaux, David Bouget, Caspar Gruijthuijsen, Danail Stoyanov, Tom Vercauteren, Sebastien Ourselin, Jan Deprest, Kathleen Denis, and Emmanuel Vander Poorten. A mixed-reality surgical trainer with comprehensive sensing for fetal laser minimally invasive surgery. *International Journal of Computer Assisted Radiology and Surgery*, 13(12):1949–1957, December 2018.

[17] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, pages 24–es, New York, NY, USA, July 2007. Association for Computing Machinery.

[18] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 61–70, Cagliari, Sardinia, Italy, June 2006. Eurographics Association.

[19] Sing Chun Lee, Bernhard Fuerst, Javad Fotouhi, Marius Fischer, Greg Osgood, and Nassir Navab. Calibration of RGBD camera and cone-beam CT for 3D intra-operative mixed reality visualization. *International Journal of Computer Assisted Radiology and Surgery*, 11(6):967–975, June 2016.

[20] Javier A. Luzon, Bojan V. Stimec, Arne O. Bakka, Bjørn Edwin, and Dejan Ignjatovic. Value of the surgeon's sightline on hologram registration and targeting in mixed reality. *International Journal of Computer Assisted Radiology and Surgery*, 15(12):2027–2039, December 2020.

[21] Meng Ma, Pascal Fallavollita, Ina Seelbach, Anna Maria Von Der Heide, Ekkehard Euler, Jens Waschke, and Nassir Navab. Personalized augmented reality for anatomy education. *Clinical Anatomy*, 29(4):446–453, 2016.

[22] Márcio Cerqueira de Farias Macedo, Antônio Lopes Apolinário Júnior, Antonio Carlos dos Santos Souza, and Gilson Antônio Giraldi. High-Quality On-Patient Medical Data Visualization in a Markerless Augmented Reality Environment. *SBC Journal on Interactive Systems*, 5(3):41–52, December 2014. Number: 3.

[23] Wayne L. Monsky, Ryan James, and Stephen S. Seslar. Virtual and Augmented Reality Applications in Medicine and Surgery-The Fantastic Voyage is here. *Anatomy & Physiology: Current Research*, 9(1):1–5, February 2019. Publisher: Longdom Publishing S.L.

[24] Frank Moosmann and Christoph Stiller. Joint self-localization and tracking of generic objects in 3D range data. In *2013 IEEE International Conference on Robotics and Automation*, pages 1146–1152, May 2013. ISSN: 1050-4729.

[25] Rafael Muñoz-Salinas, Manuel J. Marín-Jimenez, Enrique Yeguas-Bolivar, and R. Medina-Carnicer. Mapping and localization from planar markers. *Pattern Recognition*, 73:158–171, January 2018.

[26] Rafael Muñoz-Salinas and R. Medina-Carnicer. Ucoslam: Simultaneous localization and mapping by fusion of keypoints and squared planar markers. *Pattern Recognition*, page 107193, 2020.

[27] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, October 2011.

[28] Daniel Nguyen, Jad Farah, Nicolas Barbet, and Mustapha Khodri. Commissioning and performance testing of the first prototype of AlignRT InBore™ a Halcyon™ and Ethos™-dedicated surface guided radiation therapy platform. *Physica Medica*, 80:159–166, December 2020.

[29] Steven A. Parkison, Lu Gan, Maani Ghaffari Jadidi, and Ryan M. Eustice. Semantic Iterative Closest Point through Expectation-Maximization. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 280. BMVA Press, 2018.

[30] Egidijus Pelanis, Rahul P. Kumar, Davit L. Aghayan, Rafael Palomar, Åsmund A. Fretland, Henrik Brun, Ole Jakob Elle, and Bjørn Edwin. Use of mixed reality for improved spatial understanding of liver anatomy. *Minimally Invasive Therapy & Allied Technologies*, 29(3):154–160, June 2020. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/13645706.2019.1616558.

[31] Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38–47, August 2018.

[32] Martin Runz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20, October 2018. ISSN: 1554-7868.

[33] Hamid Sarmadi, Rafael Muñoz-Salinas, M. Álvaro Berbís, Antonio Luna, and R. Medina-Carnicer. 3D Reconstruction and alignment by consumer RGB-D sensors and fiducial planar markers for patient positioning in radiation therapy. *Computer Methods and Programs in Biomedicine*, 180:105004, October 2019.

[34] Aleksandr V. Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. 2009.

[35] F. Stieler, F. Wenz, D. Scherrer, M. Bernhardt, and F. Lohr. Clinical evaluation of a commercial surface-imaging system for patient positioning in radiotherapy. *Strahlentherapie und Onkologie*, 188(12):1080–1084, December 2012.

[36] Michael Strecke and Jörg Stückler. EM-Fusion: Dynamic Object-Level SLAM with Probabilistic Data Association. *arXiv:1904.11781 [cs]*, April 2019. arXiv: 1904.11781.

[37] J. Talbot, J. Meyer, R. Watts, and R. Grasset. A patient position guidance system in radiation therapy using augmented reality. In *2008 23rd International Conference Image and Vision Computing New Zealand*, pages 1–5, November 2008. ISSN: 2151-2205.

[38] J. Talbot, J. Meyer, R. Watts, and R. Grasset. A method for patient set-up guidance in radiotherapy using augmented reality. *Australasian Physical & Engineering Sciences in Medicine*, 32(4):203–211, December 2009.

[39] Kevin S. Tang, Derrick L. Cheng, Eric Mi, and Paul B. Greenberg. Augmented reality in medical education: a systematic review. *Canadian Medical Education Journal*, 11(1):e81–e96, March 2020.

[40] Souriya Trinh, Fabien Spindler, Eric Marchand, and François Chaumette. A modular framework for model-based visual tracking using edge, texture and depth features. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 89–96, October 2018. ISSN: 2153-0866.

[41] Chi-Yi Tsai, Kuang-Jui Hsu, and Humaira Nisar. Efficient Model-Based Object Pose Estimation Based on Multi-Template Tracking and PnP Algorithms. *Algorithms*, 11(8):122, August 2018. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

[42] Xiang Wang, Séverine Habert, Meng Ma, Chun-Hao Huang, Pascal Fallavollita, and Nassir Navab. [POSTER] RGB-D/C-arm Calibration and Application in Medical Augmented Reality. In *2015 IEEE International Symposium on Mixed and Augmented Reality*, pages 100–103, September 2015.

[43] Manfred Wiencierz, Kathrin Kruppa, and Lutz Lüdemann. Clinical Validation of two Surface Imaging Systems for Patient Positioning in Percutaneous Radiotherapy. *arXiv:1602.03749 [physics]*, February 2016. arXiv: 1602.03749.