Model-based Hybrid Tracking for Medical Augmented Reality

Jan Fischer[†]   Michael Eichler   Dirk Bartz   Wolfgang Straßer

WSI/GRIS - VCM,
University of Tübingen, Germany

Abstract

Camera pose estimation is one of the most important, but also one of the most challenging tasks in augmented reality. Without a highly accurate estimation of the position and orientation of the digital video camera, it is impossible to render a spatially correct overlay of graphical information. This requirement is even more crucial in medical applications, where the virtual objects are supposed to be correctly aligned with the patient. Many medical AR systems use specialized tracking devices, which can be of limited suitability for real-world scenarios. We have developed an AR framework for surgical applications based on existing medical equipment. A surgical navigation device delivers tracking information measured by a built-in infrared camera system, which is the basis for the pose estimation of the AR video camera. However, depending on the conditions in the environment, this infrared pose data can contain discernible tracking errors. One main drawback of the medical tracking device is the fact that, while it delivers a very high positional accuracy, the reported camera orientation can contain a relatively large error.

In this paper, we present a hybrid tracking scheme for medical augmented reality based on a certified medical tracking system. The final pose estimation takes the inital infrared tracking data as well as salient features in the camera image into account. The vision-based component of the tracking algorithm relies on a pre-defined graphical model of the observed scene. The infrared and vision-based tracking data are tightly integrated into a unified pose estimation algorithm. This algorithm is based on an iterative numerical optimization method. We describe an implementation of the algorithm and present experimental data showing that our new method is capable of delivering a more accurate pose estimation.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Interfaces and Presentation]: Artificial, augmented, and virtual realities; I.4.8 [Image Processing and Computer Vision]: Tracking; J.3 [Life and Medical Sciences]: Medical information systems

1. Introduction

In augmented reality (AR), virtual graphical objects are overlaid over the real environment of the user. In video see-through AR, this is achieved through the acquisition of a video stream from a camera recording the physical world. These digital video frames are then mixed with the rendered representations of graphical objects enriching the user's real surroundings [ABB*01].

One of the most important preconditions for a useful augmented reality display is the correct spatial alignment of virtual models with respect to the camera image. This means that the threedimensional position and orientation of graphical objects is defined relative to a fixed coordinate system in the real world. Changes in the location or viewing direction of the digital camera then lead to a correspondigly adapted projected 2D image of virtual models. In order to be able to achieve such a correct spatial alignment, the position and orientation of the camera relative to a real-world coordinate system have to be known. This task is commonly referred to as *camera tracking*. Since the combined position and orien-

---

[†] e-mail: fischer@gris.uni-tuebingen.de

tation information is often called *pose*, the term *camera pose estimation* is also frequently used.

A very common method for camera tracking in augmented reality is vision-based marker tracking. The widely used ARToolKit library is an example of this approach based on artifical fiducials consisting of pre-defined marker patterns, which are manually placed in the observed scene [KB99]. While this method is relatively easy to set up and cost-effective, it is sensitive to occlusion of the fiducials as well as large camera distances and angles. Moreover, a placement of fixed marker patterns in the environment is not practical in many application scenarios. In addition to the vision-based tracking of fiducials, techniques like markerless tracking and specialized tracking devices (e.g., infrared cameras or magnetic trackers) are often utilized in augmented reality.

The support of medical diagnostics and therapy has long been in the focus of application-oriented augmented reality research. An example is the system for overlaying ultrasound images over the patient developed by Bajura et al. in the early 1990s, which was one of the first medical AR applications [BFO92]. In medical applications, a highly accurate camera pose estimation is especially important. Larger errors in the determined position or orientation lead to an incorrect placement or alignment of the augmentations. This can result in the information overlay becoming obviously useless or, even more problematic, in a misinterpretation of spatial relationships by the user (i.e., the physician). As a second crucial task specific to medical augmented reality, the placement of the patient in the global coordinate system has to be determined. This problem is known as *patient registration*. Since practically all useful information overlays in medical applications are relative to the patient, inaccuracies in the patient registration can cause a significant displacement of graphical objects from their correct position in the camera image.

Most previously described medical augmented reality systems rely on specific hardware for performing camera tracking and patient registration. Frequently used types of specialized equipment include magnetic tracking systems and dedicated infrared tracking cameras (e.g., in the application described by Sauer et al. [SKB*01]). The use of specialized hardware components can be problematic for the transition of a medical AR system from an experimental state into the clinical practice. Many of these dedicated tracking devices have originally been designed for applications in industry or virtual reality and are not optimally suitable for medical scenarios. They often are expensive and can require tedious setup procedures. Furthermore, practical problems like working in a sterile environment and certification for medical settings usually have not been solved for specialized VR and AR tracking and display equipment.

Recently, we have presented a new experimental augmented reality system for medical applications. Unlike

Figure 1: *VectorVision$^{®}$ Image Guided Surgery device with infrared camera system, touchscreen, and PC (in base). The VectorVision system is manufactured by the BrainLAB company (Heimstetten, Germany). (Image taken from [Bra04].)*

most existing medical AR setups, our new system, *ARGUS*, is based on a commercially available surgical apparatus [FNFB04]. A so-called imaged guided surgery (IGS) device is the basis for a video see-through augmented reality application. We use a *VectorVision$^{®}$* IGS system, which is equipped with a pair of accurate infrared cameras (see Fig. 1). The tracking data from this infrared camera system is accessed using an application-specific network interface.

A drawback of the medical augmented reality system is the fact that sometimes discernible tracking errors can occur. One of the reasons for this is the fact that the angular accuracy of the delivered tracking data is limited. Moreover, errors can be caused by a number of other factors including an inaccurately performed system calibration or an inadequate patient registration. In this paper, we present a hybrid tracking scheme which aims at reducing the overlay error of graphical objects in medical AR while not requiring any additional equipment.

In the remainder of this paper, Section 2 will review some related previous work. Section 3 describes the medical augmented reality system in more detail. An overview of the new hybrid tracking scheme is given in Section 4. Section 5 contains a detailed discussion of the algorithm. Some results obtained with the new method are presented in Section 6. Finally, Section 7 concludes this paper with a summary.

2. Related Work

In recent years, experimental AR systems were created for the support of various medical application scenarios. Navab et al. combined a specialized mobile X-ray system, a so-called C-arm, with a CCD camera in order to overlay acquired volume data over the optical image [NBHM99].
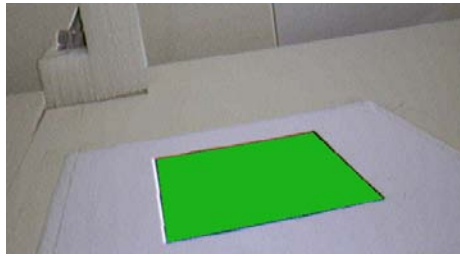
(a) Correct graphical overlay
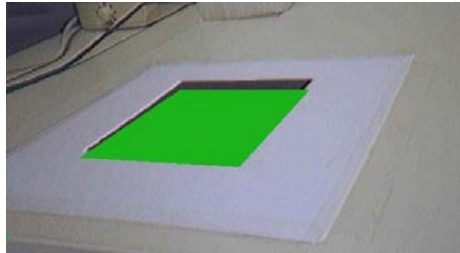
(b) Discernible tracking error

Figure 2: *Comparison of correct and erratic overlay of graphical information. In this example, a virtual green square is rendered over the measured position of an AR-ToolKit marker (however, tracking is performed based on the infrared tracking cameras of the IGS system). In 2(b), the virtual marker is visibly displaced from the correct location due to an inaccurate system calibration.*

Sauer et al. have described a video see-through augmented reality system for the visualization of ultrasound images [SKB*01]. In this system, a dedicated infrared camera, which is attached to the head-mounted display worn by the user, is used for tracking. The Medarpa project, which uses a special translucent display device mounted on a swivel arm, was described by Schwald et al. [SSW02]. An augmented reality setup for supporting livery surgery was demonstrated by Bornik et al. [BBR*03]. In this system, dedicated optical cameras are used for tracking. Among the latest developments in medical augmented reality is the method described by Feuerstein et al. for supporting optimal port placement in robotically assisted heart surgery [FWBN05].

The combination of different tracking techniques, an approach which is known as *hybrid tracking* or *sensor fusion*, has also been used in various augmented reality applications. As a very early example, State et al. [SHC*96] presented a medical AR system which integrated a magnetic tracker with optical landmark information from the camera image. The objective of this approach can be considered to be somewhat similar to our system, however, they used a specialized magnetic tracking device which which is not designed to be used in a real medical scenario. Hybrid tracking methods for wide area outdoor augmented reality were described for in-

stance by Piekarski et al. [PATM03]. You et al. developed a hybrid tracking system for AR, combining a specialized intertial tracker with vision-based pose estimation [YNA99].

3. Medical Augmented Reality based on Image Guided Surgery

In contrast to the vast majority of previous experimental augmented reality systems for medicine, our medical AR framework *ARGUS* does not require special hardware for tracking or display in addition to existing medical equipment. We have developed a video see-through system which acquires camera tracking information from an existing, commercially available, and certified image guided surgery device (see Fig. 1). As mentioned above, this IGS device is equipped with an infrared camera system for tracking surgical instruments during an intervention. In our AR setup, a marker clamp consisting of a known configuration of reflecting spheres is attached to the digital video camera used in the system. Since the pose information delivered for the infrared marker clamp is in relation to a hypothetical surgical tool, it cannot directly be used for generating a spatially correct overlay of graphical information. Therefore, we have developed a specialized calibration step in order to make the use of the IGS tracking data for AR image mixing possible [FNFB04].

The main advantage of using commercial medical equipment as the basis for an AR system is the fact that it is certified to be used in actual surgical interventions, and that many practical problems like working in a sterile environment have already been taken care of. Moreover, the IGS system can deliver useful additional information for the augmented reality application. This includes the position and orientation of tracked surgical tools or endoscopes, volumetric datasets of the patient's anatomy, and the registration of the patient in the global coordinate system. The patient registration can be performed using a number of registration procedures which are available in the IGS device.

Based on our medical AR framework, a number of extensions and applications were developed. These include a threedimensional interaction system using untethered interaction tools and a method for static occlusion handling [FBS04, FBS05]. Figure 3 shows an image generated by the medical AR system. The image contains a virtual tumor model as well as the graphical representation of a tracked surgical tool, correctly occluded by the plastic phantom skull visible in the acquired camera image.

It is a problem of the basic medical AR framework that visible overlay errors sometimes occur due to inaccurate camera tracking (see Fig. 2). These inaccuracies can be caused by a number of factors like a lack of diligence when the calibration step is performed, an inadequate patient registration, problems with the infrared camera hardware or environmental conditions. The hybrid tracking scheme presented
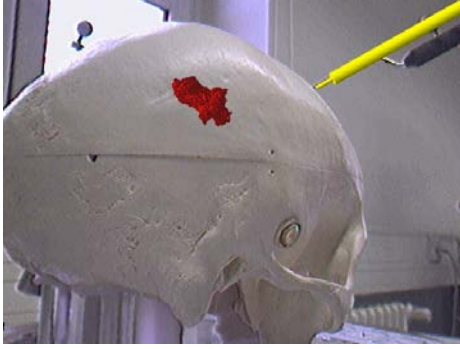
Figure 3: *Overlay of a virtual tumor model (center) and a tracked surgical instrument (top right) over the camera image.*

in this paper was developed to improve the tracking in these situations.

4. Overview of the Hybrid Tracking Method

We propose a hybrid tracking approach for medical augmented reality. This hybrid tracking system combines an initial pose estimation from the infrared cameras with information from a digital camera image. This way, the advantages of the two basic tracking methods complement each other. The infrared tracking provided by the medical device is stable in the sense that it delivers a pose estimation in practically every frame. Because the infrared cameras are mounted on a movable swivel arm and they have a large trackable volume, failures of the infrared tracking due to occlusion or visibility problems rarely occur. The stability of this infrared pose estimation is combined with the improved accuracy of the image-based component. While the image-based method can compensate errors in the position and orientation estimation of the infrared cameras, it is not able to deliver an initial global tracking. Without an initial estimate for the pose of the camera, the offscreen rendering of the model and the definition of template image search areas (see below) would not be possible. Therefore, the infrared tracking data and the image-based component complement each other, constituting a hybrid tracking system.

An overview of the method is shown in Figure 4. The presented approach is a model-based algorithm. This means that it relies on a geometrical model of a real object in the observed scene. In a preparatory step, this reference model is manually defined by the user. The model consists of a set of salient feature points defined in 3D together with associated pictures showing the respective portion of the real object.

In the central application loop of the augmented reality application, at first the initial pose estimation is acquired from the image guided surgery system. The system then renders a graphical representation of the geometrical model into

the currently not visible back frame buffer. For this rendering process, the infrared pose estimation is used as global coordinate system transformation. For each of the previously defined salient feature points, a template image is then read from the back frame buffer. The system looks for a corresponding location in the camera image for each template image, in a search area centered at the projected feature point position. This yields the position of a so-called *correspondence point* in the camera image, as well as a measure for the similarity of this camera image location with the image template.

The pairs of feature point positions and detected correspondence points are then fed into a numerical optimization algorithm. We use an iterative optimization approach, which starts with the infrared pose estimation and incrementally updates the pose parameters in order to minimize the reprojection error of the feature points. This approach can be considered a tightly integrated hybrid method because both the infrared pose estimation and the image-based feature correspondences are used in the same numerical computation. In order to improve the numerical stability of the algorithm, a number of pre-processing steps are performed on the point correspondence data, which are described in Section 5.

In the remainder of this paper, we will use the following nomenclature:

- The salient feature points defined in 3D are called the *reference points* **R**, consisting of individual reference points $r_i = (x_{r_i}, y_{r_i}, z_{r_i})$.
- The set of two-dimensional correspondence points is called **C**, containing the individual correspondence points $c_i = (x_{c_i}, y_{c_i})$.
- The initial pose estimation delivered by the infrared camera system is called *infrared pose*. This pose is represented as a transformation matrix M_{IR}.
- The (iteratively refined) pose estimation delivered by the hybrid tracking algorithm is called the *hybrid pose*, stored in a transformation matrix M_{Hyb}.
- At the beginning of the algorithm, the reference points are projected into 2D camera image space with the initial infrared pose. These projected reference points are denoted as $p_i = (x_{p_i}, y_{p_i})$.
- In order to make a simplified notation possible, we use the function $proj(a)$ for denoting the projection of a 3D point a into two-dimensional image space. In our system, the internal camera parameters are determined in the existing AR framework. The projection operation then consists of multiplying the 3D point with the camera parameter matrix and subsequently performing the perspective division. Using this notation, the connection between the reference points and the associated projections can easily be formulated as $p_i = proj(M_{IR} \cdot r_i)$.
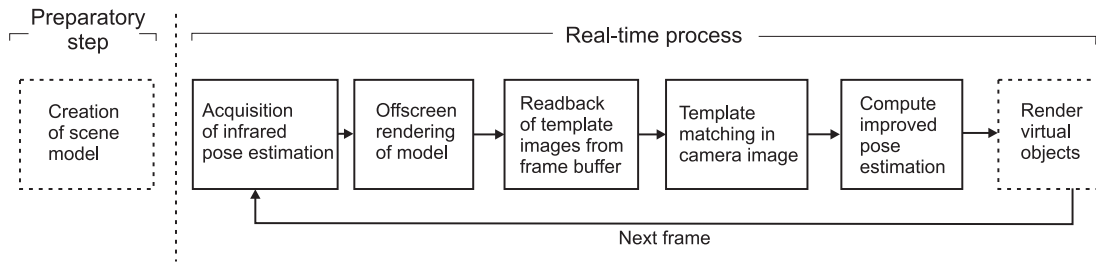
Figure 4: *Overview of the proposed hybrid tracking algorithm.*

5. Description of the Algorithm

5.1. Creation of Reference Model

Our proposed hybrid tracking algorithm requires a geometrical model of some real object in the observed scene. In the current implementation of the system, this reference model is created manually by the user with a separate software tool. The model is defined using a simple process based on a sequence of camera viewpoints. For each viewpoint, the digital camera is placed so that it can take a picture of some portion of the real reference object. The user then indicates salient reference points in the visible portion of the model. The definition of the 3D point positions is done with a special pointer tool which is tracked by the IGS device. A specific rotating pointer tool gesture is recognized by the system and used for determining the position of the salient point. This process is illustrated in Figure 5. (For a more detailed description of user interaction in the *ARGUS* system see [FBS05]).
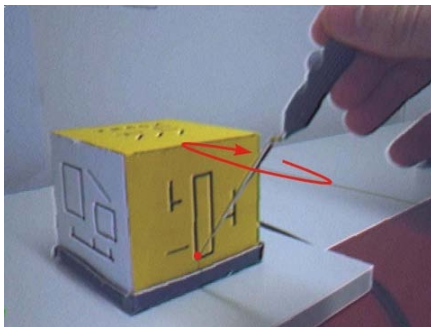

Figure 5: *Definition of a reference point with a rotating pointer tool gesture. The reference object used here is a cube with artifical high-contrast patterns.*

For each defined reference point, a square image template (60 x 60 pixels) centered at the point position is copied from the current camera image. In addition to the 3D point position and the associated image data, the spatial position of the corners of the image template are also stored. This is necessary so that it is later possible to render the reference model. These corner positions are determined by calculating their locations in the camera image and back-projecting these locations into the 3D world coordinate system. In summary, the following data are stored for each reference point:

1. Reference point location r_i
2. Associated template image
3. 3D position of the four corner points

Figure 6 shows a graphical representation of the model data acquired for the reference cube which was used during the development of the algorithm.
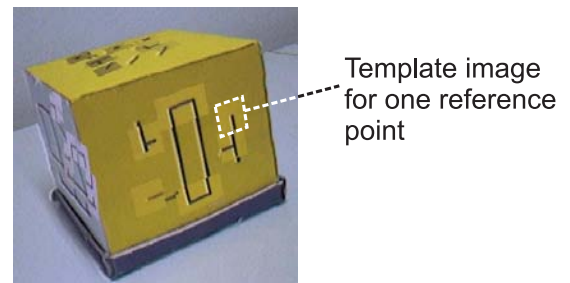

Figure 6: *Visualization of the model data acquired for the reference cube. In this image, the projected image templates are overlaid over the real camera image.*

5.2. Offscreen Rendering of Model

At the beginning of the main application loop of the AR system, the current infrared pose estimation for the digital camera, M_{IR}, is acquired from the IGS device (see Fig. 4). Subsequently, the reference model is rendered according to this initial pose estimation. This is achieved by setting the OpenGL transformation matrix such that it reflects the camera pose. For each reference point, a square is then rendered in 3D using the stored image corner positions from the model definition stage (see Sec. 5.1). Each reference point square is textured with the corresponding template image acquired during the model definition. Figure 7 shows the offscreen representation of the reference cube model used during development (also see Fig. 12(a) in the color plate).

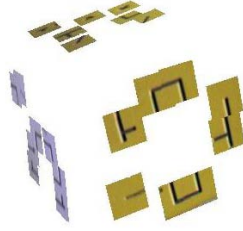After the textured polygon for each reference point has

Figure 7: *The offscreen representation of the reference model of the example cube. Shown here is the final result after all template images were rendered according to the inital infrared pose.*

been rendered, the actual template image for this point is read back from the frame buffer. A square 2D region of the frame buffer image centered at the location of the projected reference point p_i is used as the template image. During the rendering process, depth buffer tests are disabled, resulting in complete visibility for the last rendered textured square. The principle of reading back quadratic frame buffer areas centered at the projected reference points is illustrated in Figure 8.
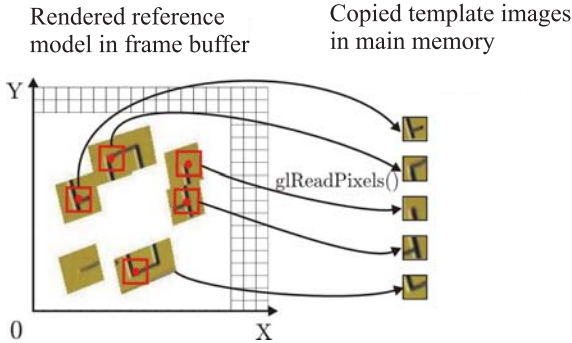

Figure 8: *After the textured square for each reference point has been rendered, the corresponding projected template image is read back from the frame buffer.*

Each final template image is read back into main memory using the OpenGL function `glReadPixels()`. The entire reference model is rendered into the currently invisible back buffer of the OpenGL doublebuffer. This rendered representation is later overwritten by the composed AR frame and is never visible to the user. This step of the algorithm can therefore be considered an offscreen rendering process.

5.3. Template Matching

After the offscreen rendering step, the correspondence point c_i is searched for each projected template image. This is done by defining a search area in the camera image which is also centered at the location of the projected reference point (p_i). Figure 9 shows a schematic overview of the template matching process. In the figure, t denotes the side length of the square template image, and s is the side length of the search area. Both s and t are user-definable parameters.
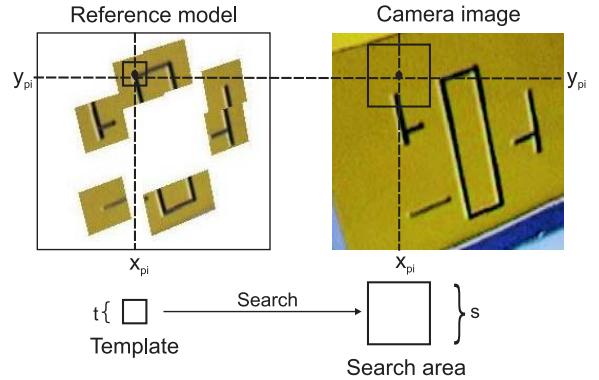

Figure 9: *Overview of the template matching process. A search area is defined which is centered at the projected reference point, p_i. The algorithm then looks for the respective correspondence point c_i in this portion of the camera image.*

Template matching is a common task in image processing, and various different approaches to template matching exist [Pra01]. In our hybrid tracking system, we use the so-called normalized correlation coefficient. A publicly available and speed-optimized implementation of this algorithm is provided by the OpenCV computer vision library, which is used for the basic image processing tasks in our system [Int01]. This template matching method uses single-channel images as input data. Therefore, we convert both the current camera image and the currently considered template image into gray images. In Equation 1, $r(x,y)$ is the computed normalized correlation coefficient at the pixel coordinates (x,y) in the camera image. The basis for the computation of the coefficient is a cross-correlation between $T(x,y)$, which is the pixel intensity of the template image, and $C(x,y)$, which is the pixel intensity of the camera image. As shown in the equation, the result of the correlation is divided by a term that accounts for the total intensity in the considered image area. This way, comparable correlation values are obtained for images of varying brightness.

$$r(x,y) = \frac{\sum_{y'=0}^{t-1} \sum_{x'=0}^{t-1} \widetilde{T}(x',y')\widetilde{C}(x+x',y+y')}{\sqrt{\sum_{y'=0}^{t-1} \sum_{x'=0}^{t-1} \widetilde{T}(x',y')^2 \widetilde{C}(x+x',y+y')^2}} \quad (1)$$

For each pixel in the search area, the summations in Equation 1 are evaluated over the area of the template image, $t \times t$.

The factors added up in the summations, $\widetilde{T}$ and $\widetilde{C}$, are indirectly derived from the pixel intensities. In order to make the resulting coefficient even more independent from varying brightness in the camera and template images, only the differences of the pixel intensities from the average intensity are considered. An average intensity $\overline{C}$ is computed for the currently regarded search area. Correspondingly, the average intensity $\overline{T}$ is calculated for the template image. The factors used in the computation of the normalized correlation coefficient are then determined as shown in Equation 2.

$$\begin{aligned} \widetilde{C}(x,y) &= C(x,y) - \overline{C} \\ \widetilde{T}(x,y) &= T(x,y) - \overline{T} \end{aligned} \qquad (2)$$

For each reference point, the normalized correlation coefficient is calculated over the entire associated search area. The coefficient specifies the similarity between the region centered at this position in the camera image and the projected template image. The greater the coefficient value, the greater is the similarity between both images. After the computation of the correlation coefficients, the location of the maximum similarity is determined. The result of this search consists of two pieces of information. The main result is the correspondence point c_i, which is considered to be the place in the camera image which corresponds to the projected reference point p_i. The second information gained from the search process is the maximum coefficient itself. This maximum coefficient, which we call the *confidence value* k_i, is also stored for later use.

5.4. Numerical Stability

After the correspondence points have been determined for all reference points, the actual computation of the improved camera pose can be performed. This pose computation, however, is embedded into several methods for increasing the stability of the numerical algorithm. At first, inadequate point correspondences are culled from the set of reference points. These are point correspondences with confidence values below a certain threshold, i.e., $k_i < k_{min}$. Here, k_{min} is a constant value, which can be defined by the user. Point correspondences with small confidence values represent invalid template matching results. These can be caused by various circumstances, e.g., if the search area is too small or if the real feature point is occluded in the camera image. Removing these invalid correspondences significantly improves the quality of the hybrid pose estimation process. Values between 0.8 and 0.9 have empirically proven to be good choices for k_{min}.

As a second measure for improving the numerical stability of the pose computation, all relevant point coordinates are normalized. This means that they are transformed into a coordinate system in which the average distance of points to the coordinate origin is $\sqrt{2}$ in 2D or $\sqrt{3}$ in 3D, respectively (see [HZ04]). Such a transformation is applied to both the reference points r_i and the correspondence points c_i. The effect of this normalization is that during the actual numerical computation, the range of occuring numerical values is relatively small. Therefore, negative effects caused by the limited accuracy of floating point variables are restricted.

Finally, the actual pose computation process is embedded in a random sample consensus algorithm (RANSAC) [FB81]. The RANSAC approach helps to minimize the impact of invalid point correspondences which have not been removed by the confidence threshold. There are several possible reasons for the occurrence of such invalid correspondences with high similarity values. Among the possible causes is the existence of repeated patterns in the image, excessively large camera movements, or inappropriately defined reference points.

5.5. Pose Estimation

The core of our hybrid tracking system is a pose estimation algorithm based on Newton's method for solving systems of non-linear equations. This pose estimation method is described in detail by Trucco and Verri [TV98]. The big advantage of this method is that it starts with an initial estimate for the pose. In our system, the acquired infrared pose, M_{IR}, is used as this initial estimate.

The algorithm uses a representation of pose information which consists of the translation from the coordinate origin, $T = (t_x, t_y, t_z)$, and the orientation expressed in Euler angles, $R = (\phi_x, \phi_y, \phi_z)$. The conversion between a transformation matrix and the representation as (R, T) can be performed with standard linear algebra methods.

Each iteration begins with the computation of the so-called residuals $(\delta x_i, \delta y_i)$. These are the difference vectors between the projections of the reference points according to the current pose estimation and the correspondence points. The calculation of the residuals is shown in Equation 3. In this equation, each reference point r_i is first transformed according to the current pose parameters, resulting in a transformed reference point $t_i = (x_{t_i}, y_{t_i}, z_{t_i})$. It is then projected into image space, producing a 2D point $q_i = (x_{q_i}, y_{q_i})$.

$$\begin{aligned} t_i &= R \cdot r_i + T \\ q_i &= proj(t_i) \\ \begin{pmatrix} \delta x_i \\ \delta y_i \end{pmatrix} &= q_i - c_i \end{aligned} \qquad (3)$$

Using Equation 3 and the definition of the projection function (see Sec. 4), the coordinates of the projected points q_i can be differentiated with respect to the translation vector T and the rotation angles R. According to [TV98], this differentiation results in the partial derivatives shown in Equation 4 for the translation components $T = (t_x, t_y, t_z)$.

$$\frac{\partial x_{q_i}}{\partial t_x} = \frac{f}{z_{t_i}}, \quad \frac{\partial x_{q_i}}{\partial t_y} = 0, \quad \frac{\partial x_{q_i}}{\partial t_z} = -f\frac{x_{t_i}}{z_{t_i}^2}$$

$$\frac{\partial y_{q_i}}{\partial t_x} = 0, \quad \frac{\partial y_{q_i}}{\partial t_y} = \frac{f}{z_{t_i}}, \quad \frac{\partial y_{q_i}}{\partial t_z} = -f\frac{y_{t_i}}{z_{t_i}^2} \tag{4}$$

The partial derivates with respect to the orientation angles (ϕ_x, ϕ_y, ϕ_z) are shown in Equation 5. In both equations, f denotes the focal length of the digital camera used in the AR system.

$$\frac{\partial x_{q_i}}{\partial \phi_x} = -\frac{f x_{t_i} y_{t_i}}{z_{t_i}^2}, \quad \frac{\partial x_{q_i}}{\partial \phi_y} = f\frac{x_{t_i}^2 + z_{t_i}^2}{z_{t_i}^2}, \quad \frac{\partial x_{q_i}}{\partial \phi_z} = -f\frac{y_{t_i}}{z_{t_i}}$$

$$\frac{\partial y_{q_i}}{\partial \phi_x} = -f\frac{z_{t_i}^2 + y_{t_i}^2}{z_{t_i}^2}, \quad \frac{\partial y_{q_i}}{\partial \phi_y} = f\frac{x_{t_i} y_{t_i}}{z_{t_i}^2}, \quad \frac{\partial y_{q_i}}{\partial \phi_z} = f\frac{x_{t_i}}{z_{t_i}} \tag{5}$$

The value of these partial derivatives is computed for each pair of transformed reference point and associated correspondence point (t_i, c_i). Using these values, an equation system is then set up for the unkowns $\Delta T = (\Delta t_x, \Delta t_y, \Delta t_z)$ and $\Delta R = (\Delta \phi_x, \Delta \phi_y, \Delta \phi_z)$. ΔT is the correction for the translation vector of the current pose estimation. Likewise, ΔR is the correction for the orientation angles of the current pose estimation. These corrections will later be applied to the current pose parameters in order to obtain an improved estimation.

For each correspondence point, the two equations shown in Equation 6 are set up. Here, A is the coefficient matrix of the equation system. The equations for all point pairs (t_i, c_i) are combined in a single large equation system. This equation system is normally over-determined. A minimum number of five point correspondences was empirically determined in order to compute a useful pose correction. We use the singular value decomposition for solving the equation system [TV98]. This yields the corrections ΔT and ΔR for the current iteration of the algorithm.

$$A = \begin{pmatrix} \frac{\partial x_{q_i}}{\partial t_x} & \frac{\partial x_{q_i}}{\partial \phi_x} & \frac{\partial x_{q_i}}{\partial t_y} & \frac{\partial x_{q_i}}{\partial \phi_y} & \frac{\partial x_{q_i}}{\partial t_z} & \frac{\partial x_{q_i}}{\partial \phi_z} \\ \frac{\partial y_{q_i}}{\partial t_x} & \frac{\partial y_{q_i}}{\partial \phi_x} & \frac{\partial y_{q_i}}{\partial t_y} & \frac{\partial y_{q_i}}{\partial \phi_y} & \frac{\partial y_{q_i}}{\partial t_z} & \frac{\partial y_{q_i}}{\partial \phi_z} \end{pmatrix}$$

$$A \cdot \begin{pmatrix} \Delta t_x \\ \Delta \phi_x \\ \Delta t_y \\ \Delta \phi_y \\ \Delta t_z \\ \Delta \phi_z \end{pmatrix} = \begin{pmatrix} \delta x_i \\ \delta y_i \end{pmatrix} \tag{6}$$

At the end of each iteration, the computed corrections are applied to the current pose estimation. The determined translation correction ΔT is subtracted from the translation vector T of the current pose, i.e., $T \leftarrow T - \Delta T$. Likewise, the orientation correction ΔR is applied to the current orientation. This is done by multiplying the rotation matrix corresponding to the current orientation with the corrections for the individual axes. This computation is shown in Equation 7. In this equation, M_R is the rotation matrix representing the orientation R, and $R_{\{x,y,z\}}(\alpha)$ denote matrices corresponding to a rotation around one of the coordinate axes.

$$M_R \leftarrow M_R \cdot R_x(-\Delta\phi_x) \cdot R_y(-\Delta\phi_y) \cdot R_z(-\Delta\phi_z) \tag{7}$$

The pose estimation algorithm can thus briefly be summarized as follows: Starting with M_{IR} as initial estimate, in each iteration **1.** compute the residuals $(\delta x_i, \delta y_i)$, **2.** for each point correspondence, calculate the partial derivatives and construct the coefficient matrix A, **3.** set up the complete equation system and solve it for ΔT and ΔR, and **4.** update the pose estimation with the obtained corrections. This is repeated until either a maximum number of iterations is reached or the average length of the residuals becomes smaller than a threshold. The finally obtained pose estimation is used as improved hybrid camera pose M_{Hyb} in the AR system.

6. Results

The presented hybrid tracking system was developed and tested with the aforementioned artificially textured cube object as reference model. Good results were obtained with the hybrid scheme, and a significantly improved pose estimation was achieved under most circumstances. Figure 10 demonstrates the effect of hybrid tracking system (also see Fig. 12(b) in the color plate). It clearly shows that the cube geometry rendered with the hybrid pose estimation corresponds significantly better to the location of the real cube in the camera image. The effect of the hybrid scheme on the projections of the reference points is shown in Figure 11. Again, the reference points projected with the hybrid pose are significantly closer to their real counterparts (locations of high-contrast corners).

Several experimental test runs were performed. Table 1 lists the parameters used for one typical test run. In this case, a reference model was used which consists of 23 reference points defined from three camera viewpoints. The experiment spanned a duration of 182 frames, during which the camera was moved relative to the example cube, but remained roughly pointed at the object.

Table 2 compares the tracking accuracies achieved with the infrared pose and the hybrid tracking. The pixel displacement listed in the table is the distance between the projected reference points and the associated correspondence points. For the infrared pose, this is $||p_i - c_i||$, for the hybrid pose
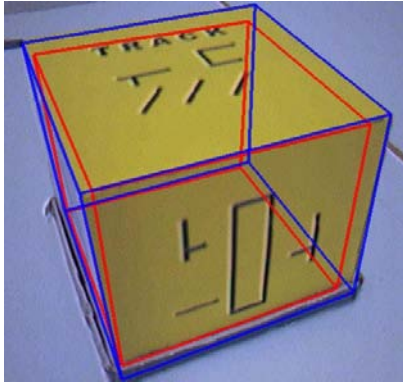
Figure 10: *Visualization of the effect of the hybrid tracking approach. The red (brighter) wireframe was rendered with the initial infrared pose, the blue (darker) wireframe with the improved hybrid pose.*
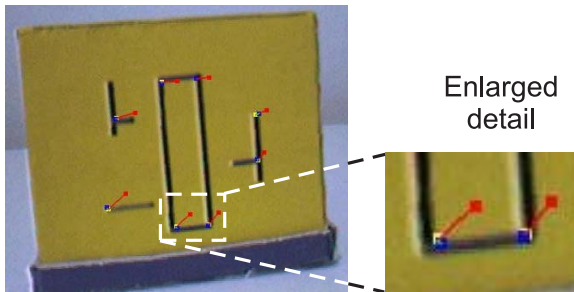

Enlarged detail

Figure 11: *The effect of the hybrid tracking scheme illustrated for the reference points. The red (brighter) dots were projected with the initial infrared pose and correspond to the p_i. They are connected with red lines to the blue (darker) projections of the reference points when the improved hybrid pose (M_{Hyb}) is used. (Also partially visible as yellow (bright) dots are the correspondence points c_i.)*

$||proj(M_{Hyb} \cdot r_i) - c_i||$. As shown in the table, the minimum average displacement per frame is significantly smaller for the hybrid pose (2.98 pixels) than for the infrared pose (6.4 pixels). Moreover, the measured overall average displacement is more than 30% less with hybrid tracking (7.89 pixels) than with pure infrared tracking (11.74 pixels).

Template side length t	16 pixels
Search area side length s	60 pixels
Confidence threshold k_{min}	0.9
Number of reference points (defined from 3 viewpoints)	23
Test duration	182 frames

Table 1: *Parameters of experimental test run.*

It has to be noted that it is the default behaviour of the hybrid tracking system to revert to the infrared pose estimate if the hybrid pose is considered to be invalid. This is the case if the average reference point displacement is too large. An invalid pose estimation can be caused by adverse environmental circumstances. These include excessively fast camera movements or rapid changes in the environment lighting, which cause the digital camera to deliver useless images. Another possible reason is a situation in which the reference object not visible or mostly occluded in the camera image. In the experiment, invalid poses were computed for three frames. These frames were also included in the statistics shown in Table 2.

	Infrared pose	Hybrid pose
Min. Ø displacement (average per frame)	6.4 pixels	2.98 pixels
Max. Ø displacement (average per frame)	18.94 pixels	12.88 pixels
Overall Ø displacement (average of all frames)	11.74 pixels	7.89 pixels
Overall Ø frame rate	20.7 fps	13.5 fps
Number invalid frames	-	3

Table 2: *Results of experimental test run.*

As shown in Table 2, the hybrid tracking system reduced the frame rate from more than 20 fps to 13.5 fps. Table 3 contains an analysis of the average runtime of the individual algorithm steps during the experiment. The major part of the hybrid pose estimation algorithm is required for the template matching step.

Offscreen rendering	17.86 msecs	(27%)
Template matching	36.92 msecs	(57%)
Pose estimation	10.18 msecs	(16%)

Table 3: *Runtime analysis of the individual algorithm steps.*

7. Conclusions

We have presented a hybrid tracking approach for medical augmented reality. The hybrid pose estimation scheme is based on a medical AR system which utilizes commercially available IGS equipment for infrared tracking. The proposed hybrid algorithm is capable of significantly improving the accuracy of graphical overlays in video see-through AR.

The hybrid tracking algorithm works stably expect in the case of adverse environmental conditions (see Sec. 6). However, the system can revert to the infrared pose if an invalid hybrid pose was computed. The hybrid tracker, which uses information from the camera image, has the typical limitations of vision-based systems. A low quality of the digital

image, an inadequate definition of the reference model, excessive camera motions, and too much occlusion in the image can have a negative impact on the tracking performance. However, we have found the hybrid pose estimation system to deliver useful output most of the time in our experiments.

The current implementation of the algorithm reduces the overall frame rate of the system. The most computationally complex algorithm step is the readback of template images using `glReadPixels()` and the subsequent template matching. This part of the method could be sped up by utilizing the programmability of modern GPUs. With appropriate fragment programs, tasks like template matching could be offloaded to the GPU, eliminating computations on the CPU and costly buffer readbacks.

The main drawback of the current implementation is the required manual definition of the reference model. A (semi)automatic acquisition of reference objects is a main topic of the future work. Moreover, the system should be tested in a more application-specific environment. Possibly, an adjusted tracking strategy could prove useful for medical scenarios, e.g., by using intraoperative registration fiducials as optical landmarks.

The current realization of the presented system is still in an experimental state, but it demonstrates the feasibility and usefulness of the approach. Medical application scenarios can benefit significantly from an improved accuracy of the camera pose estimation.

Acknowledgments

References

[ABB*01] AZUMA R., BAILLOT Y., BEHRINGER R., FEINER S., JULIER S., MACINTYRE B.: Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications 21*, 6 (November/December 2001), 34–47.

[BBR*03] BORNIK A., BEICHEL R., REITINGER B., SORANTIN E., WERKGARTNER G., LEBERL F., SONKA M.: EG2003 Medical Prize Competition: Augmented Reality based Liver Surgery Planning. *Computer Graphics Forum 22*, 4 (December 2003), 795–796.

[BFO92] BAJURA M., FUCHS H., OHBUCHI R.: Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient. In *Proc. of ACM SIGGRAPH* (July 1992), pp. 203–210.

[Bra04] BRAINLAB AG: Neurosurgery Product Brochure, 2004.

[FB81] FISCHLER M., BOLLES R.: Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM 24*, 6 (1981), 381–395.

[FBS04] FISCHER J., BARTZ D., STRASSER W.: Occlusion Handling for Medical Augmented Reality using a Volumetric Phantom Model. In *Proc. of ACM Symposium on Virtual Reality Software and Technology (VRST)* (November 2004), pp. 174–177.

[FBS05] FISCHER J., BARTZ D., STRASSER W.: Intuitive and Lightweight User Interaction for Medical Augmented Reality. In *Proc. of Vision, Modeling, and Visualization* (November 2005), pp. 375–382.

[FNFB04] FISCHER J., NEFF M., FREUDENSTEIN D., BARTZ D.: Medical Augmented Reality based on Commercial Image Guided Surgery. In *Proc. of Eurographics Symposium on Virtual Environments* (June 2004), pp. 83–86.

[FWBN05] FEUERSTEIN M., WILDHIRT S., BAUERNSCHMITT R., NAVAB N.: Automatic Patient Registration for Port Placement in Minimally Invasive Endoscopic Surgery. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (October 2005).

[HZ04] HARTLEY R., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[Int01] INTEL CORPORATION: *Open Source Computer Vision Library Reference Manual*, 2001.

[KB99] KATO H., BILLINGHURST M.: Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. In *Proc. of IEEE and ACM International Workshop on Augmented Reality (IWAR)* (October 1999), pp. 85–94.

[NBHM99] NAVAB N., BANI-HASHEMI A., MITSCHKE M.: Merging Visible and Invisible: Two Camera-Augmented Mobile C-arm (CAMC) Applications. In *Proc. of IEEE and ACM International Workshop on Augmented Reality (IWAR)* (October 1999), pp. 134–141.

[PATM03] PIEKARSKI W., AVERY B., THOMAS B., MALBEZIN P.: Hybrid Indoor and Outdoor Tracking for Mobile 3D Mixed Reality. In *IEEE and ACM International Symposium on Mixed and Augmented Reality Poster Proceedings* (October 2003), pp. 266–267.

[Pra01] PRATT W.: *Digital Image Processing*, 3rd ed. John Wiley & Sons, 2001.

[SHC*96] STATE A., HIROTA G., CHEN D., GARRETT W., LIVINGSTON M.: Superior Augmented-Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. In *Proc. of ACM SIGGRAPH* (August 1996), pp. 429–438.

[SKB*01] SAUER F., KHAMENE A., BASCLE B., SCHIMMANG L., WENZEL F., VOGT S.: Augmented Reality Visualization of Ultrasound Images: System Description, Calibration, and Features. In *Proc. of IEEE and ACM International Symposium on Augmented Reality (ISAR)* (October 2001), pp. 30–44.

[SSW02] SCHWALD B., SEIBERT H., WELLER T.: A Flexible Tracking Concept Applied to Medical Scenarios Using an AR Window. In *IEEE and ACM International Symposium on Mixed and Augmented Reality Poster Proceedings* (September 2002), pp. 261–262.

[TV98] TRUCCO E., VERRI A.: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, 1998.

[YNA99] YOU S., NEUMANN U., AZUMA R.: Hybrid Inertial and Vision Tracking for Augmented Reality Registration. In *Proc. of IEEE Virtual Reality* (March 1999), pp. 260–267.