

High-fidelity visuo-haptic interaction with virtual objects in multi-modal AR systems

Gerald Bianchi*
Computer Vision Lab
ETH Zurich

Christoph Jung
Kaiserslautern
University of Technology

Benjamin Knoerlein †
Computer Vision Lab
ETH Zurich

Gábor Székely ‡
Computer Vision Lab
ETH Zurich

Matthias Harders §
Computer Vision Lab
ETH Zurich

ABSTRACT

The driving force of our research is the precise combination of real and - possibly indistinguishable - virtual objects in an interactive augmented reality environment. This requires real-time, multimodal simulation, as well as stable and accurate overlay of the computer-generated objects. This paper describes several methods to improve accuracy and stability of our hybrid augmented reality system. In a comparison of two approaches to hybrid head pose refinement, we show that Quasi-Newton method enables high performance optimization for image space error minimization. Moreover, a 3D landmark refinement step is proposed, which significantly improves quality and robustness of the overlay process. The enhanced system is demonstrated in an interactive AR environment, which provides accurate haptic feedback from real and virtual deformable objects. Finally, the effect of landmark occlusion on tracking stability during user interaction is also analyzed.

Keywords: hybrid augmented reality, image and object space error, vision-based refinement, haptic interaction

1 INTRODUCTION

With the recent progress in modeling haptic feedback, several attempts for comparison of manual interaction with real and virtually generated objects have been proposed. Studies focusing on discrimination of stiff [11] and deformable objects [14], as well as texture [22] have been performed.

All these endeavors allow the examination of only a single perceptual channel, since solely haptic feedback is generated and the experimental mechanism has to be hidden from the user's view. Realistically comparing the behaviour of real and virtual objects is, however, only possible if visual feedback is also provided during the interaction. In this paper we describe a prototype, which allows the comparison of simple virtual and real objects in an augmented reality (AR) environment.

In order to create a sufficiently realistic environment, high stability and robustness as well as low latency and computation time are of primary importance. Research has shown that AR system fidelity can be improved by combining several tracking technologies [4][21][25]. Therefore, we have developed a hybrid AR setup, which uses an external optical tracker to provide an initial guess for the user's head pose and a subsequent visual landmark-based refinement of the pose estimate. We compare two approaches for the head pose correction step according to image and object space error definitions. Furthermore, we integrate 3D landmark based refinement

into the pose estimation in order to further improve the quality and robustness of the overlay process. Since the user is interacting with the augmented objects, visual marks are often obscured. Therefore, landmark occlusion handling also had to be addressed. By combining all these methods, we could achieve accurate and stable hybrid head tracking. Combining the system with the haptic simulation, we enabled users to simultaneously touch and observe real and virtual deformable objects.

The paper is organized as follows: After a review of related work, Section 3 describes our AR system. Next, the hybrid tracker is detailed, including landmark detection as well as occlusion handling. Thereafter, a performance comparison study of the head pose estimation is presented. Two approaches of computing registration errors - in image vs. object space - are discussed. In addition, for both approaches a number of optimizers are investigated with regard to real-time performance. In the next section, the influence of 3D landmark detection accuracy on head pose jitter is examined. Finally, the last section illustrates the performance of the hybrid tracker in the haptic AR application, simulating interaction with virtual deformable objects embedded into a real environment.

2 RELATED WORK

In the last decade several hybrid AR systems have been suggested in the literature. The proposed methods primarily differ with regard to the type of sensors and sensor data utilized. Most systems make use of a vision-based tracker and compensate its shortcomings with another tracking technology. A typical approach is to provide an estimation of the head pose to the vision-based system.

State's work [21] combines a vision-based registration with a magnetic sensor. The latter reports the position of the user's head while the machine vision system uses these data as an initial guess for estimating the head pose. Based on visual tracking of color landmarks placed at known locations in the working area, the model-based vision refines this estimate. Similar to this, in [3] the authors track the corners of black and white rectangles for head pose refinement. An extension to natural features has been suggested in [25], which combines an inertial sensor and a vision-based tracking system.

An alternative to these methods is to only use the additional tracking sensor when the first one fails. In [1], an image-based system is coupled with an inertial sensor. The vision system relies on tracking point correspondences lying on planar surfaces. From these points, a planar homography between two consecutive views is computed. In cases when the image-based tracker fails due to large rotations or abrupt movements, the inertial sensor takes over. A criterion based on corresponding point matching is defined in order to control the switch between both tracking systems.

Combining the sensor data by fusion has also been suggested. From two uncorrelated measurements of the same target, the pose of a target object is computed as a linear combination of both data weighted by the uncertainty of the sensor. In [9], a hip implant pose is first evaluated by a fixed optical sensor. In addition, a second measurement is obtained by a vision-based approach. Given the

*e-mail: bianchi@vision.ee.ethz.ch

†e-mail: knoerlein@vision.ee.ethz.ch

‡e-mail: szekely@vision.ee.ethz.ch

§e-mail: mharders@vision.ee.ethz.ch

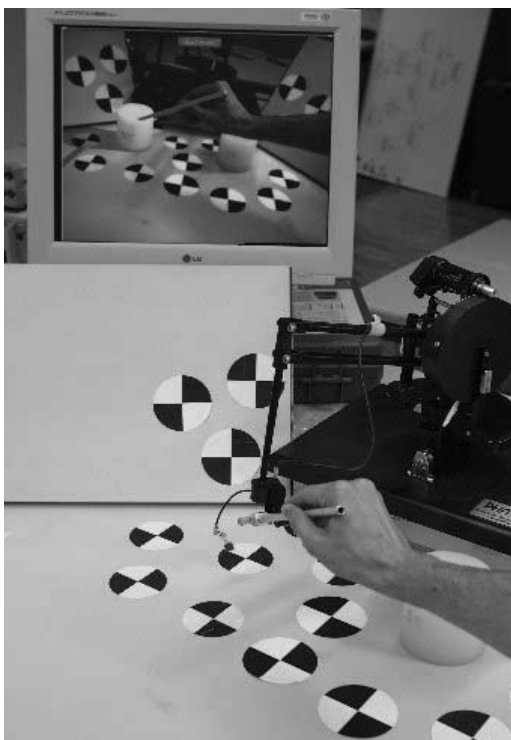


Figure 1: View depicting all components of our AR system

uncertainty of both methods, the hip pose is obtained by fusing the resulting data.

In [18], the authors use the same principle in order to automatically initialize a model-based tracking system. The AR setup consists of a fixed and a mobile camera. The former reports an approximation of the translation of the user's head by tracking an attached passive landmark, while the latter provides the image of the tracked model. Similar to the previous approach, pose refinement and coarse pose estimation are also fused.

3 AUGMENTED REALITY SETUP

Our AR system comprises an infra-red (IR) optical 6DoF tracking device OPTOTRAK 3020 manufactured by Northern Digital Inc. and a head-mounted FireWire camera. We use a Videre Design MEGA-DCS camera with 7.5mm focal length and 640x480 pixel image size. The optical tracker consists of three fixed linear cameras which detect the infrared LEDs attached to a marker. The latter is attached to the head-mounted camera for head tracking purposes. By triangulation, the optical system measures the 3D LED position with a RMS accuracy of 0.2mm at an optimal distance of 2.25m. From these measurements, the orientation and position of the marker are computed. Since the camera and the marker are rigidly attached, the camera-marker transformation is fixed and estimated by Hand-Eye calibration [24]. Given this transformation and the marker pose, the AR system can compute the camera pose with respect to the optical tracker coordinate frame. The estimated camera pose allows us to obtain a first alignment between the virtual and the real world.

To allow a user to touch virtual objects, a SensAble PHANToM 1.5 haptic device is integrated into our AR setup. Previously, we developed [5] a haptic calibration procedure in order to align the coordinate systems of the haptic and the IR optical tracking device. The method is described in more detail below. Figure 1 illustrates the integration of the haptic device into our AR setup.

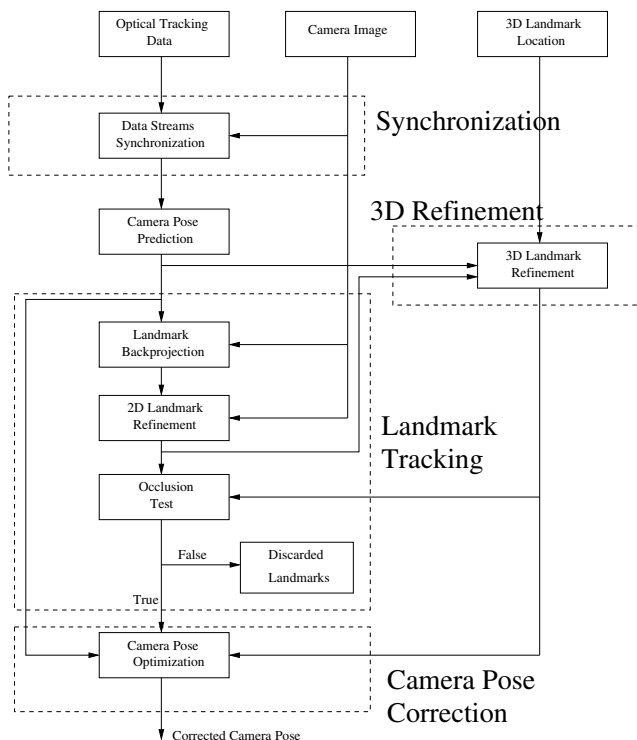


Figure 2: Pipeline of the Hybrid tracking system

4 HYBRID AR SYSTEM

The tracking data reported by the IR optical tracker are inherently noisy due to the inaccurate measurements of the LEDs' position, which leads to errors in the estimated camera pose. The precision of the measurements becomes even more limited when the head-mounted marker moves. As a consequence, the registration between the real and the virtual world is affected, which is revealed by the instability of virtual objects in the augmented images. To overcome the problem of noisy data, we combine the IR optical tracker with a vision-based tracking approach. The main idea is to correct the estimated camera pose by using image information. In this section, we provide details about our hybrid AR approach.

4.1 Principle

Our system is inspired by State's work as presented in [21]. The main idea of the hybrid AR system is the assistance of the vision-based tracker by providing an estimation of the camera pose. Compared to the previous systems based on a stereoscopic setup, our approach only uses information from a monoscopic camera to refine its pose. Moreover, the method does not require color-based tracking to identify the necessary landmarks and to properly handle occlusion. We rely on corners as primary features and a model-based landmark analysis to reliably detect occlusions. In contrast to the earlier systems [21][3], our landmark calibration does not involve any auxiliary tracker or localizer to measure the 3D positions of the corners with respect to the world coordinate system. We also apply a vision-based correction in order to refine the landmark positions.

Figure 2 illustrates the pipeline of our hybrid setup. A set of artificial landmarks is placed in the workspace. Their locations are measured with respect to the optical tracker - referred to as the **world coordinate system**. Using the predicted camera pose, the 3D landmarks are backprojected onto the current camera image. Due

to the noisy camera pose, the projected landmarks do not exactly match the observed ones in the image. Therefore, a 2D refinement is performed in order to precisely determine the center of the landmarks. Moreover, to ensure that a refined landmark is sufficiently visible, an occlusion test is carried out based on a similarity measure described below. If the test fails, the landmark is discarded from the camera pose correction. Given the 2D-3D correspondence of the remaining landmarks, the position and orientation of the camera are then refined by error minimization. Since each data stream is obtained from different processes, a synchronization step is necessary to obtain the right camera pose for the current frame. In addition, an image-based refinement of the 3D landmark locations is performed in order to reduce measurement errors and to stabilize the corrected camera pose. Throughout this section, we will describe the three major subsystems of the hybrid tracking system: synchronization, landmark tracking, and camera pose correction.

4.2 Soft Synchronization

Two major approaches have been proposed in the literature in order to synchronize data streams. The first one involves hardware-based or hard synchronization. An external signal triggers the different devices, which send the data to the AR system sampled at the same time. As a result, the relative latency is negligible.

However, not all devices are equipped with a trigger mechanism due to the necessary increase of system complexity and higher costs. Another drawback would also be the use of additional cables limiting user mobility and thus making the system more cumbersome. Alternatively, a second approach can be followed. In this case, the relative latency between two input streams is reduced by soft synchronization. Appropriate synchronization schemes can considerably reduce misregistration without any additional hardware [10]. Our software synchronization is based on the latter approach by matching the closest time stamp of tracking samples to that of camera images. Since the update rate of the tracking system (120Hz) is higher than the frame rate of the camera (30Hz), we receive several tracking samples during a single image acquisition. In order to synchronize the different simultaneously running processes, tracking samples and corresponding time stamps are stored in an N-element list. When the rendering process needs to overlay the computer-generated object on the given image, we select the best tracking sample to compute the camera pose by minimizing the difference between the time stamps of the image and the tracking sample. As a result, the relative latency between both data streams is minimized, thus leading to a better prediction of the 2D landmark positions.

4.3 Landmark Tracking Subsystem

4.3.1 Shape and Color of Landmarks

The landmark tracking has to be fast and robust. The first step is the choice of the type of features. We use corner detection, since it can reach subpixel accuracy and the tracking can be sped up by providing an initial guess of the 2D position. The latter is obtained by backprojecting the 3D landmarks onto the images. Figure 3 illustrates the shape of the landmark. The center of the pattern is characterized by the intersection of four areas. Black and white colors produce a high contrast in the images, resulting in rapid landmark recognition. In addition, we use four points located on the border of each area called *satellite points* to obtain a priori knowledge of the landmark shape based on two triangles represented by dashed lines in figure 3. These triangles are used for the occlusion test as described below.

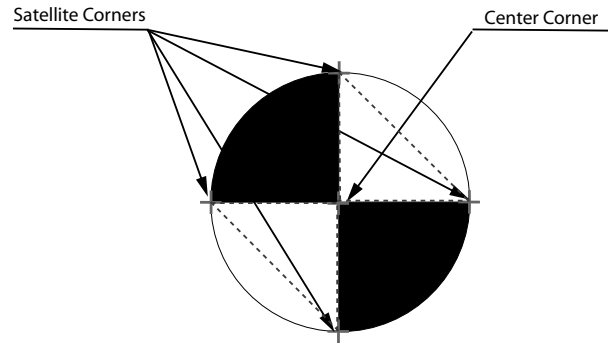


Figure 3: Artificial landmarks used by the vision-based tracking system

4.3.2 Calibration

To predict the landmark location in the images, we first have to measure their 3D location with respect to the world coordinate frame. To this end, we use a calibrated pointer. Attached to the pointer, a marker is tracked by the optical tracking device. Pivot calibration [12] is carried out in order to estimate the position of the pointer tip with respect to the marker frame. The tip is fixed at the visual landmark and the pointer-mounted marker is displaced in a spherical movement. Using the recorded marker positions, a sphere fitting the data can be determined. The position of the tip in the marker coordinate frame can then be obtained from center and radius of the sphere. It should be noted, that the estimation of the tip-marker relationship depends on the accuracy of the point measurements.

Two main sources of measurement errors can be identified: marker pose accuracy, and the stability of the fixed tip in the workspace while rotating the marker. To determine the influence of the point measurement errors on the calibration, we performed a simulation of the procedure. We use the statistical noise model described by our previous work [6] for the former error source. For the latter we used Gaussian noise to shift the tip position from the center of the sphere in order to model the instability of the pointer tip location while rotating the marker.

The simulations consisted of varying the magnitude of the introduced errors within 0-2mm for the marker noise and 0-10mm for the tip position noise. The results revealed that the instability of pointer tip position dominates the calibration error. For instance, given 1mm marker error, the pointer position error is around 0.5mm. Equivalent amount of errors for the tip position provides 1mm error. In practice, we also observed that the stylus moves about 1 – 2mm. Therefore, based on the simulations we can assume that the accuracy of the pointer calibration is within 1 – 2mm. As a consequence, the center of landmarks and the *satellite points* locations can be assumed to be measured in the working space within 2mm accuracy.

4.3.3 Corner Detection

Given the 3D landmark position and the predicted camera pose, the location of the image center of the landmark can be estimated. By backprojecting the 3D landmarks onto the image, we obtain a first initial guess of the position of the features. Around each backprojected 2D location, a square window delimits the search area in which the corner detection is performed. In our case, we set the window size 32x32 pixels in order to ensure that the search area is always maintained within the landmark image. This is possible because in our application, the distance between the user and the landmarks remains relatively constant.

From the predicted position, a corner detection with subpixel accuracy is performed to ensure high precision of the 2D location by image gradient based optimization. This low-level image processing is carried out with the OpenCV library[7].

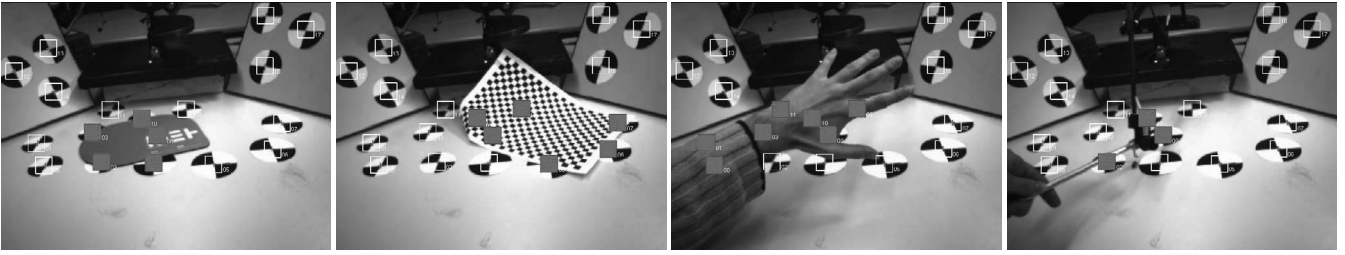


Figure 4: Landmark occlusion results. The white boxes show the search area for the 2D corner detector. The full red squares indicate the occluded landmarks

4.3.4 Occlusion Test

A model-based approach has been developed for handling landmark occlusions. Each landmark is characterized by the four satellite and the center points. These features define two triangles in the white marker region (see figure 3). Since the OPTOTRAK provides a good estimation of the camera pose, we are able to predict the 2D triangles of each landmark in the images. By picking points in a specific order during landmark calibration, the arrangement of the triangles is known.

By using the backprojected points, white triangles can be rendered in front of black background in a synthetic image. A correlation window can then be placed around the landmark center in the camera image and the synthetic image, respectively. This allows to compute a cross correlation coefficient r from both windows with $N \times M$ pixels according to

$$r = \frac{\sum_{i=1}^M \sum_{j=1}^N [(I_v(i, j) - \mu_v)(I_r(i, j) - \mu_r)]}{\sigma_v \sigma_r} \quad (1)$$

where I_v and I_r are the intensity of the synthetic and real image, and (μ_v, σ_v) and (μ_r, σ_r) the mean and standard deviation of the intensity of each correlation window. If the correlation value is lower than a pre-defined threshold, then the landmark is considered as occluded. In practice a threshold value of 0.8 gave good results. The size of the correlation window is selected to be equal to the landmark search window.

Figure 4 shows four experiments in which different types of objects occlude the landmarks. The white boxes correspond to the search window. The full red squares indicate the occluded landmarks, which will not be incorporated in the camera pose refinement. The first left picture illustrates a simple occlusion with a colored object. The next one demonstrates the robustness of the model-based approach in the presence of an occluding object with similar black and white pattern as the markers. The third picture illustrates how many landmarks can be obscured by the user's hand. Finally, the last experiment demonstrates the occlusion caused by the stylus of the haptic device. However, one of the limitations of this approach is the failure to detect occlusions caused by thin objects.

4.4 Camera Pose Refinement

Based on the 3D landmark locations, the corresponding image points, and the predicted camera pose, we refine the orientation and position of the camera with respect to the world coordinate system. We can use two possible measures to characterize the deviation between predicted and actual values, one relying on 3D measurements, the other on 2D image information. Figure 5 illustrates those errors relative to the pinhole model.

The mapping from 3D points to 2D image coordinates is performed by a rigid transformation (R, \mathbf{t}) between the world and camera coordinate system followed by a backprojection of the 3D points onto the image plane. Let M_i be a 3D point with respect to the

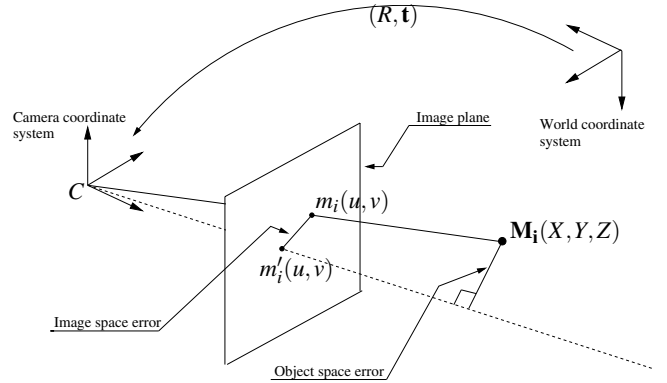


Figure 5: Definition of the image and object space errors

world coordinates and m_i be the corresponding backprojected image point. Due to uncertainty in the camera pose, the mapping transformation leads to a misalignment between the projected point m_i and the observed one m'_i in the image. The distance between m_i and m'_i characterizes the so-called *image space error*. As an alternative, Lu [16], defined the *object space error* based on the distance between the 3D point and the line-of-sight vector going through the observed image point m'_i .

Both definitions lead to an error function E depending on the camera pose (R, \mathbf{t}) . Thus, the refined camera pose can be determined by minimizing E while adjusting the extrinsic camera parameters, leading to the optimal pose (R^*, \mathbf{t}^*) .

$$(R^*, \mathbf{t}^*) = \arg \min_{R, \mathbf{t}} E(R, \mathbf{t})$$

In the following subsections, we perform a comparison of optimal camera pose determination based on both image and object space errors.

4.4.1 Image Space Approach

The image space approach, which is commonly used in photogrammetry, estimates the camera pose based on 3D-2D registration as a nonlinear least squares problem. From the notations defined above, the function E based on image space can be formulated as follows:

$$E(R, \mathbf{t}) = \frac{1}{N} \sum_{i=1}^N \|m'_i - P(RM_i + \mathbf{t})\|^2 \quad (2)$$

where N is the number of points and P is the backprojection function including the extrinsic and intrinsic parameters of the camera. Since for our haptic AR application real-time constraints have to be fulfilled, we also examine the computational efficiency of the camera pose refinement. Therefore, we investigated the efficiency of common optimization approaches used for solving nonlinear least square problems, given the good initial guess of the camera pose. Two optimization algorithms were tested for our application.

The first one is the *Levenberg-Marquardt* (LM) algorithm, which is commonly used in Computer Vision, and the second one is the *Quasi-Newton* (QN) method [20].

Both optimization approaches are based on an approximation of the function around the current position as a quadratic form. Based on the first and second derivative of the function, those methods attempt to find the optimal solution by iteratively searching the best next iterate leading to the optimum. However, computing the second derivatives of the function can be difficult and time consuming. The two approaches differ by the way the Hessian matrix is estimated.

QN computes a downhill direction vector at each iteration for determining the next step. The vector is estimated by multiplying the inverse of the Hessian matrix with the gradient of the function. The QN method gradually builds up an approximate of the inverse Hessian matrix by using gradient information from the previous steps. Thus, it accumulates a sequence of estimated inverse Hessian matrices while optimizing the function. After a certain number of iterations, the sequence converges to both an accurate inverse Hessian and the optimal solution.

In contrast to this, LM combines two optimization methods (steepest descent and Gauss-Newton) in one expression called the *augmented normal equations*. When the current solution is far from the optimum, the optimization behaves like a steepest descent and converges slowly. Conversely, LM relies on the Gauss-Newton approach when the current solution is close to the optimum, resulting in superlinear convergence. Unlike QN, LM approximates the Hessian matrix by using the Jacobian of the function and calculates a new estimate in each iteration.

Furthermore, both approaches minimize E by simultaneously adjusting the position and the orientation of the camera. We parameterized the rotation matrix R by using rotation vectors \mathbf{r} defined as $r = \theta \mathbf{n}$ where \mathbf{n} is a unit vector representing the corresponding axis and θ the angle of the rotation. This representation has the advantage that no quadratic constraint needs to be included in the optimization.

4.4.2 Object Space Approach

The principle of this method developed by Lu [16] is to compute the observed line-of-sight vector \mathbf{Cm}_i' and to project the transformed 3D point M_i onto it. Thus, we have to minimize the following error sum

$$E(R, \mathbf{t}) = \frac{1}{N} \sum_{i=1}^N \|(I - V_i)(RM_i + \mathbf{t})\|^2 \quad (3)$$

where I is the identity matrix and V_i is the observed line-of-sight projection matrix defined as:

$$V_i = \frac{\mathbf{Cm}_i' \mathbf{Cm}_i'^T}{\mathbf{Cm}_i'^T \mathbf{Cm}_i'} \quad (4)$$

The authors propose an iterative method called *Orthogonal Iterative* (OI) Algorithm to optimize the equation 3. In contrast to LM and QN, this approach is a two-stage optimization process. First, the rotation is determined by solving the absolute orientation problem. Then, the translation is computed from the optimal rotation. The process is repeated until convergence. In addition, the rotation matrix is not parameterized while optimizing. The orthogonal structure of the matrix is merely maintained by the closed form solution of the absolute orientation problem.

4.4.3 Experimental Protocol

To measure the performance of each approach, we placed 30 non-coplanar landmarks in the workspace. We measured the center and

the four satellite points of each landmark by using the calibrated pointer. To compare the optimization approaches, we recorded a set of 150 images of the real scene with all landmarks visible as the camera moves through the entire workspace. For each frame, we performed the different optimizations by using the same input data: 3D landmark position, predicted camera pose, and detected image points. The primary measurement used to compare the optimizing algorithms is the root mean square (RMS) of the backprojection error over the number of points: $\sqrt{\frac{1}{N} \sum_{i=1}^N \|m_i' - m_i\|^2}$. For the image space approach, this measurement corresponds to the RMS of the residual error of the function E . When using the object space approach, the landmarks are backprojected in the images after the optimization and the RMS measurement is computed.

Concerning the implementations of the optimizers, we used the LM code available in the VXL library, which is based on the math library MINPACK. We used the implementation of CFSQP [13] for QN. Originally, this program has been designed for solving large scale constrained nonlinear optimization problems based on sequential quadratic programming. The latter computes a descent direction and an update of the Hessian matrix similar to QN method. Finally, the OI algorithm was implemented with the VXL library.

4.4.4 Optimization Algorithm Performances

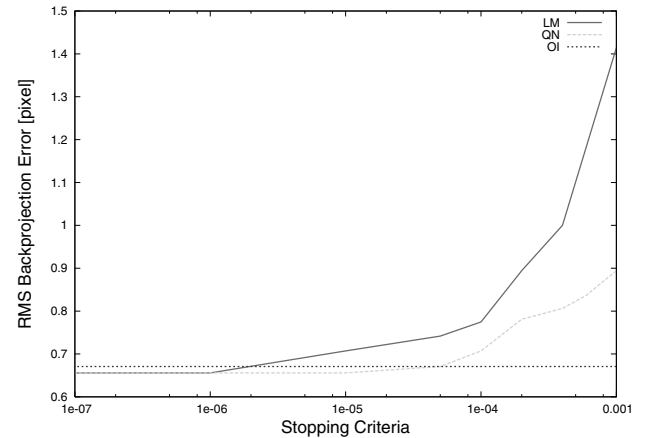


Figure 6: Influence of the stopping criteria on the backprojection error

All three optimization algorithms are based on an iterative approach. This implies that stopping criteria are used to terminate the iterative process when the optimizer is sufficiently close to the solution. Commonly the upper limit of either the changes of the parameter values or the goal function between two subsequent iterations is specified. Of course, the performance of each algorithm depends on the setting of those criteria.

We carried out the camera pose refinement with all three optimization methods by varying the stopping criteria and compared the RMS backprojection error as a function of the number of iteration and the execution time. Figure 6 illustrates the influence of the stopping criterion on the RMS backprojection error. Clearly, the OI method seems to reach an optimal solution faster than the other algorithms and is not influenced by the selection of the stopping criteria. One explanation might be that the small changes of the parameter values or the goal function defined in the 3D space are hardly noticeable in the image space when a very good initial guess is provided. However, the results show that QN and LM can provide a smaller backprojection error when setting the stopping criterion low. In addition, QN reaches the final RMS error faster than LM. Figure 7 depicts the number of iterations necessary to

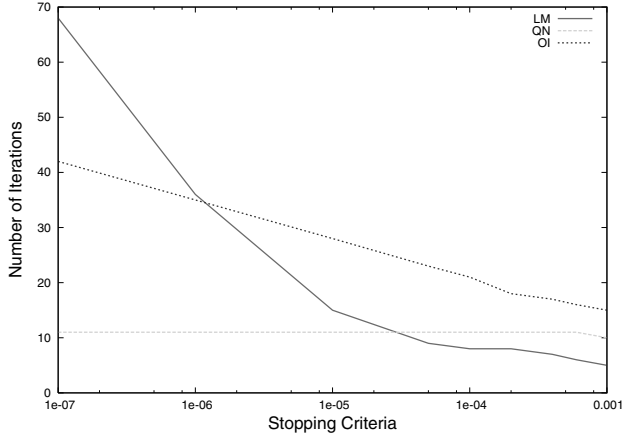


Figure 7: Influence of the stopping criteria on the number of iterations

reach an optimal solution. It can be seen that stopping criteria significantly influence the iterative process of LM and OI, whereas QN basically uses a constant number of iterations. In addition, OI needs more iterations to reach an optimal solution than the other algorithms. However, it seems that there is an optimal stopping criteria around 10^{-5} to obtain both low number of iterations and small RMS error for LM and QN. Notice that a low value of the stopping criteria ensures the convergence resulting in a stable camera pose. As revealed by figure 8, this optimal value does not provide optimal speed for both algorithms. The execution time of LM decreases significantly with increasing stopping criterion, whereas QN presents a constantly high performance. Despite the high iteration number, OI maintains a low execution time.

As a result, QN seems to perform superior in term of RMS back-projection error, number of iterations and overall time consumption. For that reason, we have chosen QN for camera pose refinement.

4.4.5 Occlusion Issues

In this section, we analyze the influence of occluded landmarks on the precision of camera pose estimation. We carried out an experiment in which the camera was randomly placed in such a way that the entire workspace was observed with all landmarks visible. We recorded 150 poses with the corresponding detected landmark positions resulting in 150 datasets. For each set we randomly disregarded landmarks, performed the optimization with the remaining visible features and finally backprojected the 3D points from the optimal solution. We repeated this test for the current set 100 times, then computed the standard deviation of the optimized camera pose and the backprojection error. This way we can characterize the uncertainty of the camera pose as well as the backprojection while occlusion occurs.

We performed three different tests. First only 5 landmarks were occluded among 30. Then, we increased the number of occluded features to 15 and 24. We did not continue further, because at least 6 visible landmarks are necessary for estimating the 6 parameters of the camera pose. Figure 9 illustrates the influence of the occlusion on the translation of the camera along the Z axis after optimization. The uncertainty of the camera pose increases significantly with diminishing number of visible landmarks.

The uncertainty of the optimized pose is revealed in the image by a jitter of the virtual object. During the experiments we could observe that some landmarks had a stronger influence on the pose stability than others. Unfortunately, we did not succeed to categorize these landmarks according to either the backprojection error or their 3D location. In order to further enhance the stability, we focused on reducing the influence of measurement noise on the optimization process.

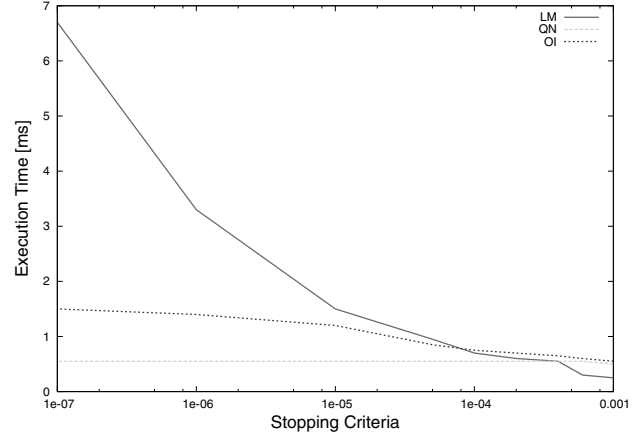


Figure 8: Influence of the stopping criterion on the execution time

4.5 3D Landmark Refinement

We have to realize that the method described above has only taken noise in 2D measurements into account. However, as analysed in Section 4.3.2 the procedure used for the pointer calibration introduces an uncertainty of the tip position in the world coordinate system, which means that the camera pose refinement used is limited by the corrupted 3D measurements. Therefore, we also examined the refinement of the 3D landmark positions in order to further reduce the jitter of virtual objects.

Incorporating the correction of the landmark location to improve the registration has been only seldomly investigated. In [19], the authors register a 3D model of the liver reconstructed from CT data with video images of the patient. Based on stereoscopic acquisition, they reconstruct the location of 3D landmarks placed on the skin of the patient and register them with the corresponding points located on the 3D model. In order to decrease the influence of landmark reconstruction errors on the registration, a new criterion is proposed by extending the goal function with the corresponding 3D deviation. As a result, an iterative optimization procedure refines both the CT-to-patient transformation and the 3D landmark positions.

Similar to our setup, in [8] a tracked endoscope is used to reconstruct 3D landmarks placed on the skin of the patient in order to overlay CT data onto images of the patient. By recording simultaneously video images and endoscope poses, the authors present a 3D landmark measurement method based on the intersection of the line-of-sight rays. They show that the vision-based method yields more accurate results in terms of registration error than measuring the 3D landmark location with a pointer tool.

In our case, the hybrid AR system is based on a monoscopic video acquisition. In addition, the landmarks are rigidly fixed in the workspace and measured approximately. The camera pose is reported by the optical tracking device. For these reasons, we use Bundle Adjustment to refine the 3D measurements in order to test whether the refinement of the landmarks can improve the quality of the overlay process. In an offline procedure, we simultaneously record the landmark image points and the camera poses. Given those data and the measured landmark locations, we carry out a bundle adjustment to solve the following problem:

$$\min_{R^j, \mathbf{t}^j, \mathbf{M}_i} \frac{1}{N} \sum_{j=1}^M \sum_{i=1}^N \|m_i^j - P(R^j \mathbf{M}_i + \mathbf{t}^j)\|^2 \quad (5)$$

We captured a sequence of $M = 500$ images, detected the landmark image points, and stored the corresponding camera poses. Since the camera pose is constrained by the field of view of the optical tracking system, we were only able to partially cover the

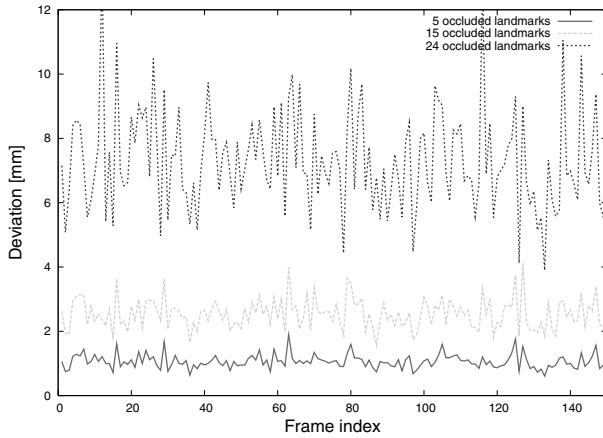


Figure 9: Influence of occluded landmarks on the camera pose before 3D refinement

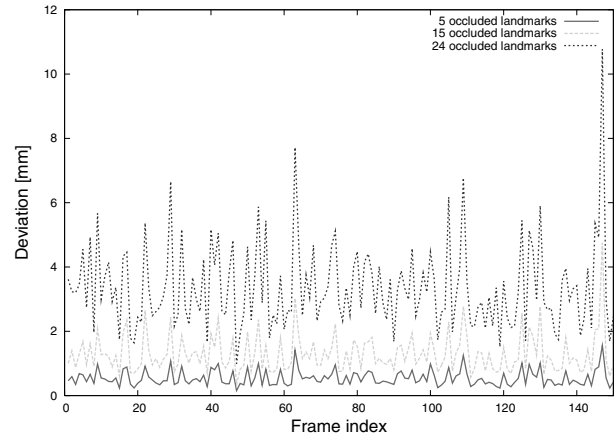


Figure 10: Influence of occluded landmarks on the camera pose after 3D refinement

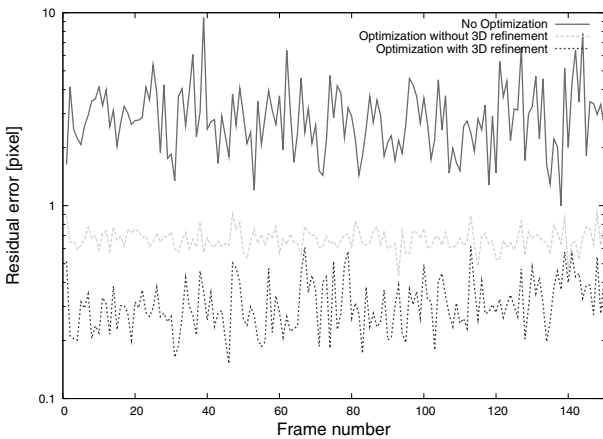


Figure 11: Comparison results of the influence of the 3D landmark refinement on the RMS error

workspace. In order not to influence the 3D landmark refinement with dominant camera poses, we sampled the workspace in cells. For each cell, we randomly selected the recorded poses located within the current cell. We fixed a maximum number of camera poses per cell in order to obtain an uniform spatial distribution. Finally, we used around 100 poses to perform the bundle adjustment. The implementation of the latter has been described in [15].

As revealed by Figure 10, the uncertainty of the pose estimation decreases significantly after the landmark refinement. Figure 11 illustrates the results of the refinement in terms of backprojection error. The latter was computed by backprojecting the 3D landmarks on images not used by the 3D refinement process. The top curve represents the overlay results when using the raw camera pose reported by the optical tracking system. The second curve shows the improvement obtained after the camera pose refinement. Finally, the last one depicts the results when both camera pose and landmark positions are optimized. In addition, the difference between the refined and measured landmark positions in average is around $1mm$, which corresponds nearly to the pointer calibration error.

As a result, the refinement of both camera pose and landmark measurements yield very low jitter of the virtual object in the video sequence.

5 AUGMENTED REALITY HAPTIC APPLICATION

5.1 Overview

In our AR test setup deformable real and virtual objects are simultaneously presented. The real objects are silicone cylinders, while the virtual counterparts are tuned to match the deformation behaviour of the former. This includes visual appearance, as well as force feedback from the interaction. A haptic device is incorporated into the system to provide the feedback.

In this section, we describe the integration of the hybrid AR system and the deformable object simulation. First, the system architecture is explained. Then, the deformation and force computation model are briefly discussed. Finally, the visual results of the haptic enhanced AR system are shown.

5.2 System architecture

One of the main challenges in an AR system is to maintain low latency in order to synchronize the augmented images with the user motion in time. Since a standard AR setup already has to meet high computational requirements, adding a haptic interface and performing physical simulations lead to excessive demand on computational power to be provided by the system. Therefore, we have developed a distributed system to meet the related requirements. As discussed in [17], several possibilities for task distribution in such an application exist. In our case, we use a *graphics server* and a *physics server*.

The former carries out all tasks typical to a standard AR setup. Thus, in our context the external input data needed are the pose of the haptic interface, as well as of the virtual object in the scene. However, the mesh of the deformable object requires an update in each frame due to the user interaction. To minimize the data transfer between both machines, we only update the surface of the mesh, which significantly decreases the number of data packets to be transmitted. The second computer takes care of force-feedback computation and the physical simulation of virtual objects. Thus, the server is completely independent from tracking or image acquisition. Communication between the two machines is accomplished via an ethernet connection with the TCP protocol. In [5], we developed a synchronization process and a communication model in order to ensure coordination during run-time.

In our testbed implementation described below, we have used two dual PCs with $2.8GHz$ and $2.4GHz$ CPUs, respectively. Both machines have $2GB$ RAM with $512KB$ cache, and are running under a Linux OS. On the haptics server, the haptic loop is updated with a fixed refresh rate of $1kHz$. The physics computation runs in

a separate loop at 200Hz. On the graphics server the AR pipeline is working with an update rate of about 25Hz. The visualization is performed by a NVIDIA Geforce 6600 GT card.

5.3 Deformation and force computation model

The computer-generated deformable object is modeled according to a real silicone cylinder. The model parameters are tuned such that the simulated deformations are close to the real ones [14]. In order to fulfill real-time requirement, a mass-spring model (MSS) [23] is used for our application. This model consists of a mesh of mass points connected by elastic links (springs). Due to the simplicity of the motion equations and of the implementation, the MSS is computationally attractive for real-time applications. System movement is evaluated by integrating Newton's second law of motion:

$$M \frac{\partial^2 x}{\partial t^2} + D \frac{\partial x}{\partial t} + F_{internal} = F_{external}$$

where M is the mass matrix, D damping matrix and F the internal and external forces respectively. A linear spring is used to compute the internal forces. In addition, the deformations are enhanced by applying volume preserving forces on each vertex of the tetrahedral structure. In order to compute the haptic feedback forces during the interaction, a proxy-based haptic rendering is performed.

5.4 Calibrations

5.4.1 Haptic Device

We integrated a SensAble PHANToM 1.5 haptic device into our setup. In order to align the haptic and the world coordinate system, a calibration procedure is required. We developed an accurate calibration method without using additional sensing devices [5]. The underlying idea of the approach is to collect 3D point measurements in both coordinate systems by rigidly attaching a marker to the tool tip of the haptic device. After estimating the marker-tip transformation with the pivot method, we recorded two sets of 3D points expressed in both coordinates. From those data, we solved the *absolute orientation* problem by following a least-square fitting approach [2]. However, additional errors in the estimation of the haptic-world transformation are introduced due to the inaccuracies in haptic encoder initialization. Therefore, we carried out a two-stage optimization process. First we determine the rigid transformation followed by an encoder joint angle correction. The final calibration results revealed that the alignment error was below 1.5mm.

5.4.2 Virtual Model

To overlay the virtual silicone block onto the video images, we have to measure its position and orientation in world coordinates. In addition, we also need to know its pose relative to the haptic frame for force feedback computation. Therefore, we calibrated the table relative to the world frame in order to place the virtual model. Around 10 points were measured by means of the pointer. Then, we defined a coordinate system attached to the table by computing the best plane fitting the data. From the table frame, a table-world transformation was estimated. As a result, the virtual object was placed with respect to the table coordinate system. Given the table-world and world-haptic transformation, the same object can be expressed in the haptic frame. The virtual cylinder has finally been placed in the middle of the haptic workspace in order to obtain optimal haptic rendering.

5.5 Realism Enhancement

The purpose of our application is to provide highly realistic overlaid images to the user in such a way that the virtual object is hardly distinguishable from the real one. Therefore, the visualization incorporated lighting effects including the shadow shed by the virtual

cylinder and the tool tip of the haptic device. To achieve this, we used controlled lighting conditions by a lamp. The position of the bulb has been measured in the world coordinate system with the pointer. Shadows are then rendered using a shadow volume rendering approach. Additionally, we rendered the virtual model with a texture corresponding to the image of the real silicone cylinder. These enhancements were necessary to create a realistic illusion.

5.6 Results

For this application, we placed 16 landmarks on three different planes in order to cover the user's field of view while moving. We also attempted to consolidate the accuracy and the stability of the virtual object by encircling the real location of the latter with the landmarks. In addition, an extra landmark was placed at the expected center position of the bottom of the virtual cylinder. Experimentally we figured out that certain configuration of landmarks led to instability of the overlaying process, particularly in the case where the user is close to the virtual object and only few landmarks are visible. If the configuration of the detected landmarks is colinear, then the camera pose refinement provides instable results.

Figure 12 shows what the user can see through the head mounted display of the hybrid AR haptic setup. The first row shows the landmark detection as the user interacts with the virtual cylinder. In the second row, the computer-generated object is deformed in real-time. As a result, the user can directly interact with the virtual cylinder by manipulating the stylus. Due to the highly precise haptic calibration, the interaction between the virtual object and the real haptic tool becomes natural for the user. Interaction with the real cylinder is shown by the last rows. As revealed by the latter, the simulation of the virtual cylinder generates similar behavior as the real one. Finally, the last row shows a close view of the augmented scene with the minimum number of landmarks necessary to maintain the virtual object stable. Furthermore, the virtual shadows of the cylinder and the real stylus improve the realism of the augmented scene. To achieve this, the stylus is also modeled by a virtual tool in order to compute the shape of the shadow. As a consequence, the virtual object clearly becomes part of the real scene

Component	Average time [ms]
Frame rendering	15
Frame unwrapping	5
Corner detection + occlusion test	20
Camera pose refinement	0.6
Object Rendering (Shadow+Texture)	4.5

Table 1: Rendering performances

As mentioned earlier, only the surface of the cylinder is displayed, which improves the performance of the rendering process. Thus, the virtual object consists of 402 triangles and the virtual haptic tool of 320 triangles. Table 1 summarizes the execution time of each component of the AR system. Frame rendering consists of applying the Bayer filter on the raw images and then rendering the resulting RGB images. The conversion is the most time consuming part of this process. Unwarping the images allows us to obtain an accurate alignment of the virtual object with the real world, since the rendering process does not enable us to distort the computer-generated images. The high execution time of the corner detector is primarily caused by the similarity measurement between the virtual and the real landmarks. Since the textures are loaded on the graphic card, the object rendering process is also fast. As a consequence, a delay between the physical world and the augmenting virtual objects could clearly be perceived by the user. Since measuring the latency is not a straightforward task, we can only estimate this delay to be between 60 and 100ms. However, the user still feels immersed into the AR world.



Figure 12: From the top, the first row shows the AR scene with detected landmarks. The second row illustrates the interaction between the haptic device and the deformable object. The third row depicts similar actions on the real object. The last row provides a close view of the deformations.

An additional limitation of the hybrid AR system is the frame rate of the camera. As the user moves fast, the image becomes blurred leading to undetected landmarks. As a result, the visual correction fails and the camera pose estimation only relies on the optical tracking data.

6 CONCLUSION AND FUTURE WORK

We have developed a stable and robust hybrid augmented reality system by combining an external tracking device with a visual landmark-based tracking system. Two approaches to compute the camera pose from 3D-2D correspondences were compared. The study showed that both methods provide similar results in terms of backprojection error. In addition, the optimization performance was investigated in order to meet the real-time constraint for the final application. It turned out that the Quasi-Newton algorithm led to higher accuracy and faster convergence than the more commonly used Levenberg-Marquardt and the Orthogonal Iteration method. Finally, we demonstrated the performance of the hybrid AR system in a haptic AR application, allowing to combine realistic haptic and visual feedbacks. The first prototype shows convincing results, performing force computation, highly realistic rendering and the simulation of deformable objects in real-time. Despite the high latency of the system, users felt immersed into the augmented world. One shortcoming of our current system is the lack of stereo rendering. Since depth cues are only present due to shadows and motion parallax, interaction with the objects is significantly impaired. Furthermore, occlusion between the physical world and the virtual object has not yet been handled. These problems can be overcome by adding a stereo camera to the setup in order to estimate the depth map of the real environment and to integrate this information into the rendering process. We are also working on an automatic landmark detection method in order to simplify the calibration step. Due to the stability, robustness and speed of the system, it can be used by far beyond the context of our original target aiming at visuo-haptic comparison of virtual and real objects. It will also serve as a general installation basis to implement numerous applications relying on multimodal interaction with complex scenes augmented with not necessarily rigid virtual objects, like in entertainment or skill training.

7 ACKNOWLEDGMENT

The authors would like to thank Peter Leskovsky for his contribution to the integration of the haptic interaction as well as the deformable object in the hybrid AR system.

This work has been performed within the frame of the Swiss National Center of Competence in Research on Computer Aided and Image Guided Medical Interventions (NCCR Co-Me) supported by the Swiss National Science Foundation.

REFERENCES

- [1] M. Aron, G. Simon, and M.-O Berger. Handling uncertain sensor data in vision-based camera tracking. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 58–67, Nov 2004.
- [2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987.
- [3] T. Auer and A. Pinz. Building a hybrid tracking system: Integration of optical and magnetic tracking. In *Proceeding of the 2nd IEEE and ACM International Workshop on Augmented Reality*, pages 13–22, October 20-21 1999.
- [4] M. Bajura and U. Neumann. Dynamic registration correction in augmented reality systems. In *Virtual Reality Annual International Symposium*, pages 189–196, March 1995.
- [5] G. Bianchi, B. Knörlein, G. Székely, and M. Harders. High precision augmented reality haptics. In *Eurohaptics 2006*, pages 169–177, July 2006.
- [6] G. Bianchi, C. Wengert, M. Harders, P. Cattin, and G. Székely. Camera-Marker Alignment Framework and Comparison with Hand-Eye Calibration for Augmented Reality Applications. In *IEEE/ACM*

- International Symposium on Mixed and Augmented Reality*, pages 188–189, 2005.
- [7] G. Bradski. The OpenCV library. *Dr Dobb's Journal*, 25(11):120, 122–125, November 2000.
- [8] M. Feuerstein, S. Wildhirt M., R. Bauernschmitt, and N. Navab. Automatic patient registration for port placement in minimally invasive endoscopic surgery. In *Lecture Notes in Computer Science*, volume 3750, pages 287–294, September 2005.
- [9] W. Hoff and T. Vincent. Analysis of head pose accuracy in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):319–334, 2000.
- [10] M. C. Jacobs, M. A. Livingston, and A. State. Managing latency in complex augmented reality systems. In *Symposium on Interactive 3D Graphics*, pages 49–54, 1997.
- [11] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer. Event-based haptics and acceleration matching: Portraying and assessing the realism of contact. In *World Haptics Conference*, 2005.
- [12] S. Lavalle, P. Cinquin, and J. Troccaz. *Computer Integrated Surgery and Therapy: State of the Art*. IS Press, Amsterdam, NL, in C. Roux and J.L. Coatrieux edition, 1997. Chapter 10, pages 239-310.
- [13] C. Lawrence, J. L. Zhou, and A. L. Tits. User's guide for cfsqp version 2.5: A c code for solving (large scale) constrained nonlinear (min-max) optimization problems, generating iterates satisfying all inequality constraints.
- [14] P. Leskovsky, M. Harders, and G. Székely. Assessing the fidelity of the haptically rendered deformable objects. In *14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE Virtual Reality, pages 19–25, March 2006.
- [15] M.I.A. Lourakis and A.A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004. Available from <http://www.ics.forth.gr/~lourakis/sba>.
- [16] C.-P. Lu, G.D. Hager, and E. Mjølness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.
- [17] W. R. Mark, S. C. Randolph, M. Finch, J. M. Van Verth, and R. M. Taylor II. Adding force feedback to graphics systems: Issues and solutions. volume 30, pages 447–452, 1996.
- [18] H. Najafi, N. Navab, and G. Klinker. Automated initialization for marker-less tracking: A sensor fusion approach. In *IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 79–88, 2004.
- [19] S.A. Nicolau, X. Pennec, L. Soler, and N. Ayache. A complete augmented reality guidance system for liver punctures: First clinical evaluation. In J. Duncan and G. Gerig, editors, *Proceedings of the 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, Part I*, volume 3749 of LNCS, pages 539–547, Palm Springs, CA, USA, October 26-29, 2005. Springer Verlag.
- [20] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.
- [21] A. State, G. Hirota, D.T. Chen, W.F. Garrett, and M. A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *SIGGRAPH*, pages 429–438, August 1996.
- [22] H. Tan, B. Adelstein, R. Traylor, M. Kocsis, and D. Hirtleman. Discrimination of real and virtual high-definition textured surfaces. In *Haptic Symposium*, 2006.
- [23] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics (Proc. SIGGRAPH'87)*, 21(4):205–214, 1987.
- [24] R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand-eye calibration. In *IEEE Transactions on Robotics and Automation*, volume 5, pages 345–358, June 1989.
- [25] S. You, U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In *IEEE Virtual Reality*, pages 260–267, March 1999.