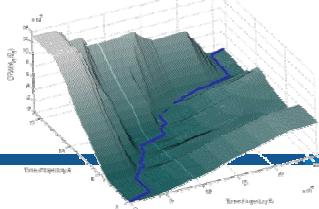
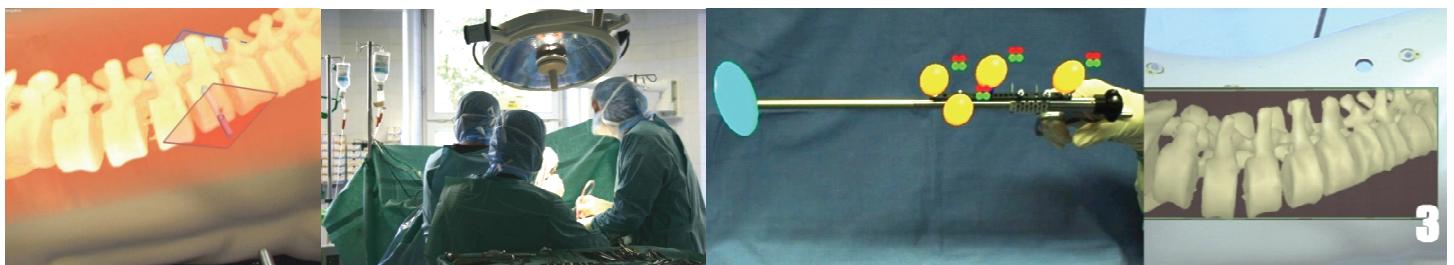


New Methods for Medical Augmented Reality

Dissertation



Tobias Sielhorst



Technische Universität München, Fakultät für Informatik
Lehrstuhl für Computeranwendungen in der Medizin, Univ.-Prof. N. Navab, Ph.D.

New Methods for Medical Augmented Reality

Tobias Sielhorst

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation.

Vorsitzende:

Univ.-Prof. G. J. Klinker, PhD

Prüfer der Dissertation:

1. Univ.-Prof. Dr. N. Navab, Ph.D.
2. Prof. H. Fuchs, Ph.D.

University of North Carolina, Chapel Hill, USA

Die Dissertation wurde am 25.10.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 25.02.2008 angenommen.

Abstract

The increasing amount of imaging data in the operating room offers new possibilities for surgeons. Preoperative data like Computed Tomography and Magnetic Resonance Imaging allow for detailed anatomical information. Recently, intraoperative imaging like X-ray C-arms and ultrasound have become increasingly accessible in the OR and provide real-time anatomical images. Also, functional imaging such as PET and fMRI is accessible in a growing number of institutions. Corresponding intraoperative functional probes allow more efficient procedures to be introduced in the near future.

The growing amount of imaging data increases the difficulty of retrieving the desired piece of information. Thus, efficient data representation becomes increasingly important for physicians. Visualization in computer assisted surgical solutions has not coped with the recent developments and does still not provide solutions which allow surgeons to take full advantage of the existing heterogeneous imaging data.

Augmented Reality (AR) is a technology that has potential to improve physicians' performance by bringing the multitude of medical data, surgical actions, and patient into the same space.

After more than a decade of research in medical Augmented Reality, the technology is sufficiently developed to create prototype systems for showrooms. However, it has still not found its way into operating rooms.

This thesis addresses current issues of medical Augmented Reality in defining the right place, right time, and the right way of data representation. The first contribution is in providing an exhaustive state-of-the-art report on medical Augmented Reality. It then features a detailed overview of components that are necessary for a medical AR system and introduces a new method for temporal registration in AR. It provides details on the integration of these components and describes the necessary software framework that has been developed. The thesis presents new methods for assessment of a medical AR system addressing latency measurement and dynamic error prediction. The thesis concludes with validation which was done in close partnership with many surgeons.

CONTENTS

1	Introduction	1
1.1	Overview of medical AR systems and technologies	2
1.1.1	Head-mounted display (HMD) based	3
1.1.2	Augmented optics	5
1.1.3	AR windows	6
1.1.4	Augmented monitors	8
1.1.5	Augmented endoscopes	8
1.1.5.1	Calibration and undistortion of wide angle optics	9
1.1.5.2	Tracking of flexible endoscopes	10
1.1.5.3	Endoscopy related visualization issues	10
1.1.6	Augmented medical imaging devices	11
1.1.7	Projections on the patient	13
1.2	Potential benefits of AR visualization	14
1.2.1	Extra value from image fusion	14
1.2.2	Implicit three-dimensional interaction	14
1.2.3	Three-dimensional visualization	15
1.2.4	Improved hand-eye coordination	15
1.3	The NARVIS Project	17
1.3.1	Targeted intervention: Minimal invasive spine fixation	18
1.3.2	Analysis of the surgery	18
1.3.2.1	Port placement	20
1.3.2.2	Laparoscopic access to target area	21
1.3.2.3	Implant screw placement	22
1.4	Contribution of the thesis	23
1.5	Document overview	23
2	Components	25
2.1	Spatial registration	25
2.1.1	Calibration	26
2.1.2	Tracking	27
2.1.3	Patient registration	28
2.2	Temporal registration	29

CONTENTS

2.2.1	Movement comparison	29
2.2.1.1	Vision	30
2.2.1.2	Setup	32
2.2.1.3	Comparison of trajectories	33
2.2.1.4	Comparison of values	34
2.2.1.5	How to match points	35
2.2.1.5.1	Longest Common Subsequence (LCSS)	35
2.2.1.5.2	Dynamic Time Warping (DTW)	38
2.2.1.6	Results of offline synchronization	41
2.2.1.7	Preliminary work on online synchronization	41
2.2.1.8	Discussion/Conclusion	44
2.2.2	Workflow recovery	45
2.2.2.1	Introduction and Related Work	45
2.2.2.2	Description of surgical workflow	46
2.2.2.3	Algorithms and methods	47
2.2.2.3.1	Principle of workflow recovery	47
2.2.2.3.2	Classical DTW	48
2.2.2.3.3	Creating an Average Surgery	48
2.2.2.3.4	Weighting of Signals From The Curves	49
2.2.2.4	Experiments and Results	50
2.2.2.4.1	Workflow Retrieval	50
2.2.2.4.2	Online temporal registration of workflow	51
2.2.2.5	Conclusion	52
2.3	Visualization	53
2.3.1	Depth perception	53
2.3.1.1	Misperception	54
2.3.1.2	Adaption	54
2.3.1.3	Motion sickness	54
2.3.2	Three-dimensional data representation	55
2.3.2.1	Slice rendering	55
2.3.2.2	Surface rendering	55
2.3.2.3	Volume Rendering	56
2.3.3	Selected medical visualization topics	57
2.3.3.1	In-situ visualization of endoscopic images	57
2.3.3.1.1	Depth reconstruction	58
2.3.3.1.2	Automatic horizon recovery	58
2.3.3.1.3	Fallback visualization	59
2.3.3.1.4	Results	59
2.3.3.1.5	Discussion and Conclusion	60
2.3.3.2	Medical navigation	61
2.4	Interaction	61

3 Integration	65
3.1 Requirements	65
3.2 Other AR frameworks	66
3.3 Framework design	68
3.3.1 Programming language and environment	68
3.3.2 Program structure	68
3.3.2.1 Micro kernel concept	68
3.3.2.2 Scheduling	69
3.3.2.3 Configuration	69
3.3.2.4 Overview of component types	70
3.3.2.5 Code organization	71
3.3.3 Real-time constraints	74
3.3.4 Synchronization issues	74
3.3.4.1 Time stamped ring buffers	75
3.3.4.2 Computer clocks	75
3.3.4.3 Determination of measurement time	75
3.3.4.4 Strategy for synchronization and timing of visualization .	76
3.3.5 Results and examples	77
3.4 Implementation of NARVIS	78
3.4.1 Software components	78
3.4.1.1 Spatial registration	78
3.4.1.1.1 Calibration	78
3.4.1.1.2 Tracking	79
3.4.1.1.3 Patient registration	80
3.4.1.2 Temporal registration	81
3.4.1.3 Visualization	81
3.4.1.4 Interaction	81
3.4.2 Hardware description	81
4 Assessment	85
4.1 Technical evaluation	86
4.1.1 Latency estimation	86
4.1.1.1 Introduction	86
4.1.1.2 Method	88
4.1.1.2.1 Time encoding	89
4.1.1.2.2 Time decoding	90
4.1.1.2.3 Computer vision	91
4.1.1.3 Results	91
4.1.1.4 Discussion	94
4.1.1.5 Conclusion	95
4.1.2 Error Propagation	95
4.1.2.1 Methods	97
4.1.2.1.1 From fiducial location error to tracking target error	97
4.1.2.1.2 Visualization setup	98

CONTENTS

4.1.2.2	Results	98
4.1.2.2.1	Measurement of the image error	99
4.1.2.2.2	From image error to fiducial error	99
4.1.2.2.3	Inclusion of visibility data	99
4.1.2.2.4	Occlusion of Single Fiducials	100
4.1.2.2.5	Occlusion of Cameras	100
4.1.2.3	Conclusion	101
4.1.3	End-to-end accuracy	102
4.2	Usability	102
4.2.1	Clinical integrability	103
4.2.2	Navigation: 6D Guidance	103
4.2.3	Navigation: Classic interface versus AR interface	105
4.2.4	Depth Perception	105
4.2.4.1	Methods	106
4.2.4.1.1	Description of the evaluation procedure	106
4.2.4.1.2	Description of the tests	106
4.2.4.1.3	Evaluated visualization modes	107
4.2.4.2	Results	108
4.2.4.3	Conclusion	110
5	Conclusion	113
References		115
Authored and coauthored publications		135

CHAPTER
ONE

INTRODUCTION

Medical augmented reality takes its main motivation from the need of visualizing medical data and the patient within the same physical space. This goes back to the vision of having x-ray vision, seeing through objects. This would require real-time in-situ visualization of co-registered heterogeneous data. This was probably the goal of many medical augmented reality solutions proposed in literature. As early as 1938, Steinhaus [161] suggested a method for visualizing a piece of metal inside tissue registered to its real view. The method was based on the geometry of the setup and the registration and augmentation was guaranteed by construction. In 1968, Sutherland [168] suggested a tracked head-mounted display as a novel human-computer interface enabling viewpoint-dependent visualization of virtual objects. His visionary idea and first prototype were conceived at a time when computers were commonly controlled in batch mode rather than interactively. It was only two decades later that the advances in computer technology allowed scientists to consider such technological ideas within a real-world application. It is interesting to note that this also corresponds to the first implementation of a medical augmented reality system proposed by Roberts et al. [133] in 1986. They developed a system integrating segmented computed tomography (CT) images into the optics of an operating microscope. After an initial interactive CT-to-patient-registration, movements of the operating microscope were measured using an ultrasonic tracking system. Early 1990s augmented reality was also considered for other applications including industrial assembly [30], paperless office [103], and machine maintenance [42].

While virtual reality (VR) aimed at immersing the user entirely into a computer-generated virtual world, augmented reality (AR) took the opposite approach, in which virtual computer generated objects, were added to real physical world [187]. Within their so-called virtuality continuum [112], Milgram and Kishino described AR as a mixture of virtual reality (VR) and the real world in which the real part is more dominant than the virtual one. Azuma described AR by its properties of aligning virtual and real objects, and running interactively and in real-time [4, 5].

In Augmented reality inheres the philosophy that intelligence amplification (IA) of a user has more potential than artificial intelligence (AI) [26], because human experience and intuition can be coupled by the computational power of computers.

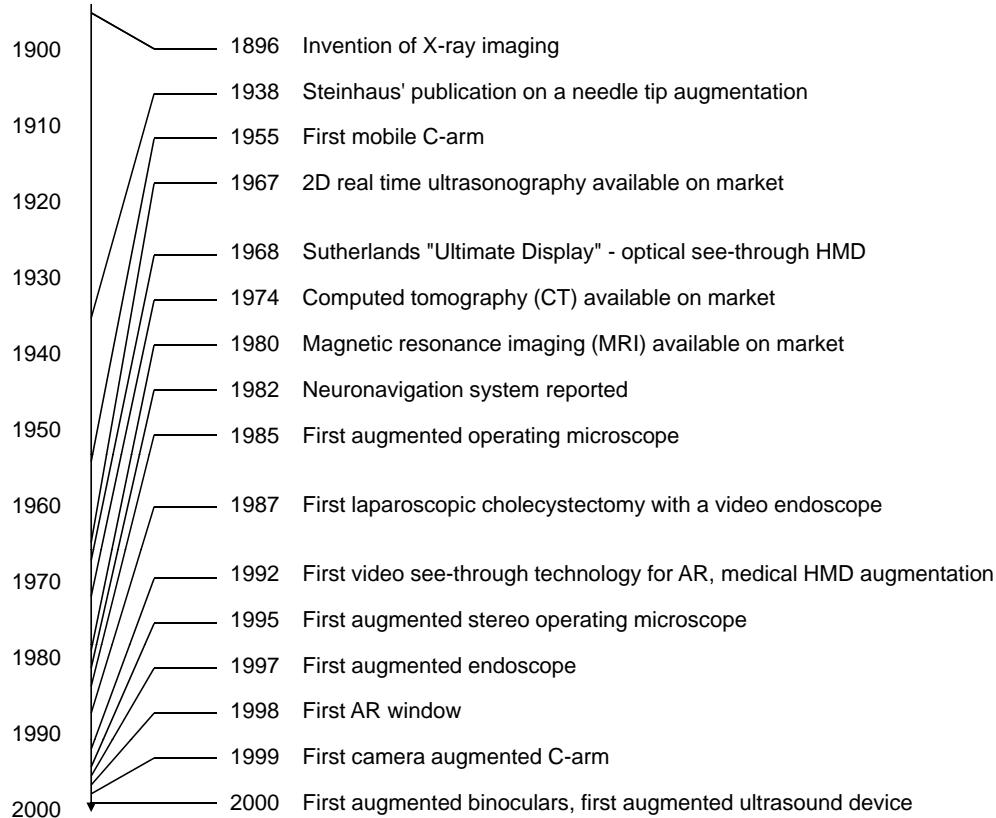


Figure 1.1: Inventions timeline of selected imaging and AR technology

1.1 Overview of medical AR systems and technologies

Different technologies have been proposed that allow for medical augmentation of images. In this section different of these AR technologies will be introduced including their specific limitations and advantages.

The first setup augmenting imaging data registered to an object was described in 1938 by the Austrian mathematician Steinhaus [161]. He described the geometric layout to reveal a bullet inside a patient with a pointer that is visually overlaid on the invisible bullet. This overlay was aligned by construction from any point of view and its registration works without any computation. However, the registration procedure is cumbersome and it has to be repeated for each patient. The setup involves two cathodes that emit X-rays projecting the bullet on a fluoroscopic screen (see figure 1.3). On the other side of the X-ray screen, two spheres are placed symmetrically to the X-ray cathodes. A third sphere is fixed on the crossing of the lines between the two spheres and the two projections of the bullet on the screen. The third sphere represents the bullet. Replacing the screen with a semi-transparent mirror and watching the object through the mirror, the third sphere is overlaid exactly on top of the bullet from any point of view. This is possible because the

	HMD based	Augmented optics	AR windows	Augmented monitors	Augmented endoscopes	Tomographic reflection	Projection on the patient
Improved hand eye coordination	x	x	x			x	x
Extra value from image fusion	x	x	x	x	x	x	x
Implicit 3D interaction	x	x	x				
Stereoscopic visualization	x	x	x		in rare cases	only in plane	
Multiuser capability	additional AR device	additional AR device	limited	x	x	x	limited

Figure 1.2: Simplified relationship between technology and potential benefits. Grey color indicates in-situ visualization.

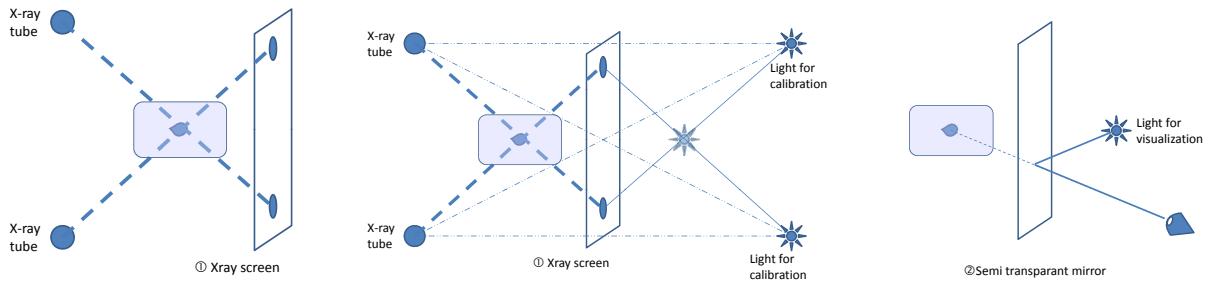


Figure 1.3: Calibration (left and middle) and visualization setup proposed by Steinhaus [161] in 1938.

third sphere is at the location to which the bullet is mirrored. Therefore, the setup yields stereoscopic depth impression. The overlay is restricted to a single point and the system has to be manually calibrated for each augmentation with the support of an X-ray image with two X-ray sources.

1.1.1 Head-mounted display (HMD) based

The first HMD-based AR system was described by Sutherland [168] in 1968. A stereoscopic monochrome HMD combined real and virtual images by means of a semi-transparent mirror. This is also referred to as optical see-through HMD. The tracking was performed mechanically. Research on this display was not application driven, but aimed at the “ultimate display” as Sutherland referred to it.

Bajura et al. [6] reported in 1992 on their video see-through system for the augmentation of ultrasound images. The system used a magnetic tracking system to determine the

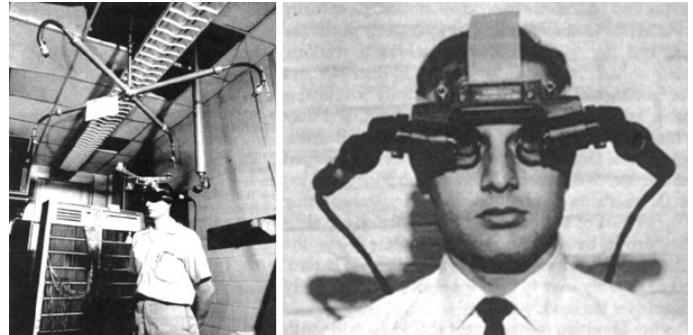


Figure 1.4: The first (optical see-through) HMD by Ivan Sutherland [168]. ©1968 IEEE.

pose of the ultrasound probe and HMD. The idea of augmenting live video instead of optical image fusion appears counterproductive at first sight since it reduces image quality and introduces latency for the real view. However, by this means the real view can be controlled electronically resulting in the following advantages:

1. No eye-to-display calibration is needed, only the camera-to-display transformation has to be calculated, which may remain fixed.
2. Arbitrary merging functions between virtual and real objects are possible as opposed to brightening up the real view by virtual objects in optical overlays. Only video overlay allows for opaque virtual objects, dark virtual objects, and correct color representation of virtual objects.
3. By delaying the real view until the data from the tracking system is available, the *relative* lag between real and virtual objects can be eliminated as described by Bajura et al. [7] and Jacobs et al. [74].
4. For the real view, the image quality is limited by the display specifications in a similar way as it is for the rendered objects. Since the color spectrum, brightness, resolution, accommodation, field of view, etc. are the same for real and virtual objects, they can be merged in a smoother way than for optical overlays.
5. The overlay is not user dependent, since the generation of the augmentation is already performed in the computer, as opposed to the physical overlay of light in the eye. The resulting image of an optical see-through system is in general not known. A validation without interaction is hardly possible with optical overlays.

In 1996, continuing of the work of Bajura et al. [6, 7], State et al. [159] reported on a system with 10 frames per second (fps) creating VGA output. This system facilitates hybrid magnetic and optical tracking and offers higher accuracy and faster performance than the previous prototypes. The speed was mainly limited by the optical tracking hardware. Nowadays, optical tracking is fast enough to be used exclusively.

In 2000, Sauer and colleagues [143] presented a video see-through system that allowed for a synchronized view of real and virtual images in real-time, i.e. 30 frames per second. In order to ensure that camera images and tracking data are from exactly the same point of time the tracking camera and the video cameras are genlocked, i.e. the tracking system shutter triggers the cameras. Their visualization software waits for the calculated tracking data before an image is augmented. This way, the relative lag is reduced to zero without interpolating tracking data. The system uses inside-out tracking, which means that the tracking system is placed on the HMD to track a reference frame rather than the other way around (see figure 1.5). This way of tracking allows for very low reprojection errors since the orientation of the head can be computed in a numerically more stable way than by outside-in tracking using the same technology [72].

Rolland and Fuchs [134] give a detailed discussion about the advantages and shortcomings of optical and video see-through technology. Cakmaci and Rolland [28] provide a recent and comprehensive overview of HMD designs.



Figure 1.5: Video see-through HMD without relative lag [143]. Left image from [178], ©2003 IEEE.

1.1.2 Augmented optics

Operating microscopes and operating binoculars can be augmented by inserting a semi transparent mirror in the optics. The mirror reflects the virtual image into the optical path of the real image. This allows for high optical quality of real images without further eye-to-display calibration, which is one of the major issues of optical see-through augmentation. Research on augmented optics evolved from stereotaxy in brain surgery in the early 1980s that brought the enabling technology together as for instance described by Kelly [81].

The first augmented microscope was proposed by Roberts et al. [133, 68] showing a segmented tumor slice of a computed tomography data set in a monocular operating microscope. This system can be said to be the first operational medical AR system. Its application area was interventional navigation. The accuracy requirement for the system was defined to be 1mm [53] in order to be in the same range as the CT slice thickness. An average error of 3mm [53] was measured for reprojection of contours, which is a remarkable result for the first system. However, the ultrasonic tracking did not allow for real-time data acquisition. A change in position of the operating microscope required approximately twenty seconds for acquiring the new position.

In 1995, Edwards et al. [127] published on their augmented stereoscopic operating microscope for neurosurgical interventions. It allowed for multicolor representation of segmented 3D imaging data as wireframe surface models or labeled 3D points. The interactive update rate of $1 - 2\text{Hz}$ was limited by the infrared tracking system. The accuracy of $2 - 5\text{mm}$ is in the same range as the system introduced by Friets et al. [53]. In 2000, the group reported on an enhanced version [84] with submillimeter accuracy, which was evaluated in phantom studies as well as clinical studies for maxillofacial surgery. The new version also allows for calibration of different focal lengths to support variable zoom level settings during the augmentation.

For ophthalmology, Berger et al. [14] suggest augmenting angiographic images into a biomicroscope. The system uses no external tracking but image-based tracking, which is possible because the retina offers a relatively flat surface that is textured with visible



Figure 1.6: Augmented binoculars by Birkfellner et al. [19]. ©2002 IEEE.

blood vessel structures. According to the authors, the system offers an update rate of $1-5\text{Hz}$ and an accuracy of 5 pixels in the digital version of the microscope image.

Birkfellner and colleagues have developed an augmented operating binocular for maxillofacial surgery in 2000 [19, 20]. It enables augmentation employing variable zoom and focus and customizable eye distances [49]. As opposed to the operating microscopes that are mounted on a swivel arm, an operating binocular is worn by the user.

A drawback of augmented optics in comparison with other augmentation technology is the process of merging real and computed images. As virtual images can only be added and may not entirely cover real ones, certain graphical effects cannot be realized. The impact of possible misperception is discussed in paragraph 2.3.1. Additionally, the relative lag between the visualization of real and virtual images cannot be neglected for head-worn systems (cf. Holloway [73]).

In addition to the superior imaging quality of the real view, a noteworthy advantage of augmented optics is a seamless integration of its technology into the surgical workflow. The augmented optics can be used as usual if the augmentation is not desired. Furthermore, the calibration or registration routine in the operating room need not be more complicated than for a navigation system.

1.1.3 AR windows

The third type of devices that allow for in-situ visualization is an AR window. In 1995, Masutani et al. [107] presented a system with a semi-transparent mirror that is placed between the user and the object to be augmented. The virtual images are created by an autostereoscopic screen with integral photography technology. With microlenses in front of an ordinary screen, different images can be created for different viewing angles. This reduces either the resolution or limits the effective viewing range of the user. However, no tracking system is necessary in this setup to maintain the registration after it is initially established. The correct alignment is independent on the point of view. Therefore, these autostereoscopic AR windows involve no lag when the viewer is moving. The first system

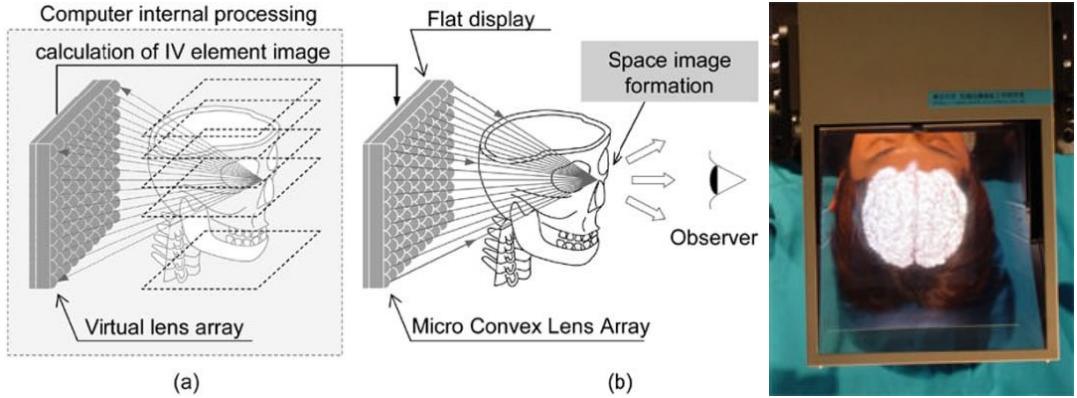


Figure 1.7: Concept of integral videography based augmentation [96]. ©2004 IEEE.

could not compute the integral photography dynamically. It had to be precomputed for a certain data set.

In 2002 Liao et al. [97, 96] proposed a medical AR window based on integral videography that could handle dynamic scenes. The authors realized the system for a navigation scenario, in which the position of an instrument was supposed to be visualized in the scene. Their algorithm performed the recalculation of a changed image in less than a second.

Blackwell et al. [21] presented in 1998 an AR window using a semi-transparent mirror for merging the real view with virtual images from an ordinary monitor. This technology requires tracked shutter glasses for the correct alignment of augmented objects and stereo vision, but it can handle dynamic images for navigation purposes at a high resolution and update rate.

For in-situ visualization, AR windows seem to be a perfect match to the operating room at first sight. For ergonomic and sterility reasons it is a good idea not to make surgeons wear a display. There are different ways of realizing AR windows. In detail, each one introduces a trade-off: Autostereoscopic displays suffer from poorer image quality in comparison with other display technologies. In principle, they offer a visualization for multiple users. However, this feature introduces another trade-off regarding image quality.

Display technology using shutter glasses need cables for trigger signals and power supply. Polarization glasses, as for instance used in the system introduced by Goebbel et al. [61], do not need cables and weigh less than an HMD, but limit the viewing angle of the surgeon to match the polarization. Non-autostereoscopic AR windows need to track the position of the user's eye in addition to the position of the patient and the AR window. This introduces another source of error.

Wesarg et al. [192] suggest a monoscopic AR window based on a transparent display. The design offers a compact setup, since no mirror is used, and no special glasses are

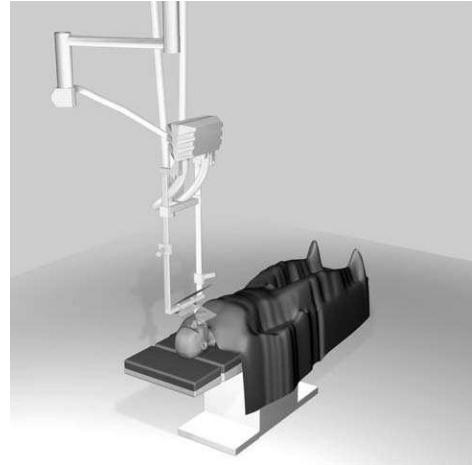


Figure 1.8: AR window with polarization glasses [61]. ©2003 IEEE.

required. However, it cannot display stereoscopic images and only one eye can see a correct image overlay. Since no mirror is used, the foci of the virtual and real image are at completely different distances.

All AR window designs have to take care of distracting reflections from different light sources. Last but not least, the display must be placed between the patient and the viewer. This may obstruct the surgeons' working area.

We believe that an optimal in-situ visualization device could consist of combination of an AR window and an HMD; an example may be an HMD attached to a boom.

1.1.4 Augmented monitors

In this paragraph, we cluster all systems that augment video images on ordinary monitors. The point of view is defined by an additional tracked video camera. In 1993, Lorensen and Kikinis [100] published their live video augmentation of segmented MRI data on a monitor. This initial system did not include tracking of the video camera yet. The camera-to-image registration had to be performed manually. The successor of this setup included a vision-based tracking system with fiducial markers [62].

Sato et al. [141] visualize segmented 3D ultrasound images registered to video camera images on a monitor for image guidance of breast cancer surgery. Nicolau et al. [121] describe a camera-based AR system using markers that are detected in the camera image. The system aims at minimally invasive liver ablation.

As an advantage of augmented monitors, users need not wear an HMD or glasses. By definition augmented monitors do, however, not offer in-situ visualization nor stereoscopic vision. Using them adds a tracked camera to the clinical setup.

1.1.5 Augmented endoscopes

A separate paragraph is dedicated to endoscope augmentation although it might be considered as a special case of monitor-based augmented reality or augmented imaging devices (see section 1.1.6). In contrast to augmented imaging devices endoscopic images need a tracking system for augmentation. As opposed to monitor-based AR the endoscopic setup already contains a camera, the integration of augmented reality techniques does not introduce additional hardware into the workflow of navigated interventions. A lot of investigative work has been carried out that dealt specifically with endoscopic augmentation.

The first usage of endoscopes as telescopic instruments utilizing a light source dates back to the 19th century. Endoscopy was mainly dedicated to diagnosis until the invention of video-based systems in the 1980s. Video endoscopy permits different team members to see the endoscopic view simultaneously. With this approach, it is possible for an assistant to position the endoscope while the operating surgeon can use both hands for the procedure. This feature opened the field of endoscopic surgeries. The removal of the gallbladder was one of the first laparoscopic surgeries. This operation also became a standard minimally invasive procedure. Since then, endoscopy has been successfully introduced into other surgical disciplines. Comprehensive literature reviews on the history of endoscopy can, for instance, be found in [92, 99], and [13].

Although endoscopes augmentation seems to be a straightforward step it has been realized as recent as the end of the 1990s by Freysinger et al. [52] for ear, nose and throat (ENT) surgery and Shahidi and colleagues [149] for brain surgery. Scholz et al. presented a navigation system for neurosurgery based on processed images [147]. Shahidi and Scholz use infrared tracking technology and a rigid endoscope while Freysinger's system uses magnetic tracking.

Mourgues et al. [115] describe endoscope augmentation in a robotic surgery system. The tracking is done implicitly by the robot since the endoscope is moved by the robots arm. Therefore no additional tracking system is necessary.

For endoscopic augmentation, the issues of calibration, tracking, and visualization are partly different than for other types of AR devices:

1.1.5.1 Calibration and undistortion of wide angle optics

Because of their wide-angle optics, endoscopes suffer from a noticeable image distortion. If a perfect, distortion-free pinhole camera model is assumed for superimposition, a particular source of error in the augmented image will be introduced [82]. This issue can be neglected in other AR systems with telephoto optics. Common types of distortion are radial distortion (also referred to as barrel distortion) and tangential distortion. Either the endoscope image has to be undistorted or the rendered overlay has to be distorted to achieve a perfect superimposition. While first approaches [158] required several minutes to undistort a single endoscope image, this process can now be completed in real-time: De Buck et al. [36] undistort sample points in the image and map a texture of the endoscope image on the resulting tiles; Shahidi et al. [148] precompute a look-up table (LUT) for each pixel for real-time undistortion.

In order to model the geometry of an endoscope camera, the intrinsic camera parameters focal length and principal point need to be determined. This can be achieved using well-established camera calibration techniques [69, 173, 197]. Most systems assume the focal length of an endoscope camera to be kept constant, although many endoscopes incorporate zoom lenses to change it intraoperatively, invalidating a certain calibration. Stoyanov et al. suggest to automatically adjust the calibration for intraoperative changes of the focal length of a stereoscopic camera [165]. Even though models for the calibration of monoscopic cameras with zoom lenses exist [195], they are not easily applicable to endoscopes. These models require the (preferably automatic) determination of the physical ranges for the lens settings e.g. in terms of motor units. However, the zoom settings of endoscopes are usually adjusted manually, rather than by a precise motor.

To obtain a rigid transformation from the camera coordinate frame to the coordinate frame of an attached tracking body or sensor, most authors employ hand-eye calibration techniques [15, 115, 120, 144, 146]. An alternative approach makes use of a tracked calibration pattern, whose physical coordinates are known with respect to the tracker [36, 105, 148].

In certain applications oblique-viewing endoscopes are used, for which the viewing directions are changeable by rotating the scope cylinder. Yamaguchi et al. developed a calibration procedure for such endoscopes [196].

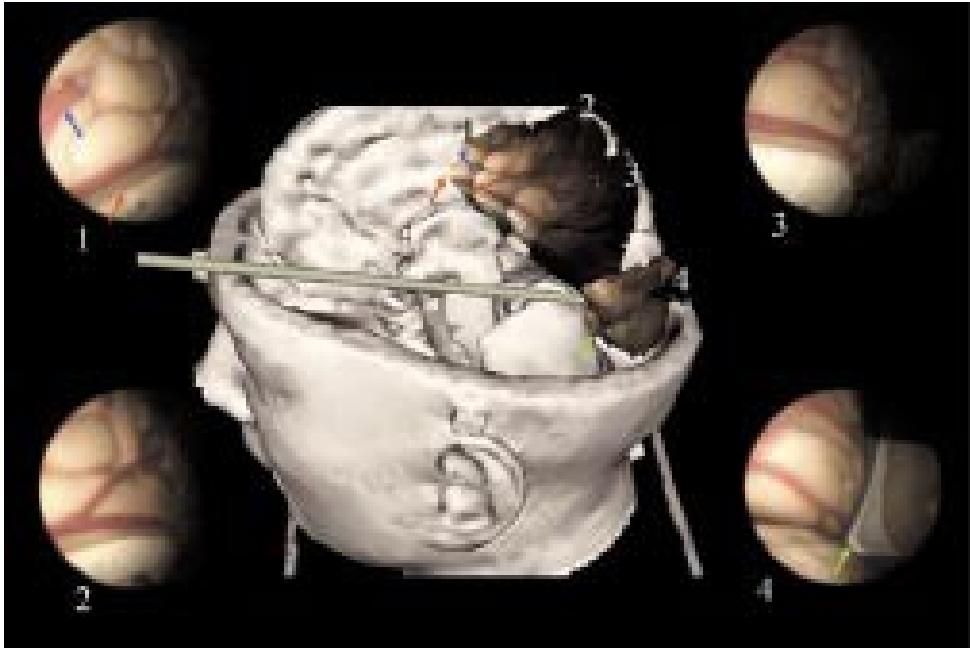


Figure 1.9: Context sensing by texturing segmented model by Dey et al. [37]. ©2002 IEEE.

1.1.5.2 Tracking of flexible endoscopes

Non-rigid endoscopes cannot be tracked by optical tracking systems. Bricault et al. [25] describe the registration of bronchoscopy and virtual bronchoscopy images using only geometric knowledge and image processing. The algorithms employed did not have real-time capability, however, they proved to be stable when used on recorded videos. In contrast to Bricault's shape from shading approach, Mori et al. [114] use epipolar geometry for image processing. In order to improve the performance of their registration algorithm they suggest the addition of electromagnetic tracking of the bronchoscope [113]. To achieve a fusion of the bronchoscopic video with a target path, Wegner et al. restrict electromagnetic tracking data onto positions inside a previously segmented bronchial tree [186]. Some groups, for instance Klein et al. [85], use electromagnetic tracking exclusively.

1.1.5.3 Endoscopy related visualization issues

The augmentation of endoscopic data does not only entail fusion with other imaging data. Konen et al. [147] suggest several image-based methods with a tracked endoscope to overcome typical limitations, such as replay of former images in case of loss of sight, image mosaicing, landmark tracking, and recalibration with anatomical landmarks. Krueger et al. [88] evaluate endoscopic distortion correction, color normalization and temporal filtering for clinical use.

One of the reasons for augmenting endoscope images is providing the anatomical context since the point of view and the horizon are changing. Recovering each of these issues requires a heightened level of concentration from the surgeon since their field of view

is very limited and the operating surgeon generally does not move the endoscope personally. Fuchs et al. [55] suggest provision of anatomical context by visualizing laparoscopic images in-situ with a head-mounted display. The necessary three-dimensional model of the surface as seen by the laparoscope is created with a pattern projector. Dey et al. [37] project endoscope images on segmented surfaces for providing context and creating endoscopic panorama images (see fig. 1.9). Kawamata et al. [80] visualize the anatomical context by drawing virtual objects in a larger area of the screen than endoscope images are available. Ellsmere and colleagues [41] suggest augmenting laparoscopic ultrasound images into CT slices and using segmented CT data for improved context sensing.

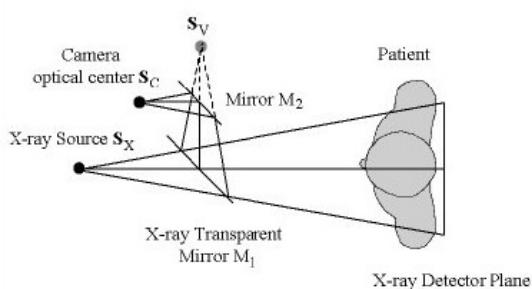
1.1.6 Augmented medical imaging devices

Augmented imaging devices can be defined as imaging devices that allow for an augmentation of their images without a tracking system. The alignment is guaranteed by their geometry.

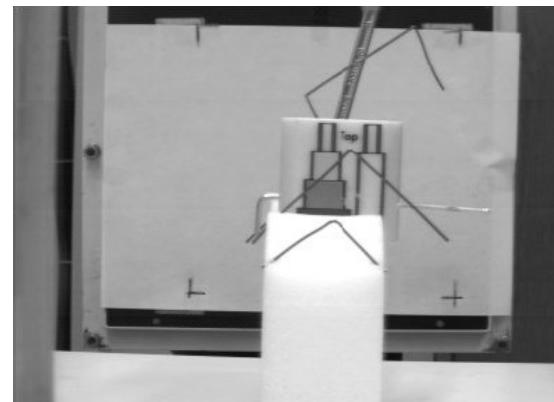
A construction for the overlay of fluoroscopic images on the scene has been proposed by Navab et al. [119] in 1999. An ordinary mirror is inserted into the X-ray path of a mobile C-arm. By this means it is possible to place a video camera that records light following the same path as the X-rays. Thus it is possible to register both images by estimating the homography between them without spatial knowledge of the objects in the image. The correct camera position is determined once during the construction of the system. The homography is estimated only once if the camera is not moved. For image fusion, one image can be transformed electronically to match the other using the estimated homography. The system provides augmented images without continuous X-ray exposure for both patient and physician. The overlay is correct until the patient moves relatively to the fluoroscope.

Tomographic reflection is a subgroup of augmented imaging devices. In 2000, Masamune and colleagues [48, 106] proposed an image overlay system that displays CT slices in-situ. A semi-transparent mirror allows for a direct view on the patient as well as a view on the aligned CT slice. The viewer may move freely while the CT slice remains registered without any tracking. The overlaid image is generated by a screen that is placed on top of the imaging plane of the scanner. The semi-transparent mirror is placed in the plane that halves the angle between the slice and the screen. The resulting overlay is correct from any point of view up to a similarity transform that has to be calibrated during the construction of the system. The system is restricted to a single slice per position of the patient. For any different slice, the patient has to be moved on the bed. Fichtinger et al. have extended this principle to magnetic resonance imaging in 2006 [50].

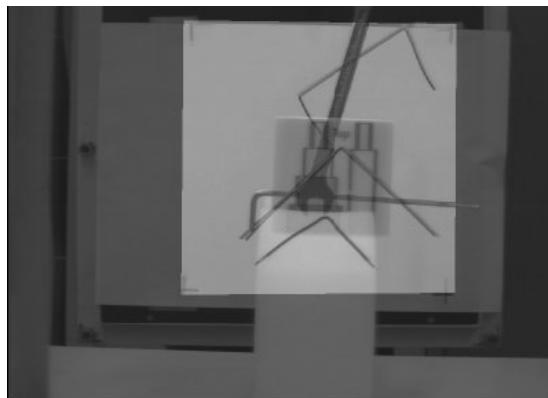
A similar principle has been applied to create an augmented ultrasound echography device. Stetten et al. [163, 162] proposed in 2000 the overlay of ultrasound images on the patient with a semi-transparent mirror and a little screen that is attached to the ultrasound probe. The mirror is placed on the plane that halves the angle between the screen and the B-scan plane of ultrasonic measurements. Similarly to the reflection of CT or MRI slices, it allows for in-situ visualization without tracking. In addition to real-time images, it allows for arbitrary slice views, as the ultrasound probe can be freely moved.



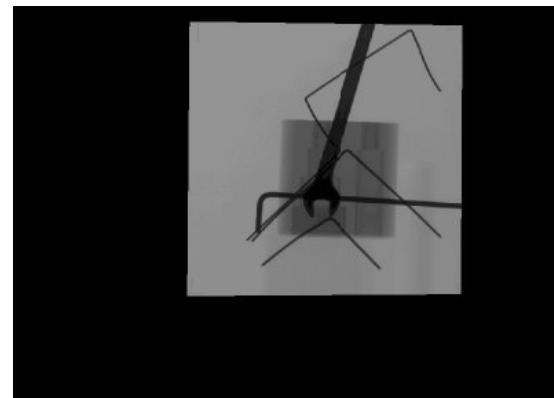
(a) Principle of CamC (CamC)



(b) Camera image



(c) fused image



(d) Fluoroscopic X-ray image

Figure 1.10: Camera-augmented c-arm (CamC) [119]. ©1999 IEEE.

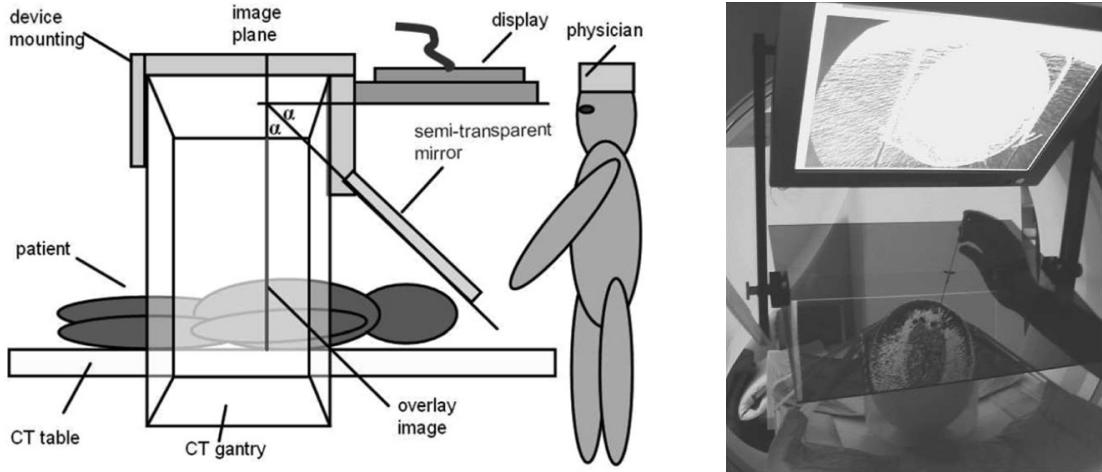


Figure 1.11: CT reflection [48]: Concept and prototypical setup. ©2005 IEEE.

1.1.7 Projections on the patient

Lastly, we present systems augmenting data directly onto the patient. The advantage of these systems is that the images are generally visible in-situ without looking through an additional device such as glasses, HMD, microscope, loupes, etc. As another beneficial feature, the user need not be tracked if visualization is meant to be on the skin rather than beneath. This also means that such a visualization can be used for multiple users. The simplicity of the system introduces certain limitations as a compromise, though.

Glossop et al. [60] suggested in 2003 a laser projector that moves a laser beam into arbitrary directions by means of controlled mirrors. Trajectories of the laser appear as lines due to the persistence of vision effect. The images are limited to a certain number of bright monochrome lines or dots and non-raster images. The system also includes an infrared laser for interactive patient digitization.

Sasama et al. [140] use two lasers for mere guidance. Each of these lasers creates a plane by means of a moving mirror system. The intersection of both planes is used to guide laparoscopic instruments in two ways. The intersecting lines or the laser on the patient mark the spot of interest, for instance an incision point. The laser planes can also be used for determining an orientation in space. The system manipulates the two laser planes in such a way that their intersecting line defines the desired orientation. If both lasers are projected in parallel to the instrument, the latter has the correct orientation. The system can only guide instruments to points and lines in space but it cannot show contours or more complex structures.

1.2 Potential benefits of AR visualization

Medical AR is a disruptive visualization technology. Clayton Christensen popularized the term disruptive technology as opposed to sustaining technology [31] in his book on economy of innovation. He denominates sustaining technology as the established technology that is improved iteratively providing a sustaining improvement rate over a certain period. In contrast, disruptive technology is not an enhancement of the prevalent technology, but a new development. It may consist of well-known parts combined in a new way. Christensen states that disruptive innovations may outperform the established technologies if the disruptive one offers properties that the established one cannot offer. The noteworthy part is that this can be true although the additional properties were not considered to be crucial and even though the disruptive technology had initially poorer specifications in the key requirements. This counterintuitive statement can be explained with the new possibilities of the disruptive technology enabling a new way of usage. Therefore, a crucial question regarding new visualization paradigms is “What can it do for us that established technology cannot?”.

The vague statement that AR provides an intuitive human computer interface can be found often. The following subsections deal with the base phenomena of AR that may lead to an intuitive visualization.

1.2.1 Extra value from image fusion

Fusing registered images into the same display offers the best of two modalities in the same view.

An extra value provided by this approach may be a better understanding of the image by visualizing anatomical context that has not been obvious before. This is the case for endoscopic camera and ultrasound images, where each image corresponds only to a small area. (See paragraph 1.1.5.3)

Another example for additional value is displaying two physical properties in the same image that can only be seen in either of the modality. An example is the overlay of beta probe activity. Wendler et al. [190] augment the doctors with visualizing previously measured activity emitted by radioactive tracers. By this means, physicians can directly relate the functional tissue information to the real view showing the anatomy and instrument position.

A further advantage concerns the surgical workflow. Currently, each imaging device introduces another display into the operating room (see fig. 1.12(b)); thus the staff spends valuable time on finding a useful arrangement of the displays. A single display integrating all data could solve this issue. Each imaging device also introduces its own interaction hardware and graphical user interface. A unified system could replace the inefficient multitude of interaction systems.

1.2.2 Implicit three-dimensional interaction

Interaction with three-dimensional data is a cumbersome task with two-dimensional displays and two-dimensional interfaces (cf. Bowman [24]). Currently, there is no best



(a) Action takes place on a very different position than the endoscope display (b) Each imaging device introduces another display

Figure 1.12: Current minimally invasive spine surgery

practice for three-dimensional user interfaces as opposed to two-dimensional interfaces using the WIMP paradigm (windows, icons, mouse, pointers).

AR technology facilitates implicit viewpoint generation by matching the viewport of the eye/endoscope on real objects to the viewport on virtual objects. Changing the eye position relative to an object is a natural approach for three-dimensional inspection.

Three-dimensional user interfaces reveal their power only in tasks that cannot be easily reduced to two dimensions, because two-dimensional user interfaces benefit from simplification by dimension reduction and the fact that they are widespread. Recent work by Traub et al. [22] suggests that navigated implant screw placement is a task that can benefit from three-dimensional user interaction, as surgeons were able to perform drilling experiments faster with in-situ visualization compared to a navigation system with a classic display.

1.2.3 Three-dimensional visualization

Many augmented reality systems allow for stereoscopic data representation. Stereo disparity and motion parallax due to viewpoint changes (see 1.2.2) can give a strong spatial impression of structures.

In digital subtraction angiography, stereoscopic displays can help doctors to analyze the complex vessel structures [125]. Calvano et al. [29] report on positive effects of the stereoscopic view provided by a stereo endoscope for in-utero surgery. The enhanced spatial perception may be also useful in other fields.

1.2.4 Improved hand-eye coordination

A differing position and orientation between image acquisition and visualization may interfere with the hand-eye coordination of the operating person. This is a typical situation in minimally invasive surgery (see fig. 1.12(a)). Hanna et al. [66] showed that the position of an endoscope display has a significant impact on the performance of a surgeon during a

Introduction

knotting task. Their experiments suggest the best positions of the display to be in front of the operator at the level of his or her hands.

Using *in-situ* visualization, there is no offset between working space and visualization. No mental transformation is necessary to convert the viewed objects to the hand coordinates.

1.3 The NARVIS Project

We must beware of needless innovation, especially when guided by logic.
– Winston Churchill (1874 - 1965)

Major parts of this thesis have been realized as part of the project NARVIS. The BFS¹-supported project aims at in-situ visualization of preoperative CT data during a minimally invasive spine surgery. The project is carried out by a consortium consisting of A.R.T GmbH, Siemens Corporate Research, Chirurgische Klinik Innenstadt der Ludwigs-Maximilian-Universität.

NARVIS aims at an application that is challenging enough to justify this complex technology. The integration of the technology into the complex surgical workflow and surgical environment is the main interest of NARVIS. In the concrete working environment, the project partners are able to create a realistic scenario for an AR system. The consortium can obtain deep insights into the technology by responding to the limitations of the surgery room.

The targeted intervention is spinal surgery. It is a good match due to medical considerations as well as for technical reasons.. For the medical team endoscopic spinal surgery is a challenging region because it is difficult to reach. Nerves and major blood vessels are close to treated vertebrae and discs. From a technical point of view, the region is optimal because it is relatively rigid; furthermore, imaging data is available whenever the surgery is indicated.

Project NARVIS	
Acronym	Navigated Augmented Reality Visualization System
Partners	A.R.T GmbH, Weilheim, Germany
	Siemens Corporate Research (SCR), Princeton, NJ, USA Imaging department
	Technische Universität München (TUM) Chair for Computer Aided Medical Procedures
	Ludwig-Maximilian Universität (LMU) Chirurgische Klinik und Poliklinik Innenstadt
Duration	3 years
Volume	1.1 Mio €

In-situ visualization can be realized with augmented optics, AR windows, and head-mounted displays (see section 1.1.1). The targeted intervention is usually not performed with optical aids. Since AR windows can be obstructive in combination with the long rigid instruments of laparoscopic surgery the project's choice is clearly the HMD. For medical applications video see-through augmentation has a number of advantages over optical see-through augmentation: It allows for synchronization between real and virtual views, hence minimizing what is potentially the largest source of error [73]. As the merging of the images happens in the computer it is possible to create an objective and automatic way of estimating augmentation errors. Furthermore, electronic merging offers more possibilities than optical merging. Lastly, the real view need not have the highest possible quality, because the real view is not used for diagnostics in the classic way employed in laparoscopic surgery. The high quality of the real view using optical augmentation is therefore not a valid criterion.

For the NARVIS project a state-of-the-art HMD-based system had to be created and integrated into the environment and workflow of a specific spine surgery.

¹Bayerische Forschungstiftung

1.3.1 Targeted intervention: Minimal invasive spine fixation

The targeted surgery is indicated in case of degradation of a vertebra or a vertebral disc. Treatment of a vertebra collapse (kyphosis) or the removal of a vertebral disc are followed by a mechanical stabilization of the spine using metal plates. The region is fixed by attachment to two neighboring healthy vertebrae. The fixations support the spine from the dorsal and the ventral side of the spine in order to have optimal mechanical properties. Accidents are the main cause for this kind of injury. The dorsal fixation is implanted in an unscheduled operation as an emergency treatment before the described scheduled surgery.

Surgical analysis	
Patients	Injury leads to ailment; Osteoporosis patients more likely to suffer from this injury
Preoperative data	CT, occasionally additional MRI
Intraoperative data	Laparoscopic live images, ISO ^{3D} cone beam reconstruction with a fluoroscope, tracking data for navigation
Duration	2-3 hours
Surgery staff	Operating surgeon, assistant surgeon, anesthetist, scrub nurse, 2 assistant nurses
Instruments	Laparoscope, drill, screwdriver, high frequency generator, scissors, knife
Critical stages	Optimal port placement, avoidance of blood vessels during the access, pedicle screw placement according to plan

Beisse et al. [11] describe the most up-to-date way of vertebral disc removal, including fixation of the link. During the surgery, a *fixateur interne* is fixed of the ventral side of the spine once the swelling has decreased. The operation starts with the definition of the entry points. The port placement is performed before the surgeon makes the first incision after the patient was cleaned. At this point, the entry points for the laparoscopic instruments and the laparoscope are defined on the skin, partly under

fluoroscopic control. The ventral side serves as access to the spine.

For ventral access, one lung lobe is inflated through a puncture. This provides space for the instruments on the path to the target region. The subsequent path to the target region is cut tissue by tissue. Once the target region has been reached, the following steps are decided upon in accordance with the indication for the surgery:

In case of a kyphosis, the injured vertebral body is straightened from a wedge to a cylindrical shape again and filled with bone cement.

In case of a damaged disc, it is removed and the surgeon estimates the dimensions of the hole with their eye. Then, the operating team focuses temporarily on another operating site. A part of the iliac crest² of the patient is cut out as bone graft for the empty intervertebral disk space. The replacements may also be heterologous³ bone material or alloplastic material, such as titanium or carbon. However, iliac crest bone grafts appears to be the best practice.

1.3.2 Analysis of the surgery

The workflow of the surgery has been intensively studied through

²rim of the upper hip

³from a different patient



Figure 1.13: Photo story board of the surgery: Port definition under fluoroscopic control – incisions – endoscopic access to target region – navigated implant screw placement – successful fixation in endoscopic image



Figure 1.14: Initial setup with body phantom

- Presence in the operating room for a realistic estimation of the environment (see figure 1.13)
- Self-created videos of such a surgery in order to enable repetition and time assessment
- Teaching videos by the implant manufacturer
- Textbook [124], [75] and literature research on the specific topic of navigated spine surgery [11], [164], [137], [59], [54], [90], [91] for understanding the current approach
- Continuous discussion with a surgical team, including the operating surgeon, in order to define the bottlenecks of the procedure

In the analysis of the surgical operation, three critical stages have been identified where three-dimensional images can provide additional necessary information for a more efficient workflow. The first stage is in the beginning of the surgery when the ports are defined. The second stages is when the laparoscopic access to the target area prepared and the third stage is when the surgeon places the implant screws.

1.3.2.1 Port placement

The definition of port positions under fluoroscopic control is a tedious and time-consuming task entailing additional X-ray exposure for the patient and the surgical team [59] (see also figure 1.15). This critical task influences the whole surgery since access to the target



Figure 1.15: Port placement: The position of the trocars is determined in an iterative procedure of fluoroscopic imaging and movements of radioopaque instruments

region is limited by the four trocars. All of the instruments and the laparoscope can only move on a straight line from the trocars to the target region since they have a rigid shaft. With in-situ visualization of the preoperative data the surgeon may take a look through a potential port to the target region in order to observe if critical tissue is on the path. A good spot for a port can be directly marked on the patient skin with a pen since the visualization can be seen in same coordinates as the real skin.

1.3.2.2 Laparoscopic access to target area

Laparoscopic access to the target area has to be established carefully in order to avoid damage to the tissue. On the way to the target area, there are often medium-sized blood vessels that cannot be seen in the endoscopic view. If these vessels are cut the minimal invasive operating site becomes filled with blood. Stopping this bleeding is a time-consuming and challenging task. In a worst case scenario, the minimal invasive operation may need to be converted to a classical one if the bleeding cannot be stopped. The major problem in closing the vessel is finding the vessel, because the operating site is filled with blood. Only small quantities of blood are necessary to cover the scene,

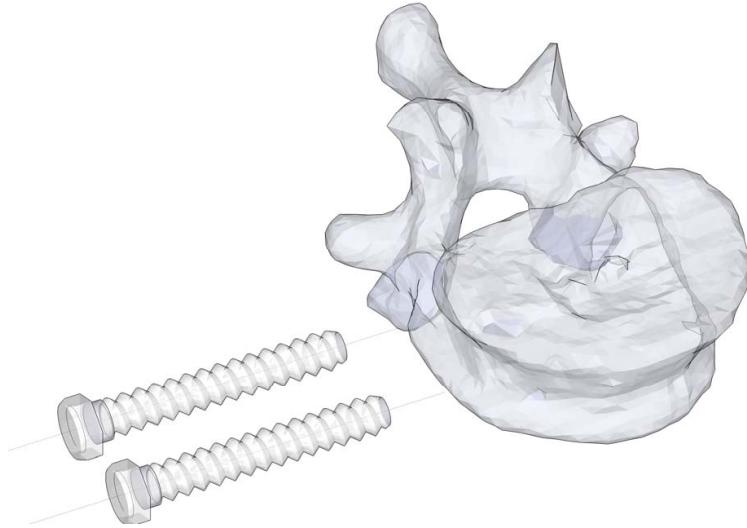


Figure 1.16: Vertebra with implant screws

and additional blood is pouring out of the hole. Avoiding cuts in these blood vessels or supporting such cuts would on average constitute a great benefit for the patient.

1.3.2.3 Implant screw placement

The implant screw placement is the most significant stage of the surgery. The permanent stability of the spine defines the success of the surgery, not regarding potential undesired side effects. In the small volume of the vertebral body, two implant screws have to be placed for optimal stiffness of the fixation while two implant screws are already placed. The screws should not touch each other, but they must not leave the vertebra body either, since it is surrounded by highly critical tissue, such as for instance the aorta and the spinal cord.

1.4 Contribution of the thesis

This thesis is intended to be a reference in medical augmented reality. The literature review (section 1.1) provides extensive research about medical AR setups. Chapter 2 provides a detailed overview on components that are necessary for a medical AR system and chapter 3 discusses how to integrate them. The last chapter describes methods for assessment of medical AR systems and validation steps that have been performed for the system developed as a part of the project NARVIS.

This thesis contains new methods for improving components of medical AR, new methods for the integration of components and new methods for the validation of a medical AR system. These include improvements to the state-of-the-art in critical issues such as latency measurement, dynamic error prediction, depth perception, AR system engineering, and temporal data registration.

1.5 Document overview

Chapter 1 introduces the history of medical AR and gives an extensive literature review on existing systems. It includes a description of the NARVIS project, as part of which most of the work of this thesis has been realized.

The second chapter contains details on the required parts of an AR system. It features a new method for temporal registration of recorded actions for synchronized display.

Integration of the components into a compelling medical AR system is described in the third chapter. It presents new concepts for engineering AR systems in order to meet of high performance requirements while preserving flexibility, transparency, and reliability. Chapter 3 ends with a technical description of the NARVIS system and three new AR systems based on similar technology.

The fourth chapter deals with the validation of a medical AR system and NARVIS in particular. The chapter includes new validation methods such as latency estimation and dynamic accuracy estimation. It contains results of several experiments that have been performed in the scope of NARVIS.

The last chapter concludes the thesis by discussing open issues and summarizing the thesis statement.

CHAPTER
TWO

COMPONENTS

An augmented reality system can be defined by its ability to display registered virtual and real objects in real-time while allowing user interaction [4, 5]. According to this definition, an AR system must include components realizing the registration, visualization of real and virtual objects, and interaction.

This chapter provides an overview of these components. New methods for temporal registration and visualization that are developed for the thesis are described in detail.

2.1 Spatial registration

In order to maintain spatial registration of the real and virtual images, the geometric and optical properties of the system are modeled and its parameters are determined. The overlay of a single point of medical data $\mathbf{p}_{MedicalData}$ onto the corresponding position $\mathbf{p}_{Display}$ of an AR display can be modeled in homogeneous coordinates with the matrix \mathbf{M}_{AR} .

$$\mathbf{p}_{display} = \mathbf{M}_{AR} \mathbf{p}_{MedicalData} \quad (2.1)$$

The projective transformation \mathbf{M}_{AR} can be decomposed into different parts modeling different aspects of a setup.

$$\mathbf{M}_{AR} = {}^{IP}\mathbf{P}_{CC} {}^{CC}\mathbf{T}_{TCA} {}^{TCA}\mathbf{T}_{TCB} {}^{TCB}\mathbf{T}_{MD} \quad (2.2)$$

The parameters of the projective transformation ${}^{IP}\mathbf{P}_{CC} := {}^{ImagePixel}\mathbf{P}_{CameraCoordinates}$ and camera position ${}^{CC}\mathbf{T}_{TCA} := {}^{CameraCoordinates}\mathbf{T}_{TrackingCoordinatesA}$ are modeled during the *calibration* process. This is valid for optical and video see-through technology. In optical see-through systems, the human eye substitutes the camera as the element in which the optical projection takes place. For video see-through systems, the calibration has to be computed only once after assembly of the system.

Changes in the relative position of the viewer and the patient are modeled with the Euclidean transformation ${}^{TCA}\mathbf{T}_{TCB} := {}^{TrackingCoordinatesA}\mathbf{T}_{TrackingCoordinatesB}$. They are determined dynamically with *tracking* hardware.

The process of finding the transformation between the tracking coordinate system and the medical data coordinates system ${}^{TCB}\mathbf{T}_{MD} = {}^{TrackingCoordinatesB}\mathbf{T}_{MedicalData}$ is chiefly referred to *patient registration*.

The overall accuracy depends on a chain of subcomputations. Errors do not simply add up, but the errors may actually be amplified by the geometry in the error propagation [72]. For each step, the algorithms have to be carefully chosen and implemented in a numerically stable fashion.

2.1.1 Calibration

The calibration process is divided into two parts. The *internal camera calibration* ${}^{ImagePixel}\mathbf{P}_{CameraCoordinates}$ recovers the matrix that projects points from camera coordinates to image coordinates. The matrix contains optical properties of a camera such as focal length and pixel geometry. The *external camera calibration* ${}^{CameraCoordinates}\mathbf{T}_{TrackingCoordinatesA}$ provides the Euclidean transformation between the optical center of the camera and a reference object as for example a tracking target. In the field of robotics the external camera calibration is often called *hand-eye calibration* or *hand-eye coordination*.

Camera calibration is commonly performed with an accurately crafted calibration pattern that includes computer vision features that can be extracted with subpixel accuracy. Examples for these are fitted lines or region features as for instance the center of mass. The image points and the fiducial positions of the calibration provide the equations for the calibration matrix. Decomposing the calibration matrix into a similarity transformation matrix and the internal camera calibration by a QR decomposition provides the correct

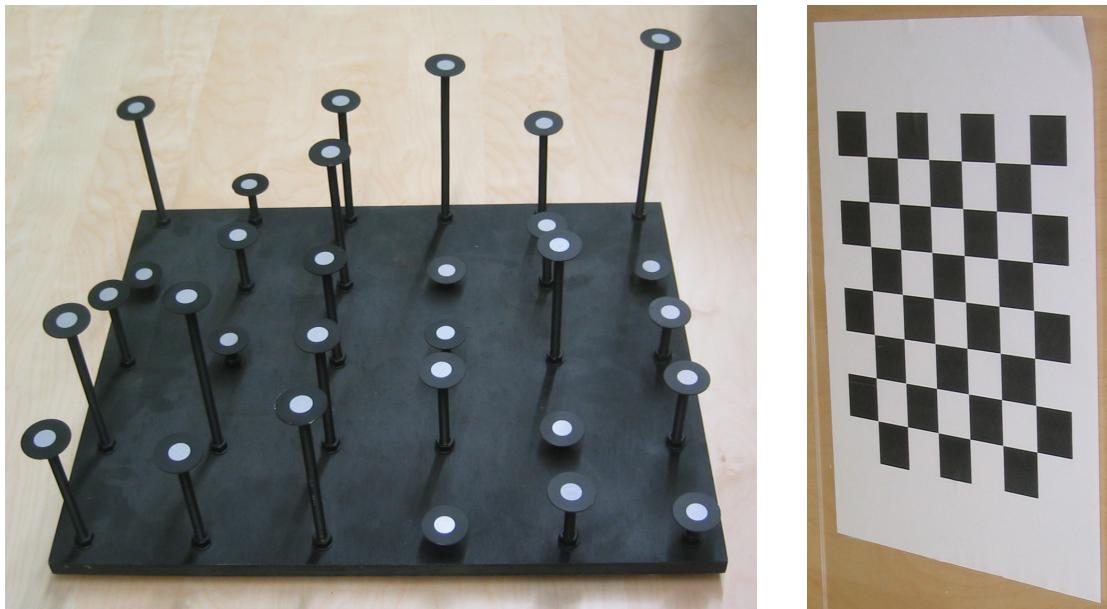


Figure 2.1: Exemplary calibration patterns: Three-dimensional pattern of circular features (left) and printed checker board pattern (right)

result [130]. However, it is mathematically more stable to model the external camera calibration using only six degrees of freedom and perform a nonlinear optimization.

The linear model of the projective transformation perfectly describes a pinhole camera. In order to take lens distortion into account, different non-linear techniques for modeling the distortion have been suggested. Determination of distortion parameters belongs to the internal camera calibration process. It can be achieved using well-established camera calibration techniques [69, 173, 197]. Tuceryan et al. [174] describe different calibration procedures that are necessary for video augmentation of tracked objects. These include image distortion determination, camera calibration and object-to-fiducial calibration, which is in the context of medical AR – the patient registration.

A popular approach for camera calibration uses printed calibration patterns, because printers allow for a cheap, fast and accurate way of pattern generation. However, printed patterns span only two dimensions. The accuracy of external camera calibration in the direction orthogonal to the paper is much lower than for other directions. Calibration with a three-dimensional pattern converges faster and therefore needs a fewer images in the calibration process.

2.1.2 Tracking

The tracking subsystem of a medical AR system provides the positions of objects. Tracking technology has already entered operating rooms for medical navigation. There is a variety of tracking systems differing in the physical way of measurement for pose estimation (see box). For a more detailed overview on tracking technology confer to the survey of Rolland et al. [135]. The prevailing method in intraoperative procedures is currently optical tracking using infrared light. The advantages for optical tracking are high accuracy and reliability. The use of infrared light can be easily explained by the fact that light conditions can be controlled for optimal measurements without disturbing human vision. Accuracy degradation is mainly caused by line-of-sight issues, which can be detected easily during the measurement.

In contrast, electromagnetic tracking measurements can be distorted by changes in the magnetic field of the tracking volume due to the presence and movement of metallic material. Such changes are hard to detect. However, electromagnetic tracking can be performed without a line of sight. Mechanical tracking offers high fidelity measurements similar to optical tracking. However, the mechanical link can be obstructive and the tracking volume is limited to the length of the mechanical linkage. Tracking accuracy degrades with the length of the mechanical link.

Tracking technology	
Type	Limitation
Optical	Line of sight, large set of markers
Magnetic	Accuracy degrades strongly with increasing speed of sensor, distance between sensor and emitter, and due to presence of ferromagnetic material
Mechanical	Obstructive mechanical link, limited range; accuracy degrades with link length
Ultrasonic	Line of sight, humidity, lower accuracy than optical
Radio frequency	Only binary information: (Not) in range
GPS	Insufficient accuracy for medical purposes

Inside-out tracking means that the sensor is considered to be the element in motion while the emitters remain static. *Outside-in tracking* uses the opposite approach. An

example for inside-out tracking is a tracking camera on an HMD that tracks a reference position in the room. In the given example, an outside-in tracking system tracks the HMD from a reference position. Although both setups produce valid results, there is a difference in accuracy even if the same camera technology with the same geometry of fiducials is used. Inside-out optical tracking offers higher rotational accuracy of the HMD position than outside-in tracking. The advantage of outside-in tracking is a larger tracking volume surrounding the tracked object. In general, outside-in tracking can afford a larger base line since the reference position is not as movable as a tracked object. A larger baseline allows for more accuracy or smaller tracking targets.

2.1.3 Patient registration

Registration algorithms are well-discussed in the community. Their integration into the surgical workflow requires a trade-off between complexity, accuracy and invasiveness. Maintz and Viergever [104] provide a general review of medical image registration and its subclassification. There are basically three options for intraoperative image registration.

Point-based registration: Registration of patient data to the AR system can be performed with fiducials that are fixed on the skin or implanted [108]. These fiducials are touched with a tracked pointer for the registration process. Alternatively, fiducials can be segmented in images provided by a tracked endoscope. While Stefansic et al. propose the direct linear transform (DLT) for mapping spatial locations of fiducials to their corresponding two-dimensional endoscope images [160], Feuerstein et al. suggest a triangulation of automatically segmented fiducials from several views [46]. Another practical possibility is using fiducials that are directly recognized by a tracking system.

Especially in maxillofacial surgery it is possible to integrate fiducials in a fixed geometry that can be attached to the body in a reproducible way[84]. For spine surgery, Thoranaghatte et al. suggest to attach optical fiducial to the vertebra and use the endoscope to track it in-situ [169]. Point-based registration is known to be a reliable solution if the set of fiducials is carefully chosen. The accuracy of fiducial-based registration depends on the number of fiducials, the quality of measurement, and the spatial fiducial arrangement. [51].

Registration points can be touched with the tip of a pointer in order to provide the coordinates in the tracking coordinate system. The position of the tip can be found out interactively by rotating the instrument around the tip [174],[56]. Registration between tracking point sets in tracking and imaging coordinates can be computed in a numerically stable fashion using the method of Umeyama et al. [175].

Image-based registration: Another approach is to track an imaging device and register the data to it. This procedure has the advantage that high accuracy is preserved without the need to attached fiducials to the patient while preserving high accuracy. Grzeszczuk et al. and Murphy [64, 117] use a fluoroscope to acquire intraoperative X-ray images and register them to digitally reconstructed radiographs (DRR) created from preoperative CT. This 2D to 3D image registration yields high accuracy parallel to the image plane, but orthogonal to it, is low. A protocol using two images from, ideally orthogonal, viewing directions is necessary to gain maximum accuracy.

Feuerstein et al. [43] directly augment cone beam reconstruction images taken with a tracked intraoperative flat panel C-arm into a laparoscope. This approach is sometimes also called registration-free [63], because doctors need not perform a registration procedure. As a drawback, such intrinsic registration is only valid as long as the patient does not move.

Surface-based registration: Grimson et al. [62] follow a completely different approach by matching surface data of a laser range scanner to CT data of the head. For sinus surgery, Burschka et al. propose reconstructing surface structures using non-tracked monocular endoscopic images and register them to a preoperative CT data set [27]. For spine surgery, Wengert et al. describe a system that uses a tracked endoscope to achieve the photogrammetric reconstruction of the surgical scene and its registration to preoperative data [191].

2.2 Temporal registration

The main idea of AR is to present data efficiently by means of an interactive presentation of registered data. The registration is not limited to spatial registration even though most of the work in the AR community focuses on it. Temporal registration can be used for visualization of similar actions that were performed at different speed. It can also be used for the temporally appropriate behavior of the system (workflow awareness). New methods for each of these cases are proposed in this section.

Generally, temporal registration can be subdivided into two types. For *offline* computation, all actions are recorded and temporal registration is computed afterwards. It can be used for temporally aligned visualization and comparison of finished actions. In *online* temporal registration, the recording of one of the data streams is not finished and that data stream is continuously updated. Matching of current states with states of the known sequence allows for context-aware behavior of a system by assigning actions to a position in the known sequence.

2.2.1 Movement comparison

Temporal registration can be used in an AR system for educational purposes. The following work has already been published in [19], [17], [13] and [4]. The work has been realized in collaboration with Tobias Blum.

For optimal learning of complex spatial movements and tasks, it is desirable to have an expert who demonstrates actions and gives immediate feedback to practicing students. In general, the schedule of experienced experts is tight. This makes it difficult to demonstrate the task to individual students and provide them with feedback. In addition, it would be desirable to allow the majority to learn from the best international experts in their field.

An AR system is proposed that learns from the expert by tracking their movements while they use a simulator or performs a real-life (often complicated) task. Using an enhanced simulator, this information can be reproduced for demonstration to students in an enhanced simulator. Direct feedback is provided by means of comparison of the expert's and students' performance.

Over the last decade, Virtual Reality (VR) simulators have been established for a number of complex educational tasks. Flight simulators and ship simulators are in common use nowadays. Using Virtual Reality for medical education is a hot topic in research. 2004's conference "Medicine Meets Virtual Reality" [193] acknowledged medical education by including it into the conference subtitle. All of the presented papers about medical education aimed at three-dimensional patient visualization or patient simulation. A number of promising prototypical simulators for medical procedures are currently under development [10, 126, 123]. Simulators offer hands-on experience without endangering a real patient. They are up to now either purely "real" like the body phantom most people know from first aid courses, or they are purely virtual [126]. Phantoms usually lack realism in terms of complexity of the model and look and feel of the material. Virtual simulators need to imitate the actual haptics if they are used for learning a critical task. Force feedback devices are still a subject of research and development [122]. Obst et al. [123] develop a simulator for delivery training for gynecologists that includes both, a phantom and a virtual reality environment interacting with each other.

VR has potentials in teaching abstract knowledge [139]. Compared with Virtual Reality systems, approaches using Augmented Reality (AR) for education rarely fully exploit the additional possibilities offered. For instance, Kaufmann et al. visualize geometric figures in an AR system for teaching mathematics [79]. The system does not necessarily makes use of the advantages of Augmented Reality except for the convenient three-dimensional interface, although the actions could be performed in a completely virtual environment.

Augmented Reality systems in medicine emphasize the in-situ navigation and visualization of imaging data (cf. section 1.1). We intend to create a system for teaching complex movements in space for medical procedures. Teaching complex practical tasks is a topic of high interest in the context of modern medical technologies such as minimally invasive surgery. Such procedures challenge the surgeon's three-dimensional perception, understanding, and manipulation skills.

We suggest a new way of teaching using augmented-reality-enhanced training simulators, which learns from the expert and replicates that expert's actions for unlimited replay and omnidirectional three-dimensional viewing by students.

2.2.1.1 Vision

The ideal education provides a sufficient level of theoretical knowledge, an expert to demonstrate the best way of performing operations and a lot of practice with expert feedback. Especially for learning medical procedures the suggested method addresses each of these three components.

- A three-dimensional representation of anatomy is not possible in textbooks or on ordinary slides. Dynamic spatial structures in the body are even more difficult to convey using textbooks. Films may help, but they do usually not offer three-dimensional views and perspectives either; furthermore the viewpoint is fixed to the camera that captures the video.
- Having one or more experts for repeatedly showing their way of performing a procedure is generally not possible because experts are rare, might have too little

time for intensive teaching, and are costly to pay. However, it would best to learn from the best.

- A lot of real-life experience would be great for medical students but not necessarily for patients. Simulators support teaching in many medical and non-medical fields. Again, an expert would be best for optimal feedback in order to learn the most. The teacher can only give immediate feedback to a student if focusing on a particular student. This match of one student to one teacher would be perfect in terms of feedback and learning, but it is inefficient for a teacher who has many students and many other (clinical) tasks to fulfill.

To summarize, the best medical education would include dynamic three-dimensional visualization for anatomy, an expert (or even a number of them for different opinions) as a mentor for demonstration, and a simulator that cannot be harmed by wrong treatment.

We propose to learn from and simulate the expert as a teacher. Therefore, we track the expert's spatial actions while he or she is performing the task either on a real case or on a simulator. The recorded movements can be shown as a way of providing feedback while the student is practicing at the simulator. We propose an AR system that does not only have a model of the patient for simulation and shows anatomical structures, but additionally models and tracks movements of the expert (or at least their instruments) for repeated demonstration. Thus, the students will be able to see the performance of the expert from their own perspective. Cheok et al. [128] capture 3D real-time content for inserting dynamic content into mixed reality. For our solution, we require additional quantitative data of the movement of the expert or their instruments in order to compare it with the movements of the student to measure differences and/or provide real-time visual feedback. Dosis et

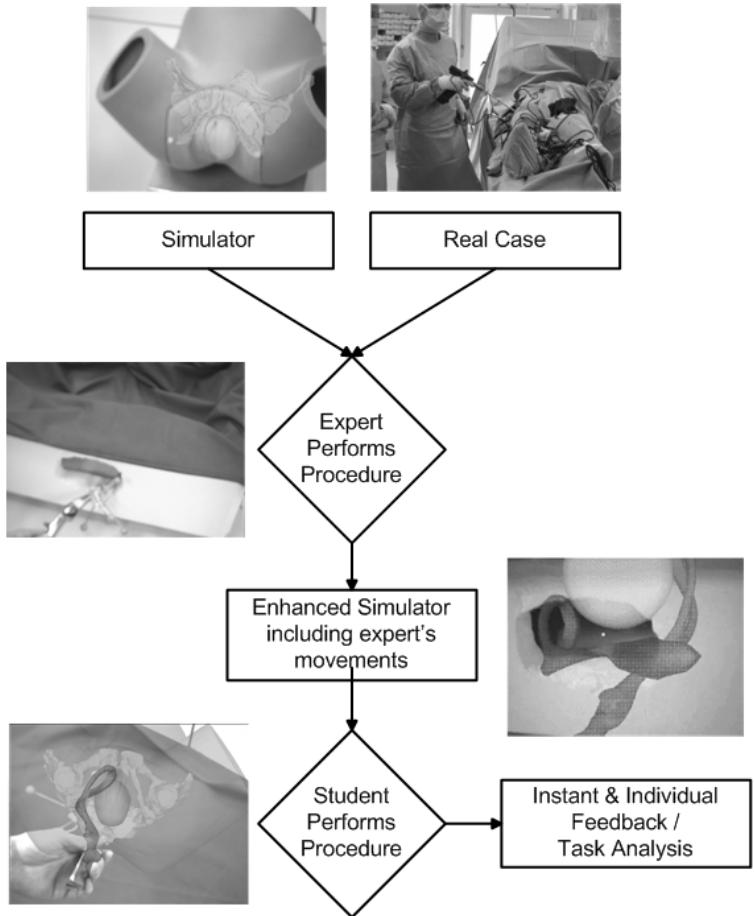
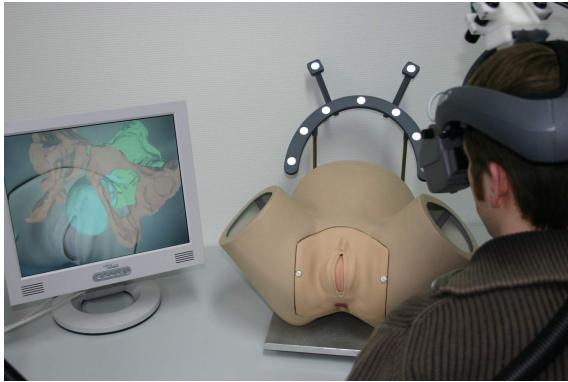
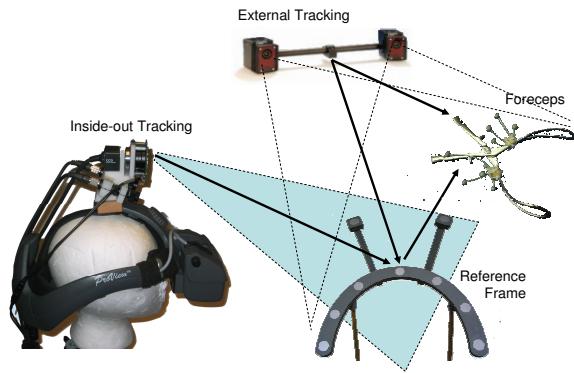


Figure 2.2: The expert's actual movements on a real case or an existing virtual or augmented reality simulator are tracked. The simulator will be enhanced and visualize these tracked actions in order to provide the students with instant feedback and perfect practical training without the permanent presence of the expert.



(a) The screen shows a duplication of the user's view (left eye)



(b) System includes inside-out and outside-in tracking

Figure 2.3: The AR Apprenticeship system setup

al. [39] record kinematic and visual data of a surgery for assessment of surgical skills. They use electromagnetic tracking with three degrees of freedom for data acquisition when tracking the instrument tip. In order to be able to display the instrument position they use video capturing. We intend to visualize the experts' movements into the real world. We cannot use video streams, because we want to allow the student to take a look from any direction. In order to assess comparable data and visualize the movements in an AR system we track the object and later visualize its 3D model.

The actions of the students are also recorded in order to correlate and compare their movements with the expert's for continuous guidance. The movements of the expert can be visualized while the students work with the simulator. This allows them to imitate the expert, thus providing a permanent feedback about whether the student's instrument movement is correct or not. Since the movements of teacher and student are tracked, it would be possible to give quantitative comparison for teaching as well as testing.

2.2.1.2 Setup

As an example for a field of use, we chose gynecologist training of human delivery. The objective is to train physicians to deal with complicated deliveries. These include the case of birth stop with forceps support. The students are supposed to learn how to insert the two parts of the forceps into the vagina without harming mother or baby. They also learn the direction and amount of force to apply when pulling. For this purpose the *Klinikum für Sportorthopädie* at *Klinikum rechts der Isar, Munich* has developed a delivery simulator for gynecologist [123]. It involves a body phantom of the mother with the baby's head inside. The head can be controlled by robot arm. Furthermore, software is used to simulate the biological functions. These functions and a virtual view of inside the phantom are shown on a computer display.

The virtual rendering of the simulator is visualized inside the phantom using the AR visualization system (see figure 2.3). The AR system is the same setup as in NARVIS (cf. section 3.4). The phantom is rigidly registered to a set of retro-reflective markers. Using

the AR system, the user has the impression of gaining a look inside the phantom. Users need not take a distracting glance away from the phantom to the displays, but can instead focus on the phantom while interacting with it.

In order to visualize the forceps in a natural way we have modeled it with the modeling software Autodesk MAYA. It generates a VRML model that can be inserted into the scene of the augmented space. Registration of the model to the real object is performed manually.

With the current setup, we can show the virtual model of the simulator in place and registered to the phantom. Furthermore, we can show the forceps even when it is occluded in the real-world view. The system can record movements of the forceps and replay it. The VRML model is rendered into the scene.

Combining an AR visualization and real world phantoms for simulators can provide a view inside the phantom without opening it physically. This retains the actual haptics of the phantom but allows for additional insights by providing views inside.

The tracked forceps may additionally visualized inside the phantom. Employing this means, we can use the phantom as a representation of a real patient to practice the procedure, but we can still show the occluded parts of the forceps inside the phantom patient.

2.2.1.3 Comparison of trajectories

In order to compare two performances of the same action we want to

- be able to visually compare two trajectories either done by professionals or by a student and a professional. For this purpose, we intend to replay two previously performed and recorded motions synchronously, using AR to have an omnidirectional view of both which helps to identify and study subtle differences.
- get a similarity measure between two previously recorded trajectories to quantitatively measure and automatically judge the performance of a student who tried to reproduce the movement of a professional

Figure 2.5(a) shows the movement of a tracked instrument when trying to perform the same motion



Figure 2.4: The system at work: Since the birth simulator is frequently used at the hospital and not available for further development, we have created an additional simple substitute. Direct view (top), occluded view (middle), augmented view (bottom)

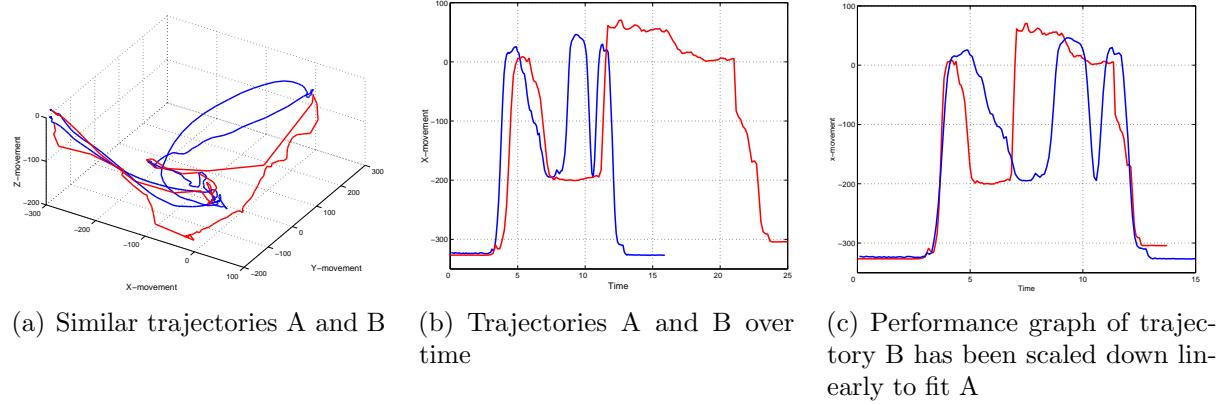


Figure 2.5: Problem statement: Exemplary trajectories

twice. A straightforward approach to getting a similarity measure between both would be to use the euclidean distance at chosen points in time, which however turns out to be inadequate. This is apparent in figure 2.5(b), which shows the x-movement over time. The motions were performed at different speeds and consequently the similarity measure would identify them as very dissimilar. Even linearly scaling them to the same length, as done in figure 2.5(c) would not yield a satisfying result because the speeds at which both tasks are performed will most likely change during execution.

Another simple approach is to take every point from the first trajectory A, find the geometrically closest point in the second trajectory B and base the similarity measure on these distances. Again, this would not deliver a satisfying result as the temporal order of the trajectories would be disregarded. To understand the problem with this method one may imagine the same motion performed once forwards and once backwards. The aforementioned simple similarity measure would consider these two movements to be equal. In general, this similarity measure provides strong response as soon as there exist similar parts within the trajectories.

Furthermore, this approach does not help us in our task to show synchronized replays, for which the temporal order must be preserved. Therefore, we need a method that on one hand is able to handle speed variations without downgrading the similarity measure and on the other hand accounts for the temporal order.

2.2.1.4 Comparison of values

The above issue can be posed as a problem, where we have two trajectories $A = (a_1, \dots, a_m)$ and $B = (b_1, \dots, b_n)$, a_i being the data we obtained at a certain time, in our case containing a time stamp, location and rotation of the tracked object. We need a time-invariant similarity measure $S(A, B)$ and a monotone mapping between the points of both, $w = ((i(1), j(1)), \dots, (i(K), j(K)))$ such that one trajectory is synchronized with the other one. Functions i and j define the mapping between the elements of the two series. This mapping w can also be seen as a warping function or warping path, that is applied to the time axis of one trajectory and synchronizes this to the other one.

At this point we refrain from giving a mathematical definition of a time invariant

similarity. This is done since different approaches use different definitions and we do not want to tie ourselves down to one at this point.

2.2.1.5 How to match points

In order to quantitatively compare trajectories we have to match points from one trajectory to another. This problem is similar to registration tasks, in which we only want to register one dimension. This registration problem cannot be solved using an approach that is analogous to rigid or affine registration, as such transformations cannot handle the synchronization appropriately. Attempts to solve this kind of problem using landmark-based registration [58] suffered from the challenge of determining the landmarks, which is a time-consuming and error-prone task. Some applications only require a time-invariant similarity measure and no mapping. Li Zhai Zeng et al. [94] suggest an algorithm that is based on a similarity measure using a singular value decomposition (SVD) to achieve gesture recognition. However, it does not return a mapping between both trajectories. More promising to fulfill our requirements are non-rigid registration techniques, which however have the drawback of being very slow. Other applications that require registration in only one dimension and that can exploit constraints use methods that are similar to non-rigid registration but take less time to compute. Firstly, there is dynamic time warping (DTW), which is well-known in speech analysis [138] and has been used in statistics [183] and signature verification [116]. Secondly, there is the longest common subsequence (LCSS) method, which has been used for similarity measures between mobile object trajectories[65]. The last two are the appropriate ones for our application. The next two sections provide detailed description of these methods.

2.2.1.5.1 Longest Common Subsequence (LCSS)

A subsequence S of the set A is a sequence of the form (a_{n_r}) , $r \in \mathbb{N}$, where n_r , $r \in \mathbb{N}$ is strictly increasing. Expressed more intuitively, one can get the subsequence S by dropping some points of A . LCSS is better known for obtaining a similarity measure between two strings by computing the longest common substring. The version for two trajectories shares the same idea and defines similarity as a high number of points that are common to both trajectories and have to be in the same temporal order. When computing common subsequences for strings we simply look for characters that are elements of both strings. With three-dimensional points it is very unlikely that we find points that are included in both movements. For this reason we regard two points a_n and b_m as equivalent if their distance $d(a_n, b_m)$ is below some chosen threshold ϵ .

Let $A_i = (a_1, \dots, a_i)$ and $B_j = (b_1, \dots, b_j)$.

DEFINITION 1. *Given a distance-function $d(x,y)$ an integer δ and a real number ϵ , the $LCSS(A, B)_{\delta, \epsilon}$ is defined as:*

$$LCSS(A, B)_{\delta, \epsilon} = \begin{cases} 0, & \text{if } A \text{ or } B \text{ is empty} \\ 1 + LCSS(A_{m-1}, B_{n-1}), & \text{if } d(a_n, b_m) < \epsilon \text{ and } n - m \leq \delta \\ \text{Max}(LCSS(A_{m-1}, B), LCSS(A, B_{n-1})), & \text{otherwise} \end{cases} \quad (2.3)$$

where δ defines a matching window that limits how far in time we search for matching points. The output is the length of the longest common subsequence, i.e. the number of matchings that are possible.

The recursion will be explained in the following paragraph. The first conditional value is the termination criteria. The second conditional value can be interpreted as follows: If the last points of both trajectories, a_m and b_n , are closer than the matching threshold ϵ , we correlate these points, memorize the mapping by increasing the result by one and continue by computing the longest common subsequence of the rest of both trajectories. Otherwise we have to leave either point a_m or b_n unmatched, depending on which gives the higher result.

	T	R	A	I	N	I	N	G
R	0	1	1	1	1	1	1	1
I	0	1	1	2	2	2	2	2
N	0	1	1	2	3	3	3	3
G	0	1	1	2	3	3	3	4
I	0	1	1	2	3	4	4	4
N	0	1	1	2	3	4	5	5
G	0	1	1	2	3	4	5	6

Figure 2.6: Matrix filled up when computing LCSS of two strings

This recursive definition is top-down and starts the computation on both complete trajectories, lessening them with each step. This problem can also be solved using a bottom-up dynamic programming approach that has a computational complexity of $\mathcal{O}(\delta(n + m))$. The corresponding algorithm fills up a n by m matrix row-wise or column-wise, where in each step (i, j) the $\text{LCSS}(A_i, B_j)$ is computed, based on the results of $\text{LCSS}(A_{i-1}, B_{j-1})$, $\text{LCSS}(A_i, B_{j-1})$ and $\text{LCSS}(A_{i-1}, B_j)$. For simplification reasons, the example in figure 2.6 shows the matrix that is used to compute the LCSS of two strings. The bold numbers denote characters that are common to both strings. Each field (i, j) of the matrix contains $\text{LCSS}(A_i, B_j)$. So the $\text{LCSS}(\text{'TRAINING'}, \text{'RING'})$ would be 4 and $\text{LCSS}(\text{'TRAIN'}, \text{'RING'})$ is 3. If only a distance measure and no warping path is required only values from the last and the current step need to be kept in memory.

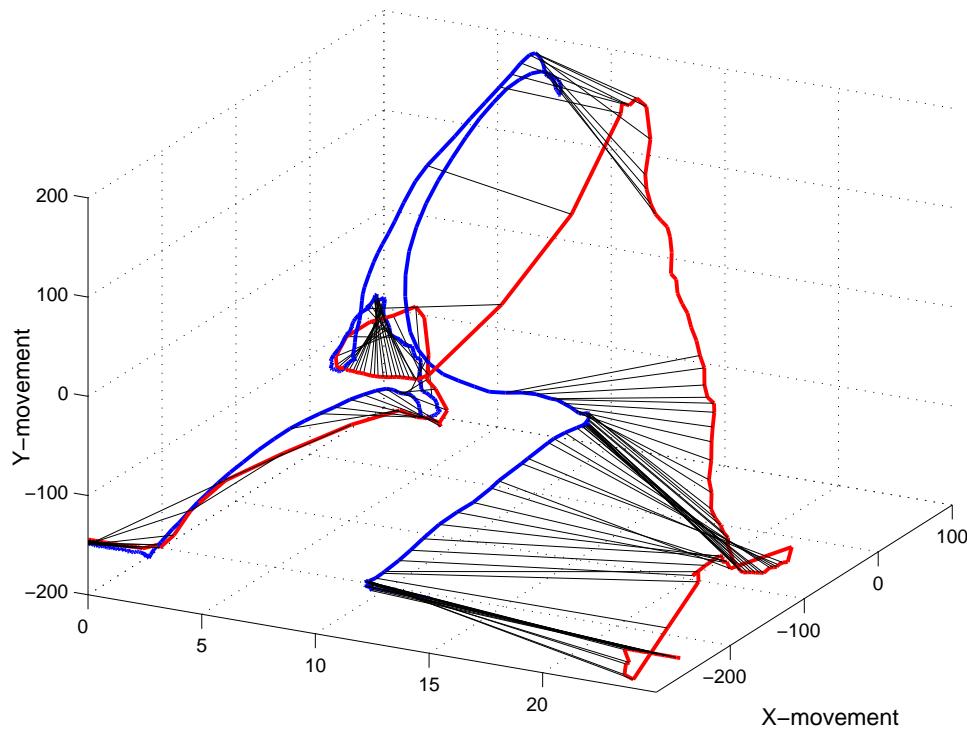
Since the value of LCSS depends on the length of both trajectories, we have to normalize the output. We define the similarity function derived from LCSS as follows:

DEFINITION 2. *The similarity function $S1_{\delta, \epsilon}$ based on the LCSS between two trajectories A and B is defined as follows:*

$$S1_{\delta, \epsilon} = \frac{\text{LCSS}(A, B)_{\delta, \epsilon}}{\min(m, n)} \quad (2.4)$$

After computing the LCSS, a mapping can be obtained that connects all points that are included in the longest common subsequence both trajectories share. To obtain this mapping the matrix LCSS created during computation must be back-tracked by starting in the lower-right field. If the upper or the left neighbor contains the same value as the current field this one is entered. Otherwise this point is stored as the upper-left neighbor is entered next. This is done until the upper-left field of the matrix is reached. In figure 2.6 this identify the common substring 'RINING', the corresponding algorithm for trajectories would generate a mapping between points.

Figure 2.7(a) shows two trajectories that have been recorded by an optical tracking



(a) Mapping between A and B

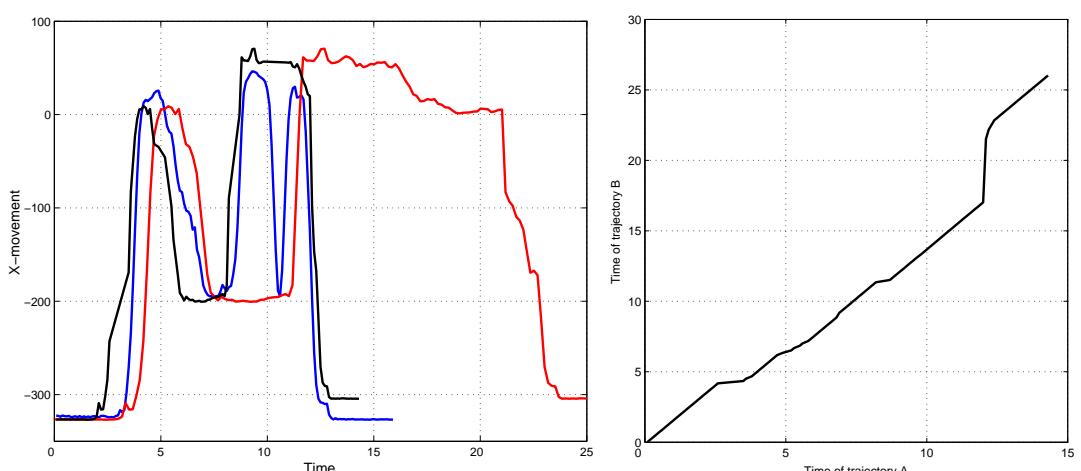


Figure 2.7: Exemplary LCSS results

system while trying to perform the same movement twice. The lines connecting both trajectories represent the mapping we got from the LCSS algorithm. Figure 2.7(b) shows the same trajectories, with one synchronized with the other. Although only movement in x-direction is shown in this figure, it has been computed also regarding the three-dimensional distance, but not the rotation of the object. The warping path we got from the point correspondences is shown in figure 2.7(c).

Since LCSS is based on points that are similar in both trajectories, we only get a mapping that includes similar points. Parts of the trajectories that are too distant are skipped. As we strive to see the differences between both movements, we cannot simply skip these parts in our replay. This problem can be addressed by interpolating between points that have been mapped.

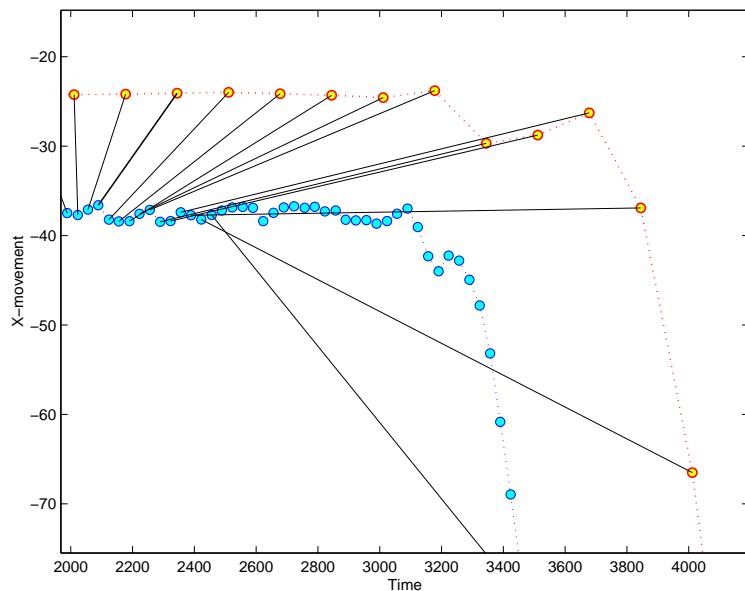


Figure 2.8: Problematic LCSS mapping when two trajectories have different update rates

where trajectory A has a higher update rate and the earlier points are assigned to every point of trajectory B that is within range ϵ . As the LCSS by definition does not regard how these points are matched, it is not possible to find a matching that is more reasonable for our case using only the similarity definition of LCSS.

2.2.1.5.2 Dynamic Time Warping (DTW)

In contrast to LCSS, DTW has to match every point with at least one point of the other trajectory. In particular the first and the last points of both trajectories must be matched to each other. The differences between both are illustrated in figure 2.10. All distances between points that are matched to each other are summed up. DTW computes the matching that has the lowest total distance with regard to a given distance function $d(x, y)$. This can be defined recursively as follows.

As long as both trajectories have been recorded with similar update rates, LCSS performs well and provides a meaningful similarity measure as well as a quite accurate and very smooth synchronized replay. Due to the normalization that is applied to the similarity measure, different update rates will not affect this measure. However, in this case, the synchronized trajectory tends to often run ahead.

This happens because it does not matter for LCSS which points are mapped to each other, as long as the maximum possible number of points is matched. An example can be seen in figure 2.8,

DEFINITION 3. Given a distance-function $d(x, y)$, the $DTW(A, B)$ is defined as:

$$DTW(A, B) = d(a_n, b_m) + \min(DTW(A_{m-1}, B_{n-1}), DTW(A_{m-1}, B), DTW(A, B_{n-1})) \quad (2.5)$$

In each recursion step, the last points of both trajectories have to be matched to each other. Either both points, or only one of them is left out in the next step, depending on which produces the lowest result. DTW can also be computed using dynamic programming in a way similar to LCSS and also requires a time of $\mathcal{O}(\delta(n+m))$ when using a matching window of δ , and another step with complexity $\mathcal{O}(n)$ afterwards to get the warping path w between A and B . Just as with LCSS, a matrix is filled up with the result of $DTW(A_i, B_j)$ in field (i, j) , which is acquired by computing $d(a_i, b_j)$ and adding the minimum of the left, upper and upper-left field. To obtain the matchings one starts at the lower-right field. In each step one stores the points a_i, b_j as correspondent with each other and proceeds either with field (a_{i-1}, b_j) , field (a_i, b_{j-1}) or field (a_{i-1}, b_{j-1}) , whichever is smaller.

Figure 2.9 shows the mappings we got, one trajectory synchronized with another one, as well as the warping function. Since all points have to be matched, outliers could have a too intense impact on how points are matched. To deal with this issue, we used a robust measure that restricts the maximum distance between two points.

Because the value DTW delivers depends on the number of correspondences, we need to define a normalized similarity function:

DEFINITION 4. The similarity function $S2$ based on the DTW between two trajectories A and B is defined as follows:

$$S2 = \frac{DTW(A, B)}{|w|} \quad (2.6)$$

where $|w|$ is the number of mappings, which is slightly higher than $\max(m, n)$.

As is the case with LCSS, issues arise if one trajectory has been recorded at a higher update rate than the other. Since every point has to be matched at least once, we would have to correlate points from the shorter trajectory with multiple points from the longer one. This would lead to a snatchy replay of the synchronized movement but can easily be dealt with. When one point has multiple correspondent points we only take one of them into account and drop the others. If both trajectories have almost the same size, points

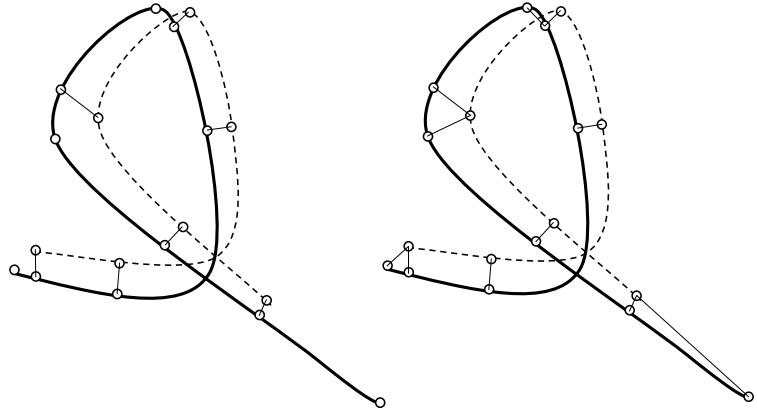


Figure 2.10: Illustration of matching of LCSS (left) and DTW (right)

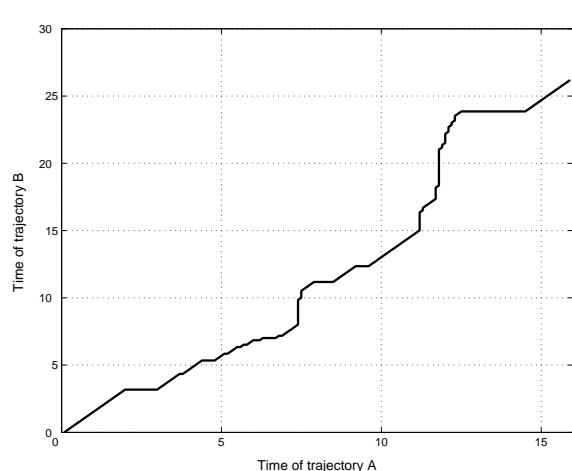
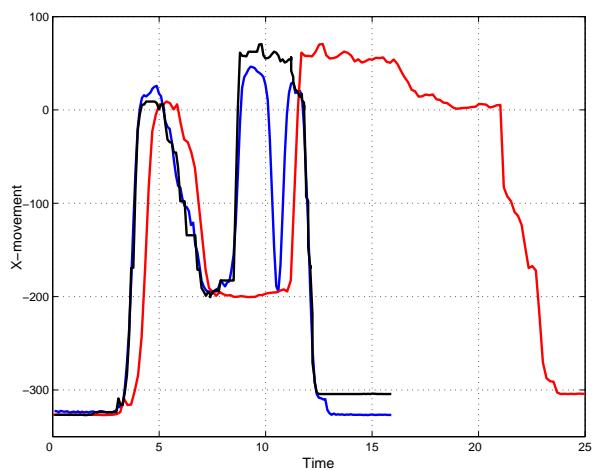
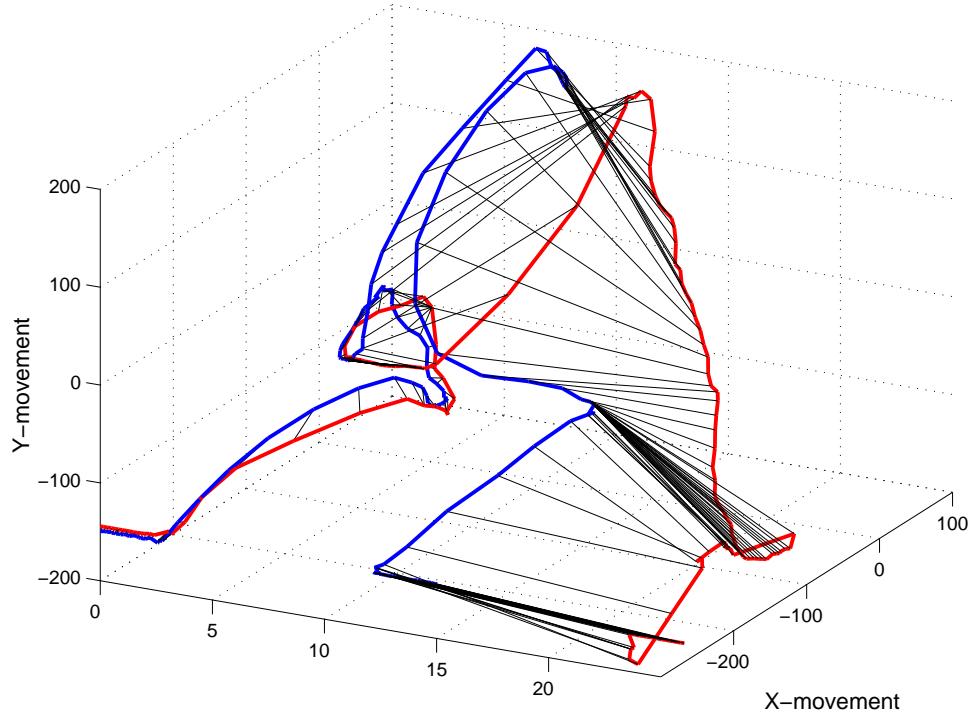


Figure 2.9: Results using the DTW

that are matched to several points rarely occur. Therefore, this will not cost us too many points. In general, we can keep nearly $\min(m, n)$ pairs of points.

2.2.1.6 Results of offline synchronization

For a better understanding of our results we recommend having a look at the video on our website¹.

We discovered that finding a time-invariant similarity measure between two three-dimensional trajectories and a temporal synchronization of both cannot be solved using simple approaches like the one shown in figure 2.5(c). However, there are problems which are similar like speech recognition or signature verification, and which can be solved using Dynamic Time Warping or Longest Common Subsequence. We implemented both methods and evaluated them for our application. LCSS is able to provide a similarity measure and a mapping that synchronizes one trajectory with the other, as can be seen in figure 2.7. Problems arise when both movements have been recorded using different update rates (see figure 2.8). DTW is also capable of providing a similarity measure and a synchronization as in figure 2.9. Problems with different update rates can be overcome with a minor change. As we implemented it in the delivery simulator we could test it in an application, where it has shown to provide an appropriate synchronization. Figure 2.11 shows two forceps that have been synchronized using DTW. When performing the same action a second time, one can see that the user follows a slightly different trajectory.

The results of offline synchronization and comparison of movements is ready to be further tested in applications.

2.2.1.7 Preliminary work on online synchronization

Real-time dynamic synchronization of a user's action with a previously recorded one is more challenging than the same process in an offline environment. We achieved a first step towards solving the problem by computing the warping based on the intermediate data of DTW, as shown in figure 2.12(b).

In our application it could be very useful to get a mapping that synchronizes a trajectory that is currently recorded with a reference trajectory. This would allow us to show a reference motion while a student tries to imitate it. By synchronizing this reference motion to the students' motion, we could adapt the speed of the reference motion to match the students' performance. Another potential application would be to automatically execute defined actions at certain points of a workflow. This means we could, for example, turn on or off augmentations or show certain information only for a period of time. This could be

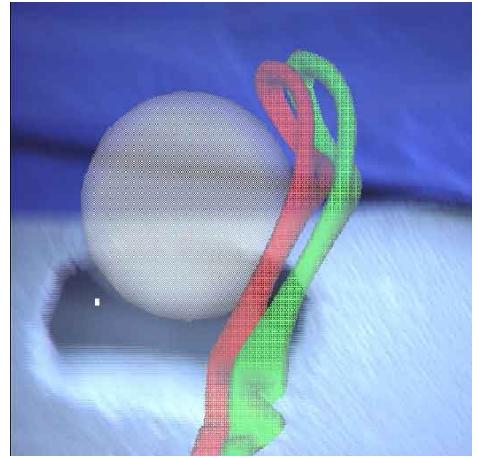


Figure 2.11: Replay of two forceps that have been synchronized

¹<http://campar.in.tum.de/pub/Sielhorst2005Synchronizing3D/Sielhorst2005Synchronizing3D.video.avi>

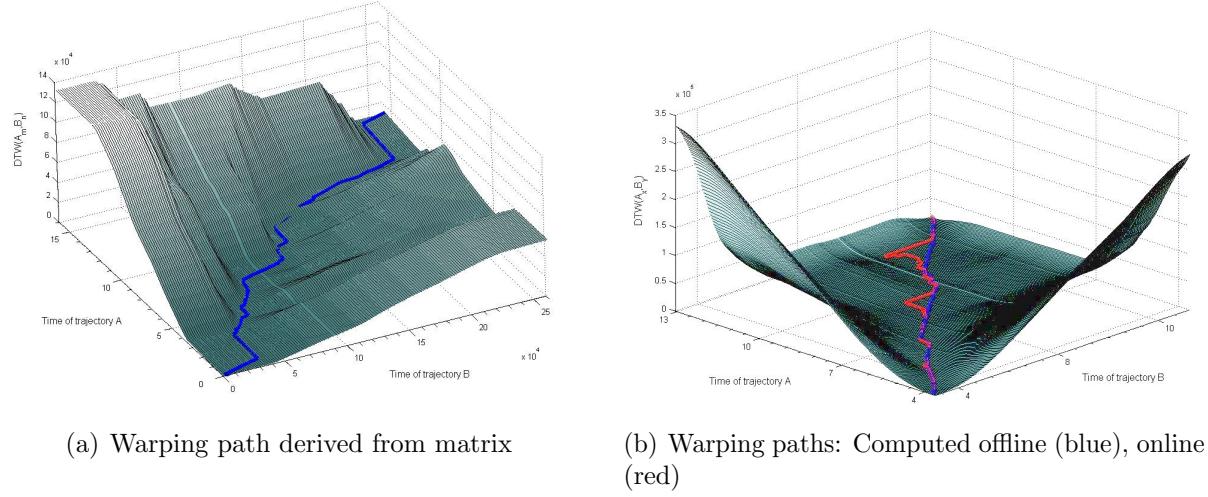


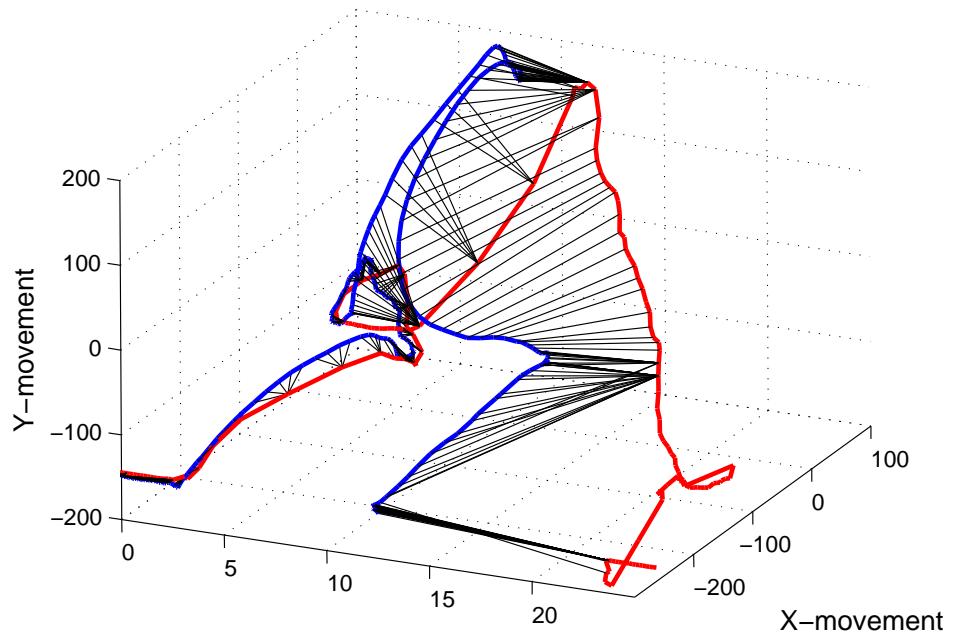
Figure 2.12: Matrix containing intermediate data from DTW visualized as a height map

done by recording a reference workflow and assigning an action to a certain point in time on the reference trajectory. By synchronizing this with the currently performed motion we could estimate when to carry out the action.

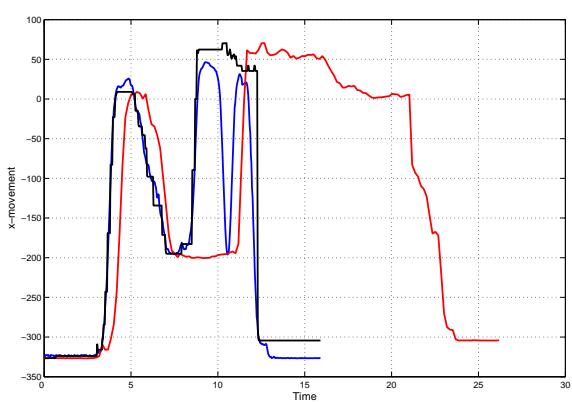
In our video we also show an exemplary illustration of this new concept. The user associates the action with a particular point of the three-dimensional reference trajectory. The action is triggered when the trainee's synchronized motion reaches the appropriate position within his/her online trajectory.

To achieve such an online synchronization we considered that we could use the intermediate data of DTW to get a preliminary result. Figure 2.12(a) shows the matrix that is generated when computing the DTW. On the z-axis, the result of $DTW(A_x, B_y)$ is drawn. The path shown in the figure is the warping path that is required to synchronize both trajectories. It is obtained by backtracking the path from the upper-right corner to the origin. When trying to get a mapping while one trajectory is not finished yet, we cannot use backtracking because we do not know at which coordinates of the matrix to start. But as we can see in figure 2.12(a), the warping path tends to take a course close to the minimum of every row. In figure 2.12(b), the warping path for another DTW matrix, obtained by backtracking, and the path obtained by selecting the minimum of $DTW(A_1, B_1) \dots DTW(A_n, B_n)$ in each step i is shown. This mapping differs from the one computed afterward, especially if we have flat areas like those viewable in figure 2.12(b). However, it delivers a synchronization that does not vary too much from the one computed offline.

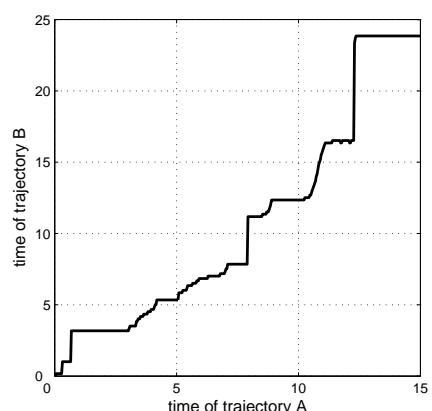
Figure 2.13 shows an online warping. In order to show comparable results in this figure, we did not synchronize a movement that was recorded during computation. Instead, we used the same two trajectories as we used for offline synchronization. Although our first results using an online synchronization are promising, it is not guaranteed to work in every case and will surely be less reliable than offline synchronization. Further testing and potentially some modifications will be necessary.



(a) Mapping between A and B



(b) B (red) synchronized with A (blue), synchronized trajectory (black)



(c) Warping path

Figure 2.13: Results using the DTW online

2.2.1.8 Discussion/Conclusion

In this section, we present systems and methods which allow us to synchronize and compare sequences of captured 3D movements. The method is applied to an AR system designed for training of physicians and midwives. The DTW gives us a similarity measure that could be used to automatically rate the performance of a trainee when trying to replicate a movement. Using only the position of a single point of the forceps is for a real application obviously not enough, but it helps understanding the problem of object motion matching with varying velocities. Since the algorithm uses an arbitrary distance measure we can easily involve other aspects of the action such as orientation of the object, velocity, or even biomechanical data (e.g. measured force) from the delivery simulator. Defining a meaningful distance measure is a non-trivial task. For instance, involving orientation the question arises how to represent the physical state of the object and how to define the distance such that it makes sense within our application. In the case of forceps, we believe that the use of multiple points, minimum of three non collinear ones, makes more sense than using position and orientation. In addition, motion of various parts of the tool have different effects on the overall result of the action. Therefore, we aim at defining the distance as *weighted* sum of distances of corresponding points. For example, parts of the tool which moves inside the patient will be weighted higher than the outside parts. This is one of the subjects of our current research and development. The synchronization process is essential because it not only provides an initial estimate for the performance of the users but also enables us to measure user's performance based on other, often more important, parameters. Force is one of the crucial parameters to take into account, speed, acceleration and angular velocity could also be considered when providing a measure of similarity/performance. Advanced visualization and HCI design could allow us to provide intuitive user interfaces to visualize these parameters and provide detailed measure of success. This is another subject of our current research and development.

In fact, a system with a well designed similarity measure can currently not replace the supervisory comments of a professional. The whole action is very complex. It has parts of different importance. An experienced supervisor knows the crucial parts and can include his knowledge into his judgment of trainee's performance. This additional high level knowledge is not yet included in our system. A possible approach for finding the importance of parts is to acquire many sequences done by different experts and use the matching algorithm proposed in this paper to find statistically significant common parts of the action, which in general are the most crucial parts. Alternatively, the expert could annotate a reference sequence for labeling crucial parts. The automatic matching proposed here would then allow us to transfer this knowledge to trainees.

Nevertheless the warping path that can be obtained when computing the similarity measure is very valuable, as it gives us the possibility to visualize both movements simultaneously and have a look at the differences between two actions. Since we use augmented reality to replay the synchronized movements, we are able to examine them from different viewpoints and analyze them in a much more tangible way than video recordings would offer. Note, that the performance velocity is not lost: The warping path (fig. 2.9(c)) contains the relative velocity of both motions. It can be used for visualization as mentioned in the previous paragraphs. Also our implementation allows

to change the speed at which the two trajectories are shown, stop them or rewind the replay through which a very detailed analysis of the trajectories is possible. In addition, an online synchronization would also be useful as one could provide timely information or execute actions at predefined points of a workflow.

2.2.2 Workflow recovery

For a context-aware system, the above-described algorithm can be used to register multidimensional signals in the operating room. The work has been realized together with Ahmad Ahmadi and Ralf Stauder. It has already been presented in [1] and [18].

In this section, we describe how to use the DTW and extend it for temporal registration of surgeries in order to perform automatic workflow recovery. Our algorithms perform this task without an implicit or explicit model of the surgery. This is achieved by the synchronization of multidimensional state vectors of signals recorded in different operations of the same type. We use an enhanced version of the dynamic time warp algorithm to calculate the temporal registration. The algorithms have been tested on 17 signals of six different surgeries of the same type. The results generated on this dataset are very promising because the algorithms register the steps in the surgery accurately to the seconds, which is our sampling rate. Our software visualizes the temporal registration by displaying the videos of different surgeries of the same type with varying duration precisely synchronized with each other. The synchronized videos of one surgery are either slowed down or sped up in order to show the same steps as the ones presented in the videos of the other surgery.

2.2.2.1 Introduction and Related Work

Surgical workflow recovery is based on the hypothesis that surgeries of the same type can be compared to each other. In general, these procedures are never performed in exactly the same way, since the surgeon experience and patient anatomy vary from case to case and complications may occur. Despite these differences, surgeries of one type generally share a common sequence of events that is typical for the procedure. Herfarth denotes that a thorough and smooth integration of surgical workflow in the clinical framework is becoming increasingly necessary. He postulates that surgical units have to become specialized and dedicated to only one type of surgery, up to a specialization on individual organs [71]. Workflow recovery is one of the crucial tasks towards the long-term goal of opinionated consultancy inside the operating room (OR), which has been drafted by experts during discussions in the OR2020 workshop of 2004 [32]. Both the medical and the administrative side of a hospital can benefit from a system, which is aware of and reactive towards the current workflow.

Riedl et al. postulate in their work more transparency of events within the OR in order to improve efficiency of room and tool usage [132]. Aggarwal et al. [1] have discussed the possibilities of surgery simulation for training purposes since it has the potential to reduce the occurrence of adverse events. Their OR simulation environment records various data streams, both video and audio, for the assessment of surgical and team skills through expert surgeons in debriefing sessions. Rosen et al. [136] read data from force/torque sensors at

laparoscopic instruments and use hidden Markov models to analyze video data about the tool/tissue interaction for the segmentation of movements. Their objective is to evaluate surgical skills. Lin et al. [98] follow a similar approach and propose a system for automated workflow recovery based on motion segmentation with linear discriminant analysis, Bayes classifiers and hidden Markov models. They analyze robot-assisted movements of a Da Vinci telemanipulator system (by Intuitive Surgical Inc.) during a suturing task.

Their aims are evaluation of surgical skill levels and training of inexperienced surgeons. Strauss et al. [166] have created manual protocols of endoscope usage time, pose changes and instrument changes in order to prove the usability and necessity of an ontology for the description of surgical workflow. They postulate and conceptualize a mechatronic assistance system for endoscope guidance. We synchronize three-dimensional hand movements of a physician using Dynamic Time Warping (DTW) [13] (also see section 2.2.1).

While Rosen et al. [136, 98] have been able to segment single and isolated tasks within a whole surgical procedure, we anticipate many benefits for a system that is able to understand the overall workflow of a complete surgery. In this paper, we propose a DTW-based system that is specialized, but not confined to the classification and labeling of workflow steps in a complete minimally-invasive procedure.

Approaches using Hidden Markov Models allow for modeling the workflow as a non-linear procedure, but they require the definition of a limited number of well-defined states. We present a complementary approach that allows for extracting these states from a large number of signals acquired during surgeries of the same type. This paper describes the implementation of our approach and first results on six minimally invasive surgeries.

2.2.2.2 Description of surgical workflow

As an exemplary surgery for workflow retrieval we have chosen cholecystectomy. 95% of the gall bladder removals are performed laparoscopically with a very low conversion rate. In spite of its complexity, the frequent performance of the surgery and its stable outcome make it a practical type of surgery for our kind of analysis.

After anesthesia and further preparation of the patient, including disinfection of the abdominal region, a small incision is made at the umbilicus or navel of the patient. The abdomen of the patient is inflated with CO_2 in order to create an abdominal cavity. For our purposes, this marks the beginning of the procedure. When the inflation is sufficient, the trocars and the laparoscope for minimally invasive surgery are inserted into the abdomen. A metal rod is inserted through one of the trocars, the liver is pulled up and the gallbladder attached beneath is revealed.

After using a grasper and a dissecting device, the surgeon identifies the positions of the gallbladder's two main vessels, the common bile duct and the bile artery. These vessels are uncovered from adhesions and adjacent tissue. When sufficiently dissected, they are clamped using a clipping device and subsequently cut using a pair of laparoscopic scissors. As a next step, the gallbladder is removed from the liver. In order to separate the gallbladder from the liver, high-frequency current is applied in two modes, cutting and coagulation mode. The metal tips of most instruments serve as electrodes and as such are able to dissect the tissue electrically. During the removal of the gallbladder, it is likely that a bleeding is caused in the so-called liver bed area, which is a large portion

at the lower side of the liver where the gallbladder used to reside. After the gallbladder has been removed completely, these bleedings are stopped using high-frequency current in coagulation mode. In order to retrieve the gallbladder, a retraction sac is inserted and the gallbladder is put into it with two or three graspers. One of the trocars is pulled out and the plastic bag is retracted through the umbilical port. After removal, the trocar is inserted again and in a last and final control phase, the situs is once more checked for bleedings. The remaining vessel stumps are checked for correct and clean clipping. A drainage tube is inserted in order to drain the saline irrigation fluid after surgery. All instruments and trocars are retrieved and the four incisions are closed up with stitches. The beginning of the abdominal suturing marks the end of laparoscopic activity and thus the end of the workflow that we consider for this work.

This sequence of events is specific to a standard laparoscopic cholecystectomy. Although variations in the duration of certain phases may occur due to various complications, the general workflow is systematic and thus reproducible.

2.2.2.3 Algorithms and methods

2.2.2.3.1 Principle of workflow recovery

In minimally-invasive surgeries, the laparoscopic instruments that are used by the surgeon at a given point of time strongly correlate with the underlying workflow. We account for the ongoing actions during the procedure with a binary model for instrument usage that will be described in the following.

A series of multi-dimensional state vectors over time takes into account that several instruments can be used simultaneously. Currently, we monitor the usage of 17 different laparoscopic instruments.

$$u(t) = \begin{cases} 1 & \text{if the instrument is used at time } t, \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

Our novel way of workflow recovery is based on the following four steps:

1. **Synchronization of events.** Using the DTW algorithm, we synchronize different surgeries with variable duration in a non-linear manner. Due to the synchronization of signals, the underlying events and workflow steps are synchronized automatically as well.
2. **Creation of an average surgery.** An average surgery is created. It reveals events that are common to all procedures of the same type.
3. **Identification of workflow phases.** Common events within the average surgery identify the start and end points of workflow phases.
4. **Obtaining workflow phases for the original surgeries.** Using the time mapping of each surgery to the average surgery, workflow phases for each original surgery can be identified at the end of the procedure.

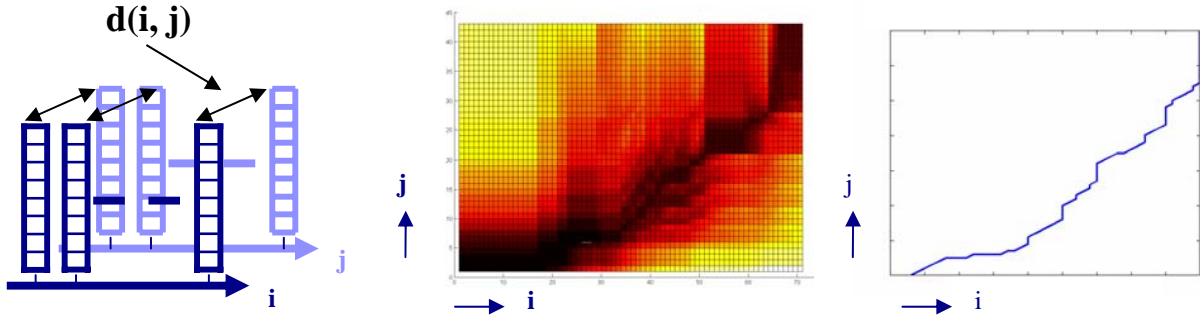


Figure 2.14: DTW warp paths synchronize the time axes i and j of two surgeries so that their state vectors are optimally aligned.

2.2.2.3.2 Classical DTW

The DTW algorithm is able to perform the non-linear synchronization that is necessary for mapping our surgery data. DTW most probably originated in the field of speech recognition [138] and was further used in the synchronization of chemical batch processes [77] as well as in many other fields such as motion detection and segmentation [95], [23], or synchronization [13].

This section will only be concerned with the basic principles of DTW and the extensions, which were necessary for this project. For details and further explanations of the algorithm, it is recommended to refer to [138], [77], and section 2.2.1.5.2. DTW is a recursive function, which aims at providing correspondences reducing the distance between two trajectories A and B . For each point in time i of curve A , the distance is calculated to each point in time j of curve B and registered in a 2D DTW matrix field.

The recursive definition of the DTW algorithm [138] and the principles of dynamic programming [12] have the effect that a trench of lowest values occurs while the DTW matrix is filled. Backtracking this trench yields the so-called warp path $h(k)$, where k denotes a newly established common timeline of the two curves. This warp path represents the optimal linkage between the two timelines i and j , in a way that the curves A and B are mapped onto each other as best possible with regard to their similarities.

By using a multi-dimensional distance function $d(i, j)$, we are able to compare the state vectors of our surgeries and obtain the desired optimal DTW time mapping (cf. Figure 2.14). For our experiments, we made use of the standard Euclidean distance.

2.2.2.3.3 Creating an Average Surgery

The creation of an average surgery is crucial for our proposed method of workflow recovery. In order to create such an average, more than two surgeries need to be synchronized onto a common timeline. Although the classic DTW algorithm only allows synchronizing two curves at a time, Wang and Gasser have overcome this problem by making use of the transitive property of the DTW [182], [183]. Using their method, we synchronize an arbitrary number of surgeries by applying the DTW several times according to the following scheme:

- Out of a set of m surgeries OP_i , ($i = 1 \dots m$), one surgery is chosen as a reference

curve OP_{Ref} and all other surgeries are mapped onto the reference using the multi-dimensional DTW. Thus, we obtain $(m - 1)$ warp paths $h_i(t)$.

$$h_i(t) = DTW(OP_i, OP_{Ref}) \quad (2.8)$$

- The warp paths are interpolated and averaged, yielding a warp path $h(t)$ between the reference curve and a common, average timing.

$$h(t) = \frac{1}{m} \sum_{i=1}^m h_i(t) \quad (2.9)$$

- In the third step, we calculate the desired warp paths $u_i(t)$ between the average timing and the timing of each original surgery. These so-called shift functions are calculated by taking the inverse of $h(t)$ as the argument of the warp paths $h_i(t)$:

$$u_i(t) \equiv h_i(h^{-1}(t)) \quad (2.10)$$

- Using these shift functions, the instrument state vectors of each surgery can be arranged onto the new, average timeline. Subsequently, the instrument vectors themselves are averaged for each time stamp, yielding the so-called structural average OP_{Avg} :

$$OP_{Avg}(t) = \frac{1}{m} \sum_{i=1}^m (u_i(t)) \quad (2.11)$$

For the reference curve, we chose the one with a duration that is closest to the average duration of the five surgeries. Due to the averaging process, state vectors in OP_{Avg} feature non-binary values as opposed to our original model. These values can be interpreted as quasi-probabilities. Let us assume that at t_a , n out of the m surgeries from the averaging set had a certain instrument in use. Then, the value for this instrument in the average will be:

$$OP_{Avg}(t_a) = \frac{n}{m} \quad (2.12)$$

These values are not a probability in the axiomatic sense, but they give an idea of how well an instrument could be synchronized by the algorithm. Furthermore, they reflect how well the surgeries correspond to each other at t_a . More importantly, these values reveal events that are typical of the surgical procedure.

2.2.2.3.4 Weighting of Signals From The Curves

It makes sense to bias the DTW synchronization towards instruments which have a high relevance for the underlying workflow. For example, the clipping device clearly indicates the work step of clamping a vessel, while the dissecting device is used as an all-round instrument in many different phases of the surgery.

Mathematically, the bias is achieved by assigning different weights to the instruments during the calculation of the distance function. The problem of weighting certain variables

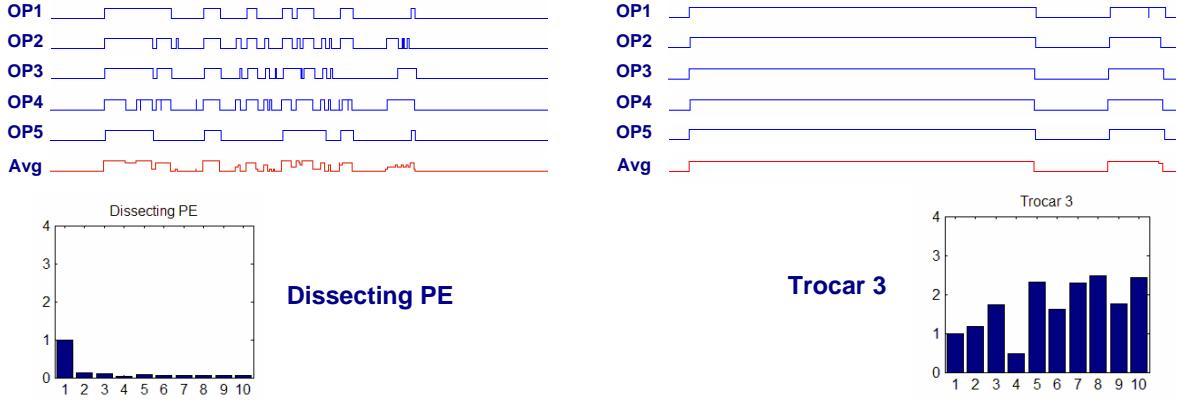


Figure 2.15: Weighting of instruments according to their workflow relevance: The better the synchronization, the higher the weight.

over others in a multi-dimensional setting has been encountered before by the group around Kassidas et al. [77].

Adopting their method, we perform iterative calculations of an average surgery. After every round, each instrument from the state vector is compared to its synchronized equivalent. A new weight is then assigned and subsequently normalized.

$$W(j, j) = \left[\sum_{i=1}^n \sum_{k=0}^{t_{avg}} [OP_i(k, j) - OP_{Avg}(k, j)]^2 \right]^{-1}, W \left\{ n_{Instr} / \left[\sum_{j=1}^{n_{Instr}} W(j, j) \right] \right\} \quad (2.13)$$

In simple words, the rule is to give higher weight to those instruments which could be synchronized very well and lower weight to instruments with only poor synchronization (cf. Figure 2.15).

2.2.2.4 Experiments and Results

2.2.2.4.1 Workflow Retrieval

For the evaluation of our algorithms, six surgeries were recorded by courtesy of our clinical partner. The average surgery was calculated with five surgeries from the set and after ten iterations of weight updating. The sampling frequency for the instrument state vectors was 1Hz. The state vectors were manually read out from the laparoscopic video. The sixth surgery was then synchronized with the average in order to evaluate the quality of workflow retrieval for surgeries. The resulting average surgery OP_{Avg} can be seen in Figure 2.16.

The curve segment from OP_{Avg} marked with a dashed ellipse demonstrates that common events are carved out while uncommon events tend to be attenuated but not neglected within the average surgery. This is due to the synchronization only and does not employ prior knowledge or models of the workflow. A short explanation shall substantiate this assertion. For every patient, two main gallbladder vessels have to be severed. This step is

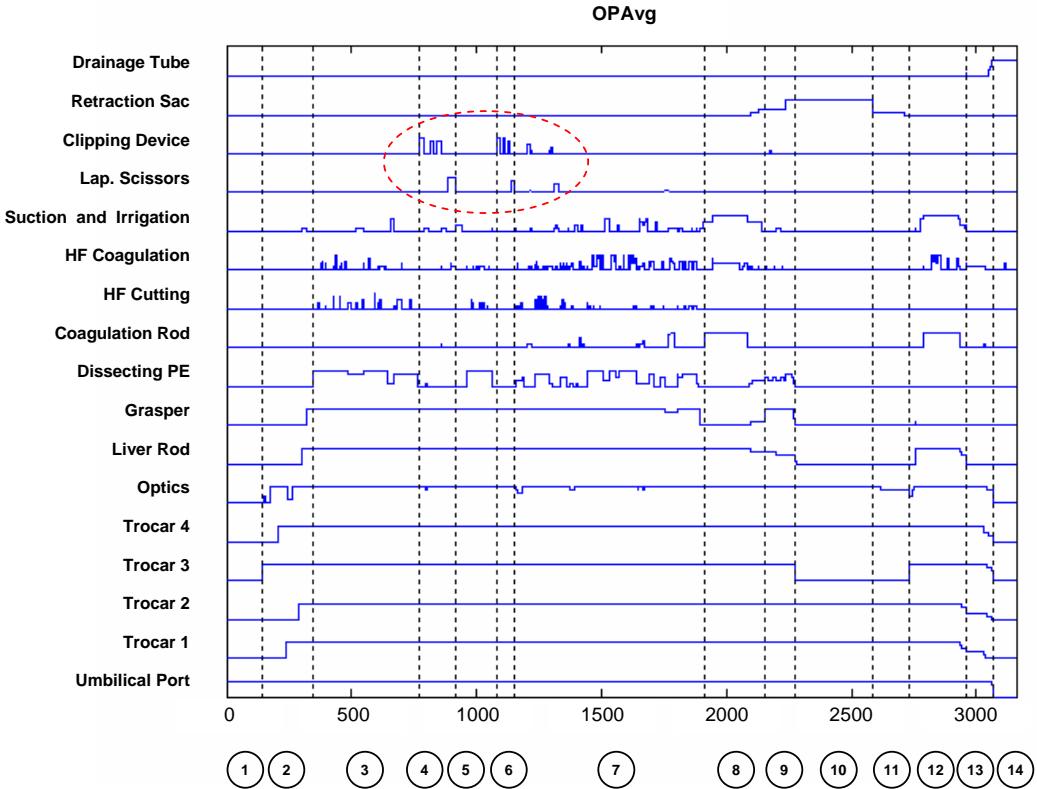


Figure 2.16: Average surgery with 14 workflow phases

represented by three high peaks for the clipping device and one high peak for the scissors. For some patients, these vessels have branched, which requires additional clipping and cutting. Since these additional events are rather uncommon, the following spikes have lower amplitudes or lower quasi-probabilities.

State changes from zero to one or one to zero within OP_{Avg} were defined as trigger events for workflow phases. Using these triggers, 14 workflow phases could be identified within the average curve. The segmentation of these phases on the multi-dimensional curves can be seen in figure 2.16. The same trigger events were identified within each original surgery. Thus, two trigger instants exist for each phase in an original surgery. The first is the original time stamp t of the phase start. The second one, t^* is obtained through the average surgery OP_{Avg} and the shift function $u_i(t)$, which connects the timeline of OP_i with the one of OP_{Avg} .

A deviation of up to five seconds between these time stamps was tolerated. Deviations higher than five seconds were classified as a wrong phase detection. In six procedures with 13 trigger events each, the proposed system was able to identify 92% of the events correctly. Moreover, 83% of the correctly classified phases were detected with a precision of one second or less. These results demonstrate that our approach is reliable and promising. The video presentation shows the user interface and some exemplary results.

2.2.2.4.2 Online temporal registration of workflow

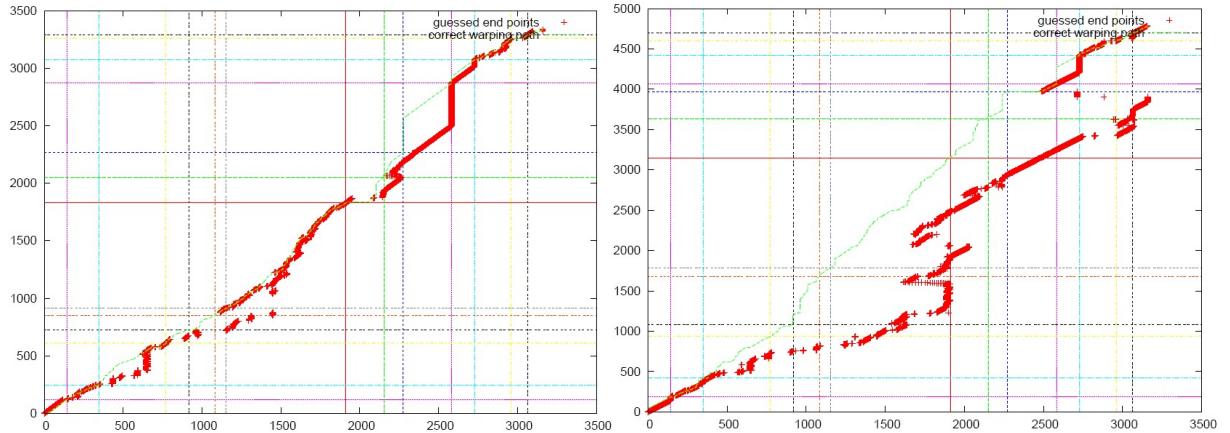


Figure 2.17: Online registration of surgical workflow. The red crosses denote the end of an online warping path. The green line denotes the offline warp path. Vertical and horizontal lines denote transitions between surgical phases. The second (left) and the fourth (right) operation have been compared to the average surgery.

Further experiments with the same data have been conducted with regard to online registration of surgical workflows. Using the same algorithms for online workflow registration, we observed strongly deviating estimations (see figure 2.17 (right)).

Currently, a subgroup of our institute continues the initial work presented here on online and offline registration of surgical workflow.

2.2.2.5 Conclusion

Workflow recovery is a key competency of context-sensitive systems and workflow-aware operating rooms. We have shown that our approach of workflow recovery is able to identify common phases in different surgeries of the same type even under the conditions of a real surgery. For this task, an average surgery is computed from a set of signals from exemplary surgeries, which already provides the desired key information for many workflow specific applications. The average curve is achieved by synchronizing multiple procedures and it reveals key events of the procedure. The change of workflow phases can be identified with a reliability of 92% with a tolerance of 5 seconds. Each of the signals including the video streams from the laparoscope and the video stream of our three external cameras can be shown simultaneously for two or more surgeries. The fine-grained synchronization of the algorithm allows for stretching the playback speed in a way that the surgeries are visualized according the workflow phases. Therefore, each single frame of all 24 videos is appropriately labeled with synchronized workflow information for further studies. The playback is useful for thorough and unprecedented analysis of surgical workflow, educational and training purposes and evaluation of surgical skills.

2.3 Visualization

The visualization component is a vital part of an augmented reality system. Due to the combination of real and virtual images, special problems occur in AR that do not appear in pure virtual visualizations. The user's depth perception can be misled in a three-dimensional display and the data representation can be ineffective due to suboptimal viewports implicitly given by the user.

2.3.1 Depth perception

The issue of wrong depth perception in AR was discussed as early as 1992 when Bajura et al. [6] described the first video see-through AR system. When merging real and virtual images, the relative position in depth may not be perceived correctly although all positions are computed correctly. When Edwards et al. [127] created their augmented microscope, they realized that "Experimentation with intra-operative graphics will be a major part of the continuation of the project". Drascic and Milgram [40] provide an overview of perceptual issues in augmented reality systems. While many problems of early systems have already been addressed, the issue of a correct depth visualization remains unsolved.

Depth cues are physical facts that the human visual system can use in order to refine the spatial model of the environment. These include visual stimuli such as shading but also muscular stimuli such as accommodation and convergence. Psychologists distinguish between a number of different depth cues.

Cutting and Vishton review and summarize psychologist's research on nine of the most relevant depth cues [33] revealing the relevance of different depth cues in comparison to each other (see figure 2.18). They identify interposition as the most important depth cue although it is only an ordinary qualifier. This means, that it can only reveal the order of objects, but not a relative or absolute distance. Stereo disparity and motion parallax are the next strongest depth cues in the personal space with a distance of up to two meters. The visual system calculates the spatial information together with the depth cues of relative size/density, accommodation, convergence and areal perspective. Especially the latter one is hardly taken into account at distances of less than two meters unless the person is in fog or under water.

It is the very nature of AR to provide a view that does not represent the present physical conditions, while the visual system expects natural behavior of its environment for correct depth perception. What happens if conflicting depth cues are present? The visual

Depth cues	Type of information	Type of physical source	Accuracy $\Delta d/d$ below 2m
Occlusion	Order	Pictural	0.1%
Binocular Disparity	Absolute	Binocular	0.2% - 0.01%
Motion Perspective	Relative	Movement	~0.6%
Relative Size	Relative (+absolute)	Pictural	2%
Height in Visual Field	Absolute	Optical (similar concept: convergence)	---
Accommodation	Absolute	Optical	3% - 20%
Convergence	Absolute	Optical	3% - 20%
Relative Density	Relative	Pictural	10%
Aerial perspective, Transparency	Relative	Pictural	---

Figure 2.18: List of depth cues ordered by importance in short range according to Cutting and Vishton [33].

system weights the estimates according to their importance and personal experience [33]. Conflicting cues could result in misperception, adaption and motion sickness.

2.3.1.1 Misperception

Conflicting depth cues indicate that at least one depth cue is wrong. Since the visual system is weighting the depth cues, the overall estimate will generally not be correct. This also applies to computer-generated images. If a contradicting depth cue is present depth perception can be wrong although the computer has generated geometrically correct images.

Especially optical augmentation provides images with different parameters for their real and virtual parts, resulting in incompatible depth cues. The effect is described as ghost-like visualization, which refers to its unreal and confusing spatial relationship with the real world. The visual system is quite sensitive to relative differences.

Current AR systems handle depth cues well that are based on geometry, such as for instance relative size, motion parallax and stereo disparity. Incorrect visualization of interposition between real and virtual objects has already been identified to be a critical issue by Bajura et al. [6]. It has been discussed in more detail by Johnson et al. [76] for augmentation in operating microscopes, and by our group (see [12] and section 4.2.4) for video see-through HMDs. Our work suggests that opaque superimposition of virtual objects that are inside a real object is not advisable. Alternatives can be a transparent overlay, wireframes, and a virtual window. Each possibility imposes a trade-off: Transparent overlay reduces perceptibility of the virtual image, the wireframe is not suitable for complex spatial geometry, and the virtual window locally occludes the real view.

2.3.1.2 Adaption

Incorrect visual depth perception can be corrected by learning if another sense can disambiguate the spatial constellation. The sense of proprioception provides exact information about the position of the human body. Biocca and Rolland [18] set up an experiment, in which the point of view of each subject was repositioned with a video see-through HMD. The adaption time for hand-eye coordination was relatively short and the adaption was generally successful. Unfortunately, another adaption process is necessary when the subject is exposed to the normal view again.

2.3.1.3 Motion sickness

In the worst case, conflicting visual cues can cause reduced concentration, headache, nausea etc. These effects have been discussed in the virtual reality and psychology community [86].

Modern theories state that the sickness is not caused by conflicting cues, but by the absence of better information to keep the body upright [131]. This means that a subject with a weak sense of balance who is standing while performing a task is more likely to encounter cybersickness than a sitting pilot with a non-emersive system, even if the latter subject is exposed to many conflicting depth cues.

In our video see-through HMD-based experiment with 20 surgeons, we [12] found no indication of the above symptoms even after an average performance time of 16 minutes. In the experiment, the subjects were asked to perform a pointing task while standing. The overall lag was reported to be $100ms$ for most of the visualizations. The remote field of view was not covered. For less emersive AR systems than this HMD-based one and for systems with similar properties, motion sickness is therefore regarded to be unlikely.

2.3.2 Three-dimensional data representation

Three-dimensional voxel data cannot be displayed directly using an opaque value for each voxel as done for two-dimensional bitmaps. There are three major ways of three-dimensional data representation. Each of the representations has its own advantages and drawbacks, depending on the application.

2.3.2.1 Slice rendering

Slice rendering is the most simple way of rendering. A slice of the whole volume is taken for visualization. Radiologists commonly examine CT or MRI data represented by three orthogonal slices intersecting a certain point. The main advantage of this visualization technique is the prevalence of this method in medicine and its simplicity. Another advantage that should not be underestimated is the fact that this type of visualization defines a plane. Since one degree of freedom is fixed, distances in this plane can be perceived easily. Traub et al. [22] showed that slice representations as used in first generation navigation systems have superior capabilities in representing the precise position of a target point. They also found that for finding a target point it can be more efficient to employ a different data representation.

Whenever two or three points of interest and their spatial relationship are supposed to be displayed, an oblique slice visualization can be useful. Without a tracked instrument it is cumbersome to position such a plane.

The major drawback of slice rendering is that this visualization does not show any data outside the plane. This is not a constraint for measuring visualizations in the plane, à la *How far can I go with a drill?* However, optimizing questions like *In which direction would a drill be furthest from critical tissue?* cannot be answered efficiently with slice rendering.

2.3.2.2 Surface rendering

Surface rendering shows transition surfaces between structures. Often these transitions are segmented and converted to polygons. The desired tissue is segmented either manually, semi-automatically or automatically, depending on the image source and the target tissue. The surface polygons of a segmented volume can be calculated using the marching cubes algorithm [101]. Graphic cards offer hardware support for this type of vertex-based data. This includes light effects based on the normals of the surfaces, which consequently consume only little additional computation time.

Since recently, ray casting techniques can be computed sufficiently fast using graphic cards equipped with a programmable graphics processing unit (GPU) [87]. They can also

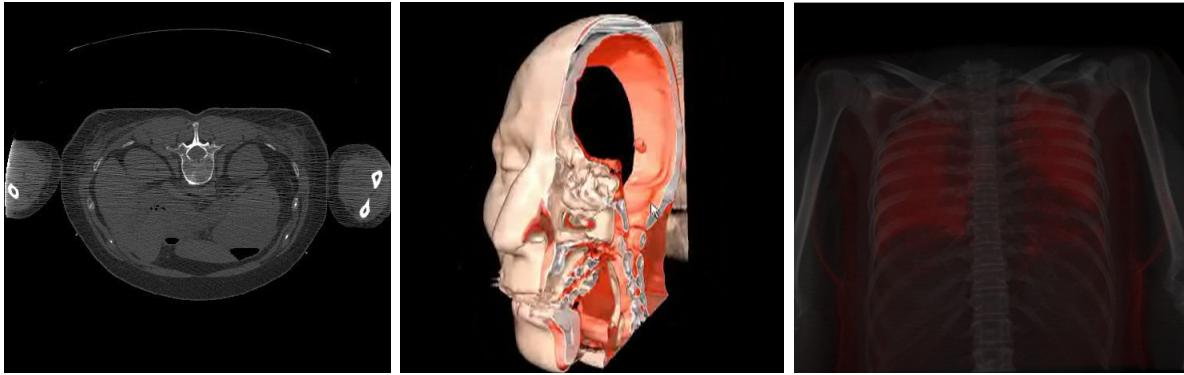


Figure 2.19: Exemplary visualization of CT data: Slice view, surface rendering (courtesy of Oliver Kutter), and volume rendering

be used for surface rendering of iso-surfaces. As the surfaces need not be transformed to polygons using raycasting, the rendered images are smoother. The sampling rate is optimal for a specific viewing direction. Additional effects, such as refraction, reflexion, and shadows, are also possible with this rendering technique.

As a beneficial side effect of surface rendering, distances and cutting points can be computed easily.

The segmentation step is a major obstacle for this kind of visualization. Segmentation of image data is still a challenging problem with brisk research going on. Available solutions offer automatic segmentation only for a limited number of structures. Manual and semiautomatic solutions can be time-consuming or at least time-consuming to learn. The benefits of a visualization using an interactive segmentation needs to justify the extra workload on the team. Also, the ray-casting technique needs clear structures in the image for acceptable results, which can be achieved with contrast agents or segmentation.

2.3.2.3 Volume Rendering

Direct volume rendering [93] creates the visualization by following rays from a certain viewpoint through three-dimensional voxel data. Depending on the source of data and the intended visualization, different functions are available for generating a pixel from the ray. The most prominent function is the weighted sum of voxels. A transfer function assigns a color and transparency to each voxel intensity. It can be further combined with the image gradient. A special kind of volume rendering is the digitally reconstructed radiograph (DRR) provides projections of a CT data set that are similar to X-ray images.

The advantage of direct volume rendering is that it has the capability of emphasizing certain tissues without an explicit segmentation. This is achieved by using a transfer function for accenting the tissue. Clear transitions between structures are not necessary. Additionally, cloudy structures and their density can be visualized.

Particularly for AR applications, the major disadvantage of this approach used to be that rendering was too slow. However, hardware-supported rendering algorithms on current graphic cards can provide sufficient frame rates on three-dimensional data that has a clinically relevant size. Currently, 3D-texture-based [176] and GPU-accelerated raycast

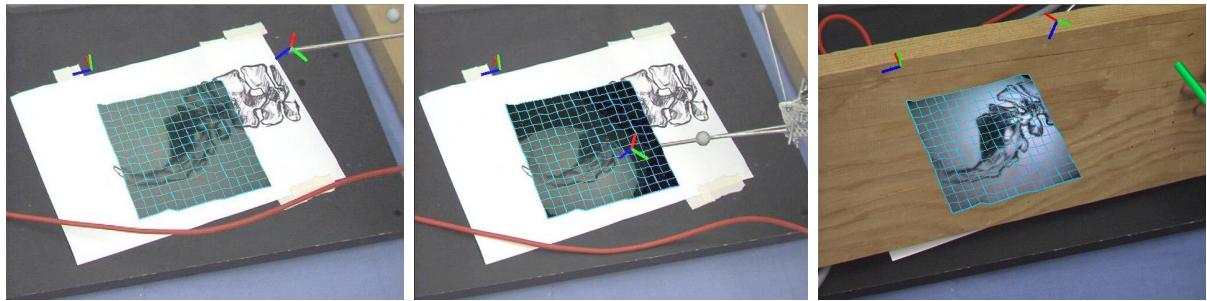


Figure 2.20: Direct view using endoscope data

[87] renderers are the state of the art in terms of speed and image quality, with the latter offering better image quality.

2.3.3 Selected medical visualization topics

The following two sections describe the preliminary work on two selected medical visualization topics. The work is a follow-up of the evaluation described in section 4.2.2. The work described in both sections is still in progress.

2.3.3.1 In-situ visualization of endoscopic images

Endoscopic images are the enabling technology of minimally invasive surgery. They provide insight into the body by providing high resolution images without radiation exposure. The chief limitation of endoscopic images is clearly their small field of view. They cannot provide an overview of the anatomy and are only able to show a particular detail instead. Gathering where this detail originates from is a non-trivial task for the operating surgeon, as it is the assistant surgeon - who may stand opposite the surgeon - that actually moves the endoscope. For providing the context via medical AR, two approaches are conceivable: The endoscope can be either augmented (see 1.1.5) or endoscope images can be integrated into an in-situ visualization device.

The integration of endoscopic images into a three-dimensional display is at first sight a straightforward task, since the images have one dimension less than the display. A virtual display can be integrated easily into the three-dimensional scene. However, if the tracked endoscope data is supposed to be overlaid registered to the real view, depth information has to be calculated. In contrast to ultrasound images, the pixels do not originate from a plane, but they are projected from a three-dimensional structure.

Fuchs et al. [55] integrated endoscopic data into in-situ visualization by means of structured light for the computation of depth. The pattern generator is an additional device that has to be inserted into an additional trocar for a reliable computation.

Dey el al. [38] texturize the surface of preoperative CT with endoscopic data and display it on an ordinary screen. They aim at providing an overview by visualizing the image position registered to the CT and mosaicing of the endoscopic images. This approach requires a surface that can be visualized with an endoscope and that has not changed since the CT generation.

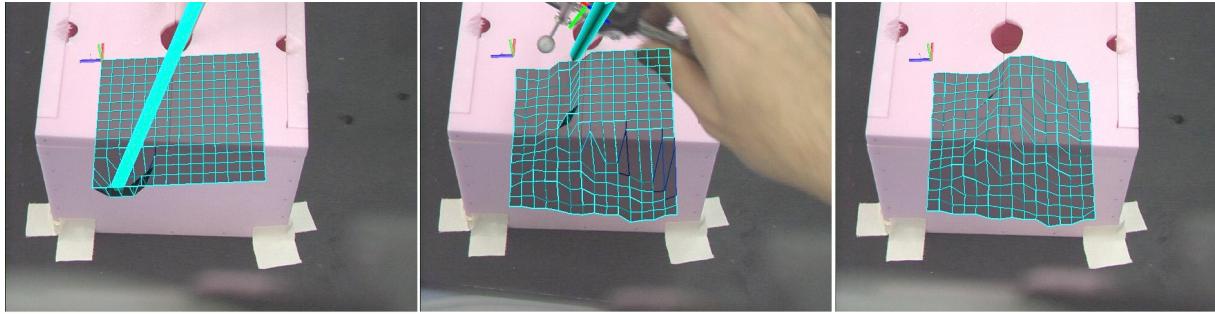


Figure 2.21: Updating the grid inside the rectangular phantom

In the following, we describe a new method for depth reconstruction, show a simple but effective extension of endoscopic visualization on a screen, and suggest an effective way of interaction for obtaining the fallback visualization. The implementation has been realized for the Diploma thesis of Alfredo Higueras Esteban.

2.3.3.1.1 Depth reconstruction

The challenging part of in-situ visualization of endoscopic images is the generation of depth information. In our approach, the surface is generated interactively via the tracked surgery instruments. We assume that in minimally invasive surgery, the instruments implicitly define the surface by touching the tissue, since the cavity has very limited space.

Point-based generation of surfaces could be done via Delaunay triangulation. However, the system has to know when the instruments touch the surface. For that reason, we have chosen a grid that is updated only in the direction of the operation target. If an instrument intersects with the surface of the grid the corresponding grid points are moved perpendicularly to the initial grid (see figure 2.21).

Texturizing a surface with an image can be done with the support of the graphics hardware. The position of each point of the surface is projected into camera coordinates using the camera calibration matrix. The resulting points have to be transformed from pixels to texture coordinates ($\in [0; 1]$). These texture coordinates can be directly assigned to the texture.

2.3.3.1.2 Automatic horizon recovery

The assistant surgeon who steers the endoscope has to recover the horizon manually by updating the rotation between the camera and the endoscope optics.

The tracking data information of a calibrated endoscope can be used for automatic horizon recovery by rotating the image electronically. This frees one of his hands and relieves the mental load. The proposed method allows for seamless integration with the classic way horizon recovery. The camera may still be rotated by classic manual interaction.

The horizon can be defined by a vector \mathbf{v}_h that is perpendicular to the horizon of any view. This vector can be obtained by means of a tracking system that is fixed to a room. It can also be defined relative to the patient in order to see a certain landmark in the endoscope aligned to one of the axes of the endoscope image.

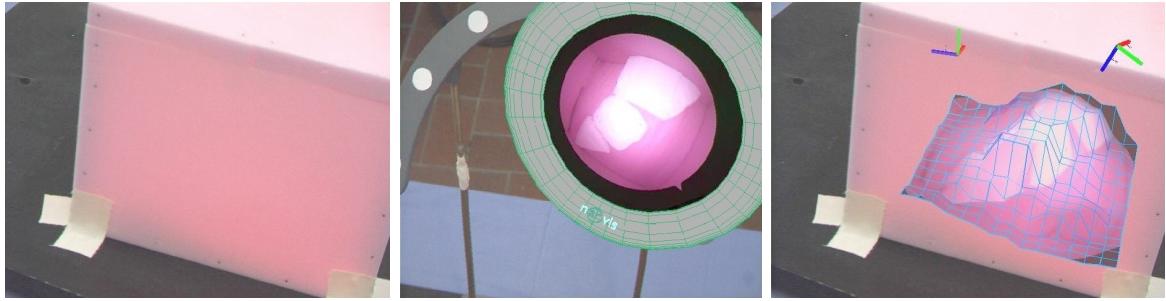


Figure 2.22: Rectangular phantom with white stones inside

The rotation angle θ of the image can be computed by projecting \mathbf{v}_h into the image plane and subsequently computing the angle between \mathbf{v}_h and the y -axis.

$$\frac{\mathbf{M}\mathbf{v}_h}{\|\mathbf{M}\mathbf{v}_h\|} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -\cos(\theta) \quad (2.14)$$

where \mathbf{M} denotes the matrix that projects world coordinates into image coordinates.

The automatic horizon recovery can be shown on an ordinary display.

2.3.3.1.3 Fallback visualization

In order to provide a fallback visualization, we have added a virtual display on which the endoscope image is shown. It can be placed close to the operation site because it is only virtual. With this metaphor, we provide the original image to be displayed in the HMD. At first glance, this does not appear noteworthy. However, we would like to emphasize that this way, we are able to see and monitor the interaction with the AR system. We regard watching a virtual screen to be faster and easier than choosing a different visualization option from a menu.

The virtual display has a circular design since the optics of the endoscope provide a circular clipping of the image. The image part that is black is not visible in the scene due to the circular shape.

The virtual display has a static position in the scene in order to be found easily. However, it is automatically rotated to be perpendicular to the viewer in order to present the original image data in an optimal way without perspective distortion.

2.3.3.1.4 Results

Our initial results show that this kind of visualization is feasible in a phantom experiment. The surgical team showed excitement about the possibility to easily and quickly return to a certain endoscope view using the technology. This is a practical issue during surgery. When an instrument is changed, the old one often needs endoscopic guidance back to the trocar, while the new one needs guidance to the operation site.

Furthermore, the possibility to create the same visualization from different points of view using a different trocar was well-received by the surgeons.

The back-projection error level looks promising at the image center of the endoscope, but is much higher in the regions closer to the image border. A back-projection error of

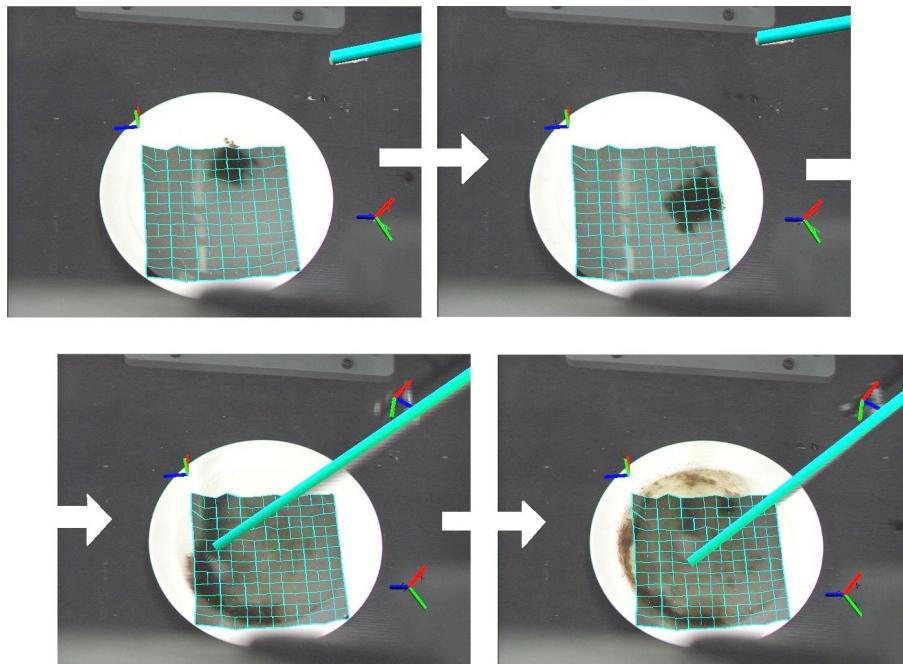


Figure 2.24: Demonstration of speed: dynamic images of stirred powder in water

up to 2 cm (see figure 2.23) at the image border is not untypical in this setup with an arthroscope.

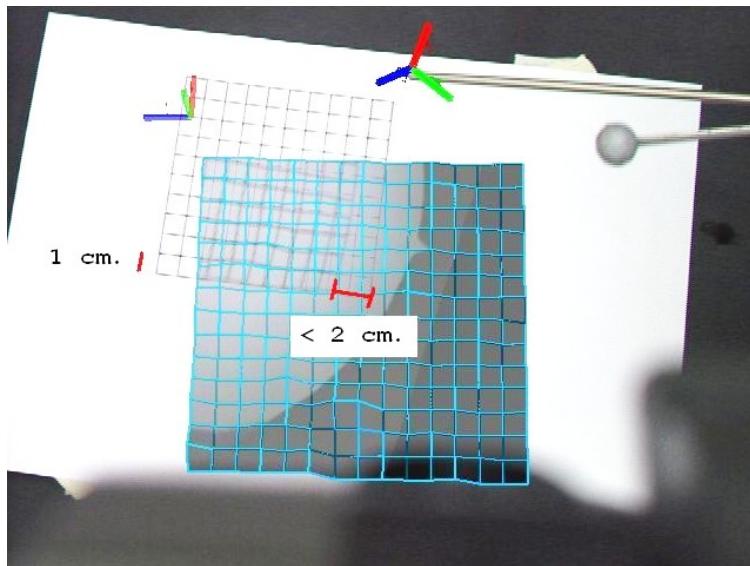


Figure 2.23: Rough accuracy estimation

We have initial results that show the order of magnitude of errors to be expected. However, we do not know yet what effects chiefly contribute to the back-projection error and whether our measurements generalize well. In order to complete the work, the sources of error are going to be analyzed.

The surgeons exposed to the technology agreed that undistorted visualization of endoscopic images is indispensable. However, the in-situ visualization could be an interesting addition to the classic visualization.

The automatic horizon recovery works stably. It has also been well-received by the surgeons.

2.3.3.1.5 Discussion and Conclusion

We have presented a method for the integration of endoscopic images into in-situ visualization, including a way of depth measurement.

There are many open questions for this technology:

Is the data from the moving instruments sufficient for surface generation? Can the grid approach model the surface well enough? How should the grid be parametrized?

These questions are going to be investigated together with the question whether depth data can be recovered by time-of-flight technology.

The initial results are exciting from a technical and medical point of view. From a technical point of view, it is a novel approach to build up a model by implicit interaction with the system. Via the interaction the model is continuously updated, creating dense dynamic data from relatively sparse data. From a medical point of view, the suggested visualizations of endoscopic images allow for novel aids that support the average endoscopic surgery. An efficient guidance to a former endoscope position is possible by picking a return point in the image. This point in the image is known in three dimensions. For this task the accuracy is subordinate since the assistant surgeon is able to recognize the region when they approach it.

2.3.3.2 Medical navigation

The visualization of spatial data has different objectives. A typical application of intraoperative visualization is instrument guidance.

Traditional navigation systems visualize this type of navigation on a screen using orthogonal planes and a plane which includes the instrument. This is useful for navigation to a certain point in space without prescribing the path. Ordinary navigation systems realize 6D navigation, such as for instance drilling, by starting from an entry point and guiding the user to a point in space. In laparoscopic surgery, the entry point is not directly accessible and thus it is not possible to guide first to the entry point and then to the exit point since the instruments are rigid.

Our experiments on IM nail locking (see section 4.2.2) suggest that angular differences are not visualized well by simple projection of the line that defines the 6D position of a screw. (see also figure 2.25).

We propose to visualize the 6D guidance as a guidance from the entry point to the exit point. A plane through the entry point (red) and one through the exit (blue) is visualized. Each plane is orthogonal to the planned line. The planes are used to intersect the virtual extension of the instrument with the planes. With this visualization it is possible to guide a path without starting at the entry point. By this means, the error that is expected at the entry and the exit point is visualized. The implementation of the visualization has been realized in the scope of the Master thesis of Martin Schulze.

Further research is planned to evaluate the effectiveness of this kind of visualization.

2.4 Interaction

Classic two-dimensional computer interaction paradigms such as windows, mouse pointers, menus, and keyboards, do generally not translate well for three-dimensional displays. Bowman [24] gives a comprehensive introduction into 3D user interfaces and detailed information why three-dimensional interaction is difficult. The book gives general advice

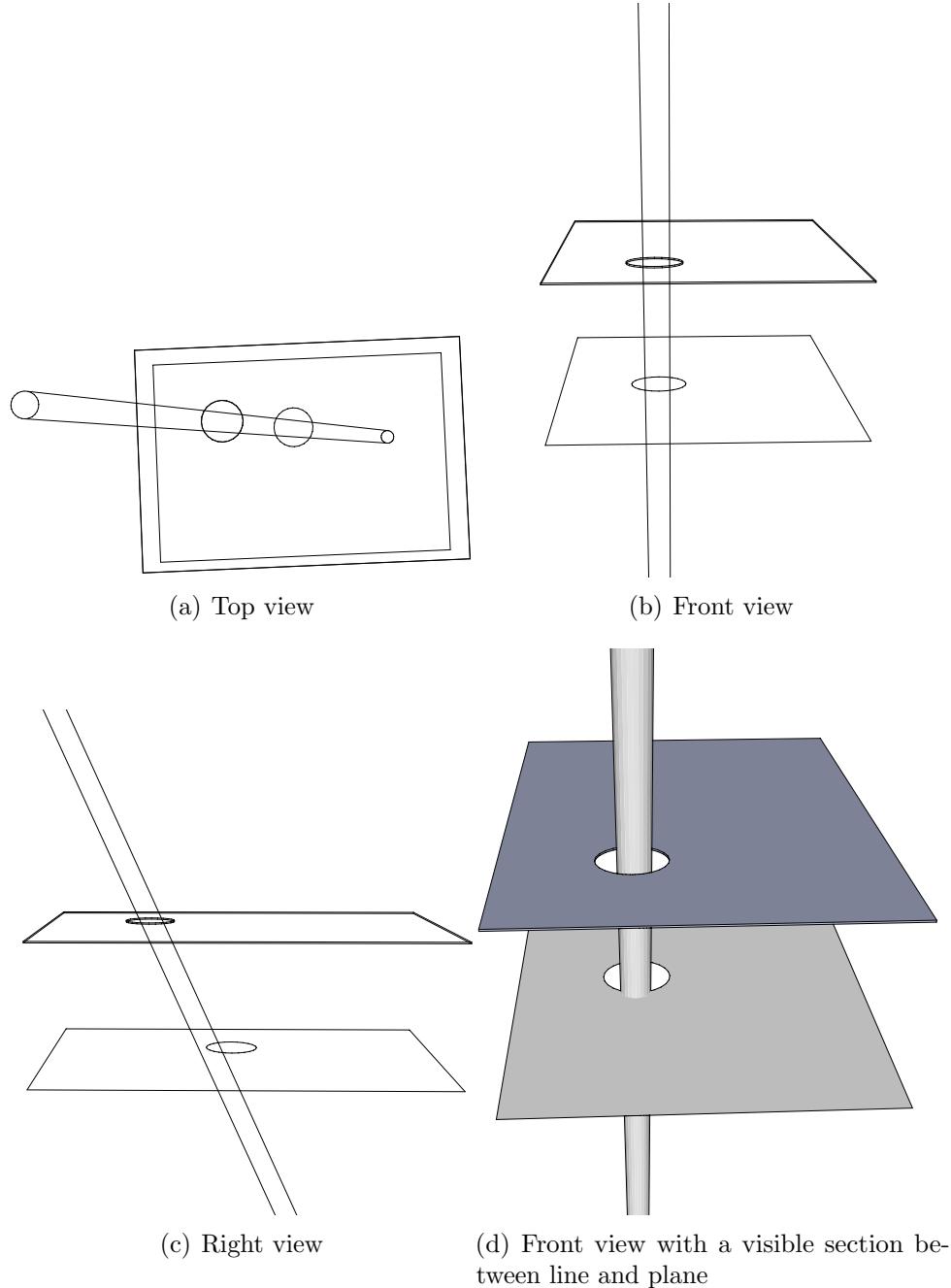


Figure 2.25: Four visualizations of the same geometry: In (a) and (b) it is not visible that the line does not traverse the plane through the hole. Visualization (c) benefits from a more favorable viewing direction. Visualization (d) has a similar viewing direction as (b), but the section between plane and the line reveals the misalignment.

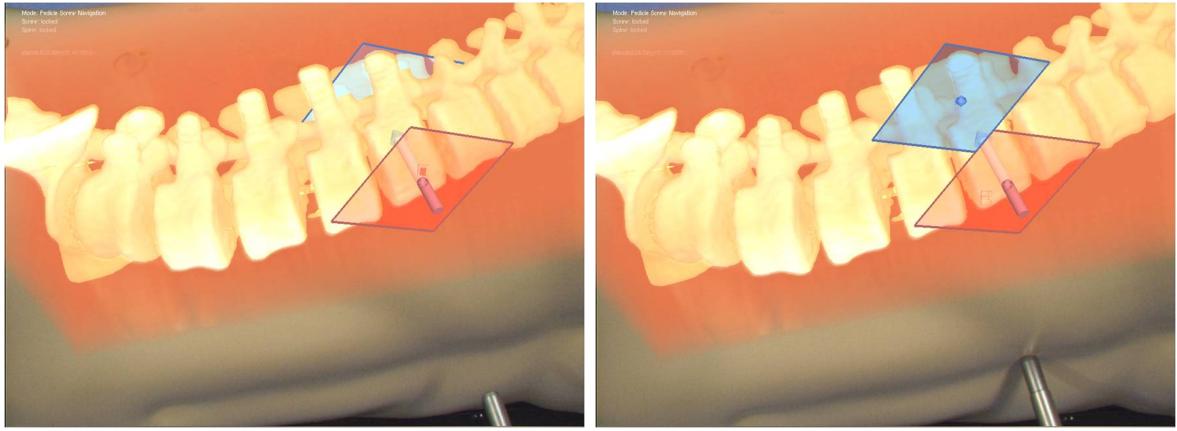


Figure 2.26: 6D navigation. Correct visualization of the planes (left), back plane in the foreground (right)

on creating new user interfaces. Reitinger et al. [129] suggest a 3D user interface for planning of liver surgery. Although the suggested planning is done in pure virtual space, the ideas apply to AR as well. They use tracked instruments and a tracked glass plane for defining points and planes in a tangible way. They combine tangible 3D interaction and classic 2D user interfaces in an effective way. Using tangible user interfaces can be an effective way of interaction, but it requires manual interaction, which is difficult when the user needs both hands for the job.

Navab et al. [118], [3] suggest a new paradigm for interaction with three dimensional data. A virtual mirror is augmented into the scene. The physician has the possibility to explore the data from any side using the mirror without giving up the registered view. Since this type of interaction uses a metaphor that has a very similar real pendant, no extra learning is expected for users. The virtual mirror can also be added to instruments, for instance in order to get a sideview of the instrument tip and the surrounding objects.

Besides from 2D/3D issues standard 2D computer interfaces, e.g. mouse pointers, are in particular not suited for the OR because of sterility and ergonomics reasons. Fortunately, medical systems are highly specialized on a specific therapy. Since a specialized application has a limited number of meaningful visualization modes, the user interface can be highly specialized as well. Context-aware systems can further reduce the degree of interaction. Temporal registration can be a key for minimizing the interaction with the system as much as possible.

CHAPTER
THREE

INTEGRATION

The whole is more than the sum of its parts.

Aristotle, 384 BC-322 BC

This chapter will describe the integration of the components of a medical AR system. It will first highlight with the requirements elicitation for the generic medical AR software framework that has been created for thesis. Alternatives to introducing a completely new software will be discussed as well. The design of the software framework will be explained in detail. The chapter ends with a detailed description of the AR system of the project NARVIS. It is based on the previously explained software framework.

3.1 Requirements

The integration of the parts of an AR system is non-trivial task. The following numbered requirements will be refined to characterizations of the system.

Each part (tracking, registration, visualization, interaction) requires a specialist in the field in order not to be the weakest link in the chain₁. The requirements of each part must be integrated into the design of the whole AR system₂. In order to integrate contradictory general design goals, a software framework should supply customizable services that realize a compromise₃ (as opposed to preferring certain design goals by default). Since AR and its components are still a subject of research, the design of an AR system needs to facilitate a flexible combination of different technologies and a flexible integration of new developments₄. Particularly medical applications require means of measuring the reliability of the output₅. According to the definition of AR, the system has to run in real-time₆.

These requirements may look vague at first sight, but they characterize the resulting software. Therefore, a medical AR system should have the following characteristics:

1. Requirement 1 favors specialists working on each part. As a consequence, it should be possible to developed independently from each other. This requires a library with common interfaces.

2. The complete set of features of each part should be accessible after integration (Requirement 2). This is especially important for temporal properties.
3. Requirement 3 stipulates that a trade-off between contrary design goals should not be made at the design level of the software framework. A service (active software) or another way of customization (passive software) at run-time level are more appropriate.
4. The flexible integration of parts (requirement 4) requires a *high level design* using abstract interfaces. Recombination of components should be possible without new compilation. Addition of new features or components should not cause recompilation and relinkage of the whole software.
5. A system that is able to provide a measure of reliability (requirement 5) must be able to control each component. To ensure reliability of data during concurrent events, the design must include mechanisms for synchronizing of the data access of the components.
6. The definition of the real-time property is fuzzy, as there are different definitions of *real-time* in computer science. Since characterizations 1-5, which are deduced from the definition of medical AR, demand something similar to a high-level operating system, the thesis will use the real-time definition for operating systems. IEEE POSIX Standard 1003.1 defines: "Realtime in operating systems: the ability of the operating system to provide a required level of service in a bounded response time". The bounded response time can be defined by the update rate of the incoming data. This means that the system should be able to handle incoming data on the fly.
7. A real-time system does not only need services and interfaces for complying with a certain response time. It must also offer means of exploiting the maximum performance of the hardware in order to adhere to the time limit. *Low level access* to hardware and software components is desirable.

The framework has been realized in C++ in order to comply with the requirement of full hardware control while also incorporating means of high abstraction. Class inheritance has been heavily used in the component design to allow high abstraction using top classes or specialized classes for more specific control.

3.2 Other AR frameworks

There are different AR frameworks known in the community.

- ARToolkit [78] is probably one of the first approaches to provide a software framework for AR. It is mainly a toolkit for monitor-based AR where the real view provides the tracking data via image processing. The simplicity of the concept established its success as it allowed people to produce AR applications without owning tracking hardware and without experience in computer vision. Other toolkits similar to

ARToolkit have been introduced to the community (e.g. ARTag [47]) using improved computer vision algorithms.

- The RAMP system[142] uses software that is specific to its hardware. The software is designed for high performance in terms of short latency, high resolution, and low relative lag between real and virtual objects. However, the design does not entail flexible scheduling of components. Scheduling is given implicitly and is optimized for framegrabber image input.
- DWARF [8] is an AR framework that can be compared to a distributed operating system. The main idea is that components like sensors, cameras and applications are distributed in a network and connect to each other via CORBA. This type of system may be dynamic and change components at runtime. The system is started via scripts which start up the autonomous components. The distributed design is the strength and the weakness of DWARF at the same time. The concurrent behavior is almost unpredictable in detail, the initialization of the distributed services can be tedious, and data transfer between layers could be faster. However, it allows for the realization of spontaneous and distributed systems.
- Studierstube [145] is a framework that has a focus on high abstraction of scene visualization. Data tracking is provided by a high abstraction component called OpenTracker. Transparent use of tracking data in scenegraphs makes the design of complex scenes and interactions possible (as e.g. used in [129]). Many applications have been designed with this framework because of its design that makes it simple to create a new application or even a new AR system.
- DART [102] is a high-level AR framework based on Macromedia Director. It aims at high level content generation using the commercial and widespread host software. Building on professional media software supports the generation of compelling AR content.

For state-of-the-art research, our group needed a software framework that can be set up as easily as ARToolkit, is as modular as DWARF, as transparent in handling as Studierstube, and allowing for the same performance as RAMP. However, none of these frameworks covers entirely the above characteristics of a medical AR system. Choosing one of the above frameworks implicates a compromise between complexity and performance, although system performance and simplicity of programming interfaces do not contradict each other in theory. None of the frameworks offers services to monitor real-time properties. The enhancement of an existing and successful framework would not have been a viable option, since characteristics 5 - 7 (reliability and real-time concerns) introduce deep-seated methods that have to be integrated into the design of the whole framework. Therefore, a new software framework has been designed, which can be comparable to a microkernel design of a computer operating system. It has been realized jointly with Marco Feuerstein, Jörg Traub, and Oliver Kutter.

3.3 Framework design

This section describes the design in accordance to the requirements and characteristics of section 3.1.

3.3.1 Programming language and environment

In order to comply with characteristic 7, which requires full hardware control and provide means of high abstraction (cf. characteristic 4), the ideal programming language appears to be C++. It has been designed to allow for programming close to hardware while offering high-level structures, such as object-oriented programming [167]. By design, it allows for faster response times than any interpreted language, like e.g. JAVA or Perl.

Programming environment	
Language	C++
IDE	Visual Studio 2005
Version Control	Subversion
Dependencies	STL, boost, loki (no impact on license, no impact on OS)
Target operating system	Windows XP

The software subversion is used as the version control system. High quality libraries like STL [111], loki [3] and boost have been used together with state-of-the-art programming paradigms [57], [110], and [109].

The target operating system is Microsoft Windows XP which is used by all of the developers because of the availability of drivers for recent hardware. However, the design does not

include operating system-specific code that cannot compile and run under another operating system, like for instance, Linux.

3.3.2 Program structure

In order to comply with characteristics 5 and 6, the software framework needs an active software component that is able to control the (concurrent) components. It must be able to make sure that all components are ready and find out dynamically whether the real-time property was met.

The AR framework with an active control component acts like a high-level operating system, which manages AR resources, provides abstract interfaces to hardware, and offers services that AR applications need for an effective implementation.

3.3.2.1 Micro kernel concept

The naive approach for a structure with a central control component would be a software framework consisting of the control software containing all other parts as associations. However, this would be quite limiting for the design of the components, their interaction and recombination. For compliance with characteristics 3 and 4, the design of the central component has been conceived to be minimalistic. It only has means of loading components and assessing their temporal performance. However, the actual behavioral performance is defined by loaded components. Hence, the central control software offers full access to

each component, while decisions on the program flow are computed in at least one of the components, which can be changed flexibly.

A network of independently running processes, as suggested for DWARF, would neither offer the same level of access due to memory access restrictions, nor a rigid scheme for scheduling. Both are helpful for maintaining real-time properties (characteristics 6 and 7).

3.3.2.2 Scheduling

Since the AR framework behaves like a high-level operating system, in which arbitrary components can be loaded and turned into an active state, it needs to define a scheduling pattern for the components. The lightweighted control software (kernel) loads the components (`Device`) and sequentially provides them with computation time by calling their function `stateUpdate`. The sequential call of the components seems limiting at first sight, but it can guarantee that no component is waiting for the next CPU time for longer than the time a single output of data takes. If the time difference between two outputs is less than the desired value for computations, the system fulfills the real-time property (characteristic 6). Note that this kind of scheduling is not possible using components as processes in an operating system, since that would require the scheduling algorithm to be able to configure time slices flexibly. The detection of timely real-time computation is still valid in case the data is not generated in the function `stateUpdate`, but in a callback function triggered by an interrupt. This is because a callback is executed immediately. The same is actually true for computations in threads that have a higher priority than the output component.

In the software framework, all active components are derived from the class `Device` and implement `stateUpdate` and the initialization function `configure`.

3.3.2.3 Configuration

The software framework configuration via XML has been introduced to cover characteristic 4. The implementation of the configuration allows for an easy-to-use and easy-to-extend configuration via a human-readable text file, saving a configuration of a running system as well as interactive configuration changes in a graphical user interface.

The XML-based serialization makes use of the object factory pattern (cf. [57] and [3]) and the composite pattern. Object factories create specific instances of classes that can be chosen at runtime depending on the configuration of the program. This offers a maximum of flexibility for a compiled program. Combining this feature with a plugin system offers a maximum of extensibility without recompilation of the whole framework.

XML configuration is realized in a recursive scheme. On the implementation level, it looks as follows: The XML tree is read into a certain variable by the function `node->handleNode(variable&, "subNodeName")`. The node `<subNodeName>...<\subNodeName>` is searched in `node` and its content is assigned to the variable. If the variable is a non-recurring type the content is assigned directly. These are

- standard types (e.g., `int`, `double`, ...)
- strings

- arrays of selected data types (`std::string`, `std::vector<double>`)
- selected classes of our common library (`CAMP::Matrix4<double>`, `CAMP::Vector3<double>`)

Recurring types are all classes that inherit from `MappableObject` and implement the function `xmlConfigure`, a clone function, and the registration function. The registration function simply notifies the object factory that a class XYZ is accessible without the need of including the header of class XYZ in the object factory class. In case of a recursing type, an object is created, its function `xmlConfigure` is called and the subnodes of the XML node are evaluated.

XML configuration is also able to read and instantiate arbitrary lists of objects into an STL container. What is interesting about it is that it can either create a set of objects for which the type may be specified at compile time or in an XML file, using the type specifier. The latter makes the design very flexible and allows for XML configuration of plugins that did not exist at compile time of the framework core. In all other cases, the type need not be specified in the XML file, since it is defined by the program code. This allows for minimal redundant information and makes the XML file easily readable and maintainable.

The function `handleData` only defines the link between a variable and the node name, as opposed to defining how the variable is processed. Complex types are defined recursively and trace back to basic types. These basic types are defined in a central class for reading nodes. Hence, it is possible to create a full write-out of the XML-configurable variables without changing any line in the XML-configurable classes. Only the polymorph function `handleData` needs to be overwritten. The function is defined in the node class and has an implementation for each basic type. Since the system features nodes for writing and reading, it offers round-trip serialization.

With the same technique, a simple but effective editor has been created that allows for direct manipulation of XML-configurable variables during runtime for debugging purposes. All variables that can be configured by the XML file can be accessed and altered in this graphical editor.

3.3.2.4 Overview of component types

The system basically includes components for:

1. Dynamic incoming data, for instance tracking data and real-time imaging data. Each piece of data is stored into the ring buffer including a time stamp that corresponds to the time of measurement (as opposed to the point of arrival in the system). Examples can be found in figure 3.2 on page 73
2. Generating and controlling graphical user interface elements. Currently, there is a choice between toolkits Qt and OpenGLUT (bridge pattern).
3. Synchronization of incoming data in order to determine the best time to schedule a new output. Since there is no optimal strategy for all purposes, the strategy can be tailored to the actual system (strategy pattern).

4. Output generation at different levels of abstraction in order to provide a choice between low level (for detailed specification) and high level (convenient programming interface) abstraction.

The simplified class diagrams in figure 3.1 on the next page give a short overview of the structure.

3.3.2.5 Code organization

This subsection deals with the structure of development. It is not specific to medical AR, but it might be worth reading how the software has been created. The code organization has been defined by the initial PhD students of CAMP (Jörg Traub, Wolfgang Wein, Marco Feuerstein and the author). Since it is difficult to predict the requirements of scientific software – apart from very abstract assumptions such as those in section 3.1 – and the requirements are likely to be different for each PhD student, an ordinary software design is virtually impossible to develop. Nonetheless, we aimed at reusing code without creating organizational overhead. Thus, we agreed on the basic interfaces of the framework and organized all code in a single repository in folders according to their reusability.

The code is divided into three shells in order to increase the chance of code to be reused without needing a dedicated person to take care of the quality. Each shell is compiled and linked separately.

- Common code and interfaces
- Shared code
- Personal code

Common code provides the common classes and interfaces that are used at CAMP. It does not have references to other libraries than itself. Changes are in general not welcome, since most of the code depends on this code.

Shared code is supposed to be in a stable state. Extensions are welcome unlike changes to the interfaces, because other shared or personal code may depend on it. Shared code is subdivided into several libraries for a maximum encapsulation of dependencies.

Personal code can be changed without restrictions. It may depend on other libraries, but not vice versa.

This principle has been applied to the AR framework and is applied to other libraries that are created at CAMP¹. It allows for a decentralized organization of code while offering a maximum of freedom for each programmer.

The use of common code and interfaces reduces the chances of data type incompatibilities when elevating code from personal to shared code. Only the position of code in the folder structure tells the other programmers how stable particular interfaces are and whether the code may be reused.

¹Chair for Computer Aided Medical Procedures

Integration

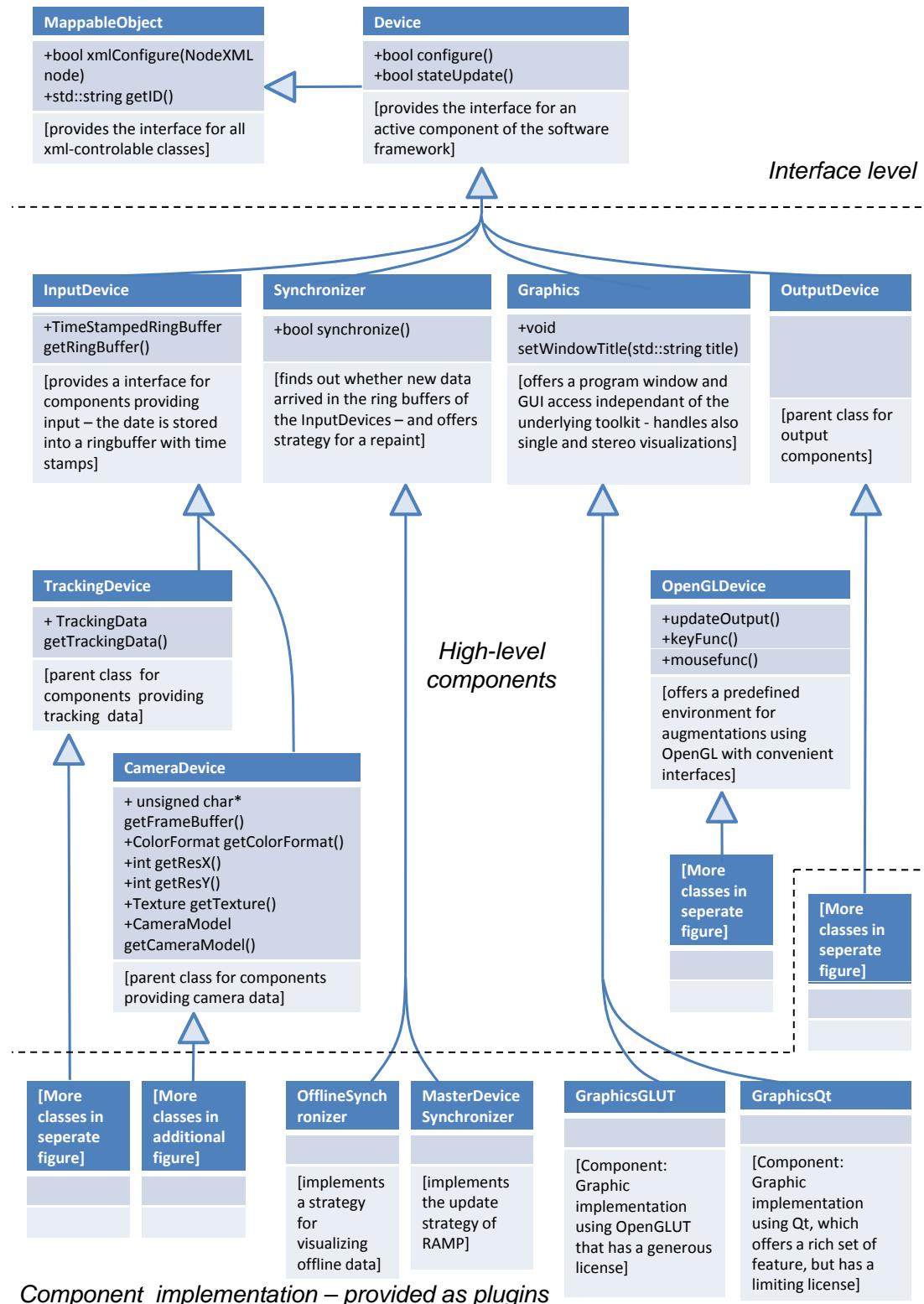
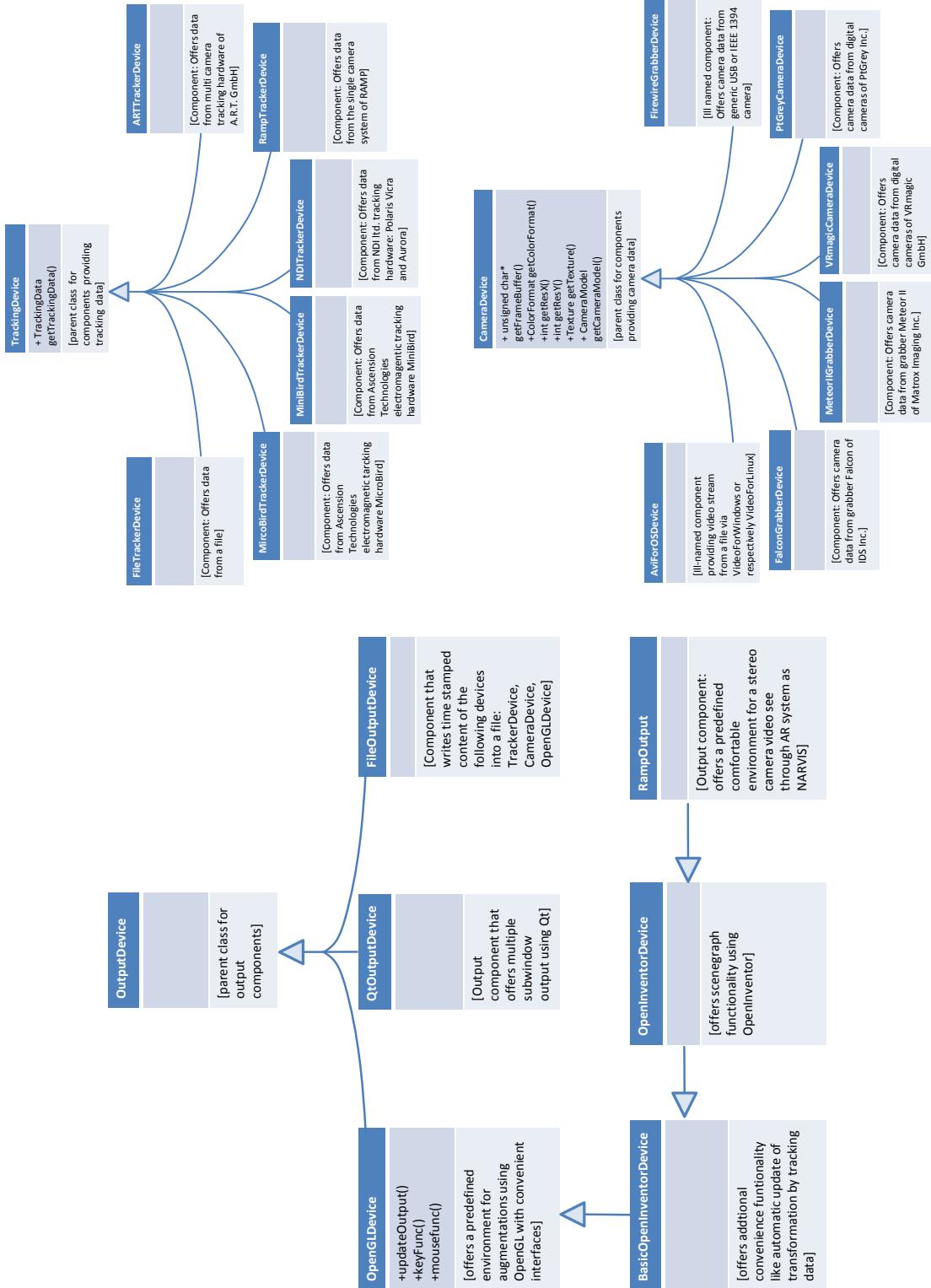


Figure 3.1: Simplified class diagram of the components, which are modeled with the class **Device**. Only characteristic functions are depicted. The functions are only depicted in the top class providing this interface.


 Figure 3.2: Simplified class diagrams of `TrackingDevices`, `CameraDevices`, and `OutputDevices`

3.3.3 Real-time constraints

The current system offers the following means for keeping the real-time constraints

- Time-stamped and buffered dynamic data
- Memory access to buffered and time stamped data for all components
- Semaphore-like flag for avoiding concurrent read during a write operation
- High precision clock without drift
- Conversion between different clocks such as CPU counter, bus clock, and the clocks of other computers via network time protocol (NTP)
- Sequential scheduling (see section 3.3.2.2)
- Strategies for synchronized visualization of dynamic data (see section 3.3.4)

This set of services allows to check within components – in particular output components – if the other components work at the necessary speed and react appropriately on failure.

3.3.4 Synchronization issues

Time synchronization of tracking data and video images is an important issue for an augmented reality system. In general, a mismatch in time between different data streams results in visualization errors although each data stream offers correct data. Holloway et al. [73] investigated the source of errors for augmented reality systems.

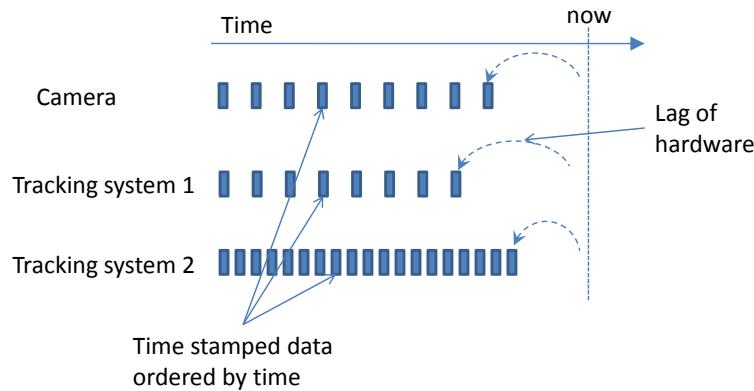


Figure 3.3: Synchronization of dynamic data. The exemplary system consists of one camera and two tracking systems.

augmented reality system that synchronizes tracking and video data by means of hardware triggering. Their software waits for the slowest component before the visualization is updated. For endoscopic surgery, Vogt [177] also uses hardware triggering to synchronize tracking and video data by connecting the video signal of the endoscope system to the synchronization card of the tracking system.

The errors of time mismatch can raise to be the most significant error sources when the camera is moving. In order to solve this problem, Bajura et al. [7] propose to “effectively reduce the delay in measuring image registration to zero by delaying the the real video image stream” in case the tracking system is the slower component. Jacobs et al. [74] provide general methods for measuring and compensating relative latencies by software. Sauer et al. [143] describe an aug-

If virtual and real images do not show a relative lag, the images are consistent and there is no error due to a time shift. However, there is still the visual offset to haptic senses. A method for measuring this latency is presented in section 4.1.1.

For data synchronization, a system needs time-stamped ring buffers, an accurate clock, means of determining the measurement time, and a strategy that implements the desired temporal behavior.

3.3.4.1 Time stamped ring buffers

In order to be able to synchronize data of different sources, the system must retain the information once the data has been measured. Since the hardware lag varies in the range of several milliseconds, it is insufficient to use the point of time at which the data arrives in the system as an approximation. As it is possible that a piece of hardware provides new data although older data fits temporally better to the other data in the system (see figure 3.5), dynamic data has to be stored into buffers. Thus, the system can partially use old data to realize a synchronized visualization of data. This is only possible if the time stamp of the data is coherent within the system.

3.3.4.2 Computer clocks

Current computers have different kinds of clocks with different resolutions and accuracies. There is the PC clock, which offers absolute time and date. It is a low resolution timer with a frequency of 18Hz, which is far too inaccurate for data synchronization in an AR system. PC clocks can be synchronized via NTP in a network.

For a higher resolution, the CPU² cycle clock offers the number of cycles after computer startup. This high-resolution timer shows some drifting because the frequency of the quartz is only roughly known by the computer system. Repeated estimations of the frequency using the PC clock can merely reduce the issue of drift rather than solving it.

The software framework offers a general purpose timer class, which combines the rough PC clock and the CPU counter. It provides a high accuracy timer of absolute values allowing for exchanging time-stamped data with other computers in a network. This is necessary since many tracking systems compute the data on a different machine with a different clock for time stamps.

Last but not least, each IEEE 1394 bus provides a timer. It offers a high temporal resolution and low drifts, but it is only capable of representing a small time span. Especially time stamps that are offered by digital cameras use the timer of the connected bus. The software framework provides a conversion function between the general purpose timer and IEEE 1394 buses.

3.3.4.3 Determination of measurement time

For data synchronization, the exact point in time of measurement has to be determined for all dynamic data, which consists at least of tracking and camera data.

²central processing unit

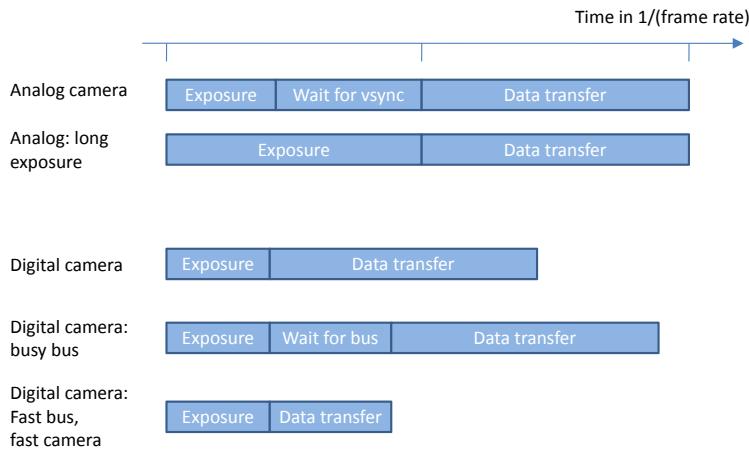


Figure 3.4: Timing of analog and digital cameras

Analog cameras and digital cameras employ quite different timing methods (see figure 3.4).

Analog cameras have a regular data flow, because they produce an analog image signal. This continuous image signal is determined by the vertical synchronization (vsync) signal that notifies a display that the new image begins after the signal. Knowing the update rate and the exposure time of an analog camera, it is easy to calculate the beginning and the end of the exposure time from the time of arrival in the system. The digital version of the image is generated from the analog signal with auxiliary framegrabber hardware. Since framegrabbers typically offer a callback function that is called when starting to grab, it is straightforward to compute the measurement time of analog cameras.

Digital cameras have a more variable timing pattern (see figure 3.4). When the exposure is finished, the camera tries to send the data to the computer via a bus. Depending on the bus traffic, camera transfer speed, and the workload of the receiving computer, the transfer time may vary significantly. Some digital cameras include a mechanism that encodes the exposure time stamp into the image. A few pixels are lost with this method, in exchange for an accurate estimate of measurement time. If the time stamp is given in bus clock units, it has to be converted to the general purpose timer of the software framework.

Optical tracking systems often offer an exposure time stamp. Since their exposure is very short, the influence of when exactly the time stamp was generated (beginning, middle or end of exposure) is neglectable. If the tracking data is obtained from a different computer, it is necessary to synchronize the timers of the computers involved. The recommended way is using NTP. The use of local NTP servers to synchronize the time of the AR system is advisable, since the time can be determined more accurately in a local network due to shorter response times.

3.3.4.4 Strategy for synchronization and timing of visualization

For the update of augmented images, there is a choice of data in the buffer. The correct choice of data results in synchronized visualization.

The point of time for an update can be freely chosen. This instant is implicitly given in many AR systems as the next point in time at which the CPU is able to process the next augmentation. This behavior can be suboptimal with regard to latency, because new data arriving during the visualization cannot be integrated. Such new data can only be shown in the next visualization update. The latency is expected to be shortest when the data is processed as soon as possible. A strategy that updates when new data has arrived is not efficient if one input device produces a high number of updates.

A better strategy updates when the point of synchronization has changed (see figure 3.5). Some AR systems (e.g. RAMP) include this strategy implicitly by triggering the update on the arrival of new tracking data of the slowest component. However, hard coding can be problematic when adding new hardware to the system or upgrading the slowest component. The component that models the strategy for synchronization and timing of output in the software framework is called `Synchronize`.

Different strategies have been implemented. The default is a general purpose strategy that updates when the synchronization point changes. There is another strategy (`OfflineSynchronizer`) for file input (as opposed to live input) and one (`MasterDeviceSynchronizer`) that triggers updates only if new data has arrived from a certain `InputDevice`.

3.3.5 Results and examples

The framework has proved to be a reliable and flexible tool for modeling different compelling medical AR prototypes which have been used for international publications in, respectively at, AR and medical imaging conferences and journals, as well as for public demonstrations. The prototypes based on this framework include the following:

- HMD-based AR
 - Four NARVIS systems (Christoph Bichlmeier and the author) [12, 11, 4, 5, 22, 118, 16, 155]. For a detailed description refer to section 3.4.2)

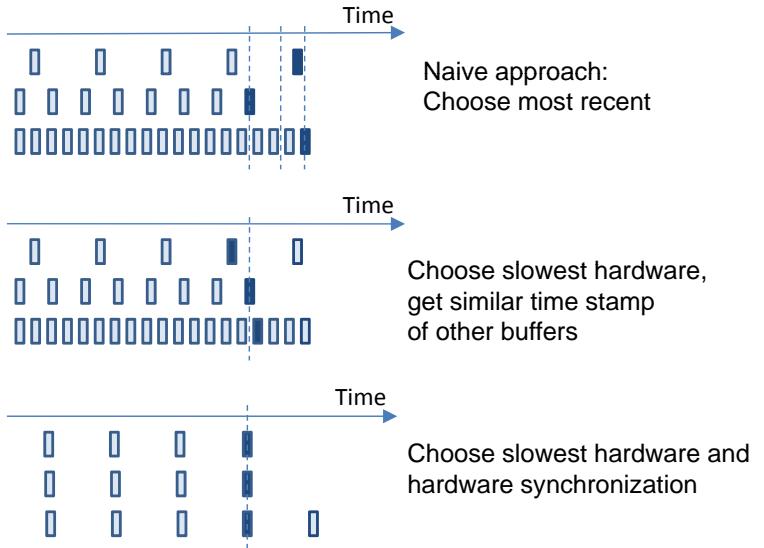


Figure 3.5: Different behavior of an AR system using the same hardware with different synchronization strategies: No exploitation of buffer (top), synchronization using time stamps (middle), and hardware synchronization (bottom)

- **Monitor-based AR** (as described in chapter 1.1.4)
 - Audiovisual augmentation of β - and γ -probe data on a screen (Thomas Wendler et al.) [190, 188, 189]
 - Bronchoscopy navigation (Jörg Traub et al.) [85]
 - Orthopedic navigation (Jörg Traub et al.) [24]
- Augmented Endoscopes (as described in chapter 1.1.5)
 - Laparoscope augmentation (Marco Feuerstein) [46, 45, 43, 44]
 - Arthroscope augmentation
- Augmented C-arm (as in chapter 1.1.6)
 - CamC (Jörg Traub et al.)[170]

3.4 Implementation of NARVIS

The AR system NARVIS has been realized using the software framework. One version uses the same hardware as RAMP and it models the same temporal behavior. However, it does not rely on any of the former software.

In the NARVIS project, two other systems and an experimental setup have been realized.

3.4.1 Software components

This section is structured in the same way as chapter 2, since NARVIS needs all of the components described there.

3.4.1.1 Spatial registration

3.4.1.1.1 Calibration

The calibration of the system is performed with a three-dimensional pattern of retroreflective circles on a black background (see figure 2.1). The circular fiducials in combination with a momentum detector generally offer better accuracy than crossing markers using an edge detector. Especially if the calibration pattern is out of focus the momentum-based feature detector offers superior performance compared to the edge detector. The three-dimensional spatial arrangement of the pattern allows for a lower number of images in the calibration process. The retroreflective fiducials of the calibration pattern have the advantage that they are visible for infrared tracking systems as well as for cameras in the visible spectrum. The three-dimensional arrangement of the calibration pattern was determined with an industrial measurement system that uses multiple infrared cameras.

The calibration includes estimation of the internal parameters of the cameras including the infrared camera of the single camera tracking system. Furthermore, the transformations

between the color cameras and the tracking camera have to be estimated accurately in the calibration process. In principle, this procedure is not complicated: The camera parameters of each camera are estimated with the method of Tsai et al. [173]. The transformations ${}^{TC}\mathbf{T}_L$ and ${}^{TC}\mathbf{T}_R$ from the tracking camera to each of the color cameras can be computed from the external camera calibration ${}^{TC}\mathbf{T}_P$, ${}^L\mathbf{T}_P$, and ${}^L\mathbf{T}_P$ (camera to pattern).

$${}^{TC}\mathbf{T}_L = {}^{TC}\mathbf{T}_P \cdot {}^L\mathbf{T}_P^{-1} \quad (3.1)$$

This is only possible if the calibration pattern is not moved - relative to the cameras - in between the calibration processes for each cameras. It is numerically unstable for the calibration of the internal parameter if the pattern is forced to be in a position that is visible for all three cameras. It is particularly unstable if the camera objectives have different focal lengths. It is therefore recommended to perform the internal calibration separately ideally using a number of different views. In a second step, the internal camera parameters are used as static parameters in the optimization step of the calibration method. This is the same procedure as the one used by Sauer et al. [142, 143].

The calibration routine is integrated in the software framework as an output device (`CameraCalibrationGUI`) by Markus Urban.

3.4.1.1.2 Tracking

In the setup of NARVIS, the tracking data is provided by a combination of two tracking systems (see figure 3.6). Both use circular or spherical retroreflective fiducials and infrared flash lights. One is an inside-out tracking system that provides the head tracking relative to a reference target. The other system provides tool and patient tracking relative to the reference target. Since both system use the same tracking targets and the same way of generating the coordinate system from the fiducials, no additional registration is required. The transformation from the coordinates defined by the reference frame into instrument coordinates is provided by the simple formula:

$$\mathbf{T}_{Frame2Instrument} = \mathbf{T}_{Ext2Frame}^{-1} \cdot \mathbf{T}_{Ext2Instrument} \quad (3.2)$$

Two tracking systems are used in order to exploit the high rotational accuracy for head tracking of the inside-out tracking and relatively small tracking targets of the outside-in tracking system that has a larger baseline.

The inside-out tracking system is a custom-made construction using only a single camera. The setup is similar to RAMP [142, 143]. It comprises a wide-angle lens, an infrared light filter, and an infrared flash that is triggered by the camera. Markers are detected in the camera image by means of a connected component analysis, while the position is determined with a center-of-mass method using grayscale values. The target position is calculated with the algorithm proposed by Tsai [173]. Before running this algorithm, is it necessary to match the uniform markers to their counterpart in the three-dimensional model. This is solved by an algorithm proposed in the author's diploma thesis[150]. The single camera tracking system needs in practice eight markers [179] for sufficiently accurate, reliable, and robust tracking although in theory six markers suffice. The error function of targets of the single camera tracking is unequally distributed in space

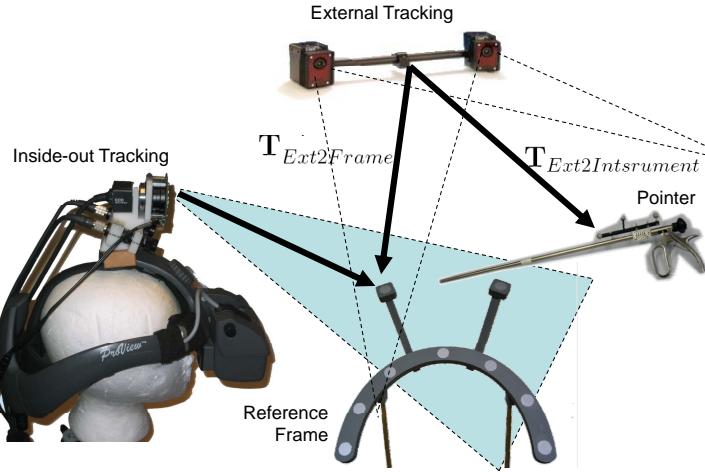


Figure 3.6: Inside-out and outside-in tracking of NARVIS

[179]. The lowest accuracy is that the view orientation. For real-time augmentation this is fine, however, because a satisfactory overlay can still be achieved with lower directional viewing accuracy.[72].

The outside-in tracking system is an off-the-shelf industrial multiple camera system that allows for variable camera positions and a variable number of cameras (produced by A.R.T³).

3.4.1.1.3 Patient registration Two different implementations of patient registration have been realized for NARVIS.

Firstly, there is point-based registration of radioopaque and retroreflective fiducials. This does not require interaction, but requires the fiducials to be visible and remain in the same spot as in the imaging device. The extraction of fiducials from the imaging data and correct assignment to the optical fiducials is performed automatically, in a fashion similar to the approach of Wang et al. [184]. After an automatic segmentation step, the shape of the fiducial candidates is checked. The position of each fiducial is computed with the center of mass of the segmented voxels. The three-dimensional positions are assigned by means of the method proposed by Umeyama [175]. In the context of spine surgery the idea is to attach a tracking target rigidly to a fixateur that has already been implanted prior to the surgery as a first step of the treatment. Two scenarios are possible with this approach. If the tracking target can be fixed to the implant in a reproducible way the surgical team has to affix the target before the visualization is initiated. In the other case, the target has to be fixed before the imaging data is taken. Both scenarios do not involve any kind of interaction with the system. The automatic point-based patient registration has been realized within the scope of a interdisciplinary project of Latifa Omary and Phillip Stephan.

The other method of patient registration that has been implemented is often called *registration-free* [43], because the surgical team does not actively perform any additional

³Advanced Realtime Tracking GmbH, Herrsching, Germany

steps in their workflow. In this approach, the calibrated imaging device is tracked rather than the patient. The major drawback is that the registration is lost if the patient is moved. For navigation tasks, it is in many cases unrealistic to assume that the patient will not be physically moved in any way. However, the assumption is valid for intraoperative planning tasks, such as, for instance, the port placement.

3.4.1.2 Temporal registration

The work on temporal registration is still at a too early stage to be integrated into the operating room as a reliable component. However, the algorithms described in section 2.2 are accessible in the software framework and the research on movement comparison (section 2.2.1) has been done using the software framework and NARVIS.

3.4.1.3 Visualization

All types of visualizations that are mentioned in chapter 2.3 are implemented for NARVIS. The choice of the correct combination is a challenging task. First insights are presented in section 4.2. In order to be able to create scenes easily, NARVIS applications include an OpenInventor interface for rapid prototyping of scenes and objects. Objects can be integrated by a single additional line in the XML file.

```
<IVModels>MyModel1</IVModels>
```

With another additional line labeled transformations in the scenegraph can be automatically replaced by live tracking data.

```
<ReplaceIVNode>
<Probe>TrackingDataName</Probe>
<IVNode>LabelName</IVNode>
</ReplaceIVNode>
```

By this means, three-dimensional structures can be modeled using professional modeling software and integrated as an OpenInventor or VRML file without recompilation of the program.

3.4.1.4 Interaction

Similarly to the description in section 2.4, NARVIS adheres to the philosophy that explicit interaction with the system is meant to kept to a minimum. Only a single visualization with a preset of parameters per situation is offered in order to keep the system as simple as possible for doctors. That visualization should contain all required information.

The virtual mirror concept is also available in NARVIS, as it is available in the software framework.

3.4.2 Hardware description

The hardware setup of the AR system that has been mainly used and extended in the NARVIS project was originally developed at Siemens Corporate Research in Princeton,

NY, USA. The original system as described in 2002 was an HMD-based video see-through system. It included a inside-out tracking system based on an infrared camera and a triggered flash. The original system and three new similar setups are described in figure 3.7. The latency in the specification table has been measured using the method as described in section 4.1.1.

System description 1: Classic system

HMD resolution	1024x768 px, 2.34 arcmin/px
HMD contrast ratio	40:1
HMD field of view	40° (H) x 30° (V)
Color camera resolution	640x480 px (interlaced, analog)
Color camera update rate	30 Hz
Lens	M-Mount 8mm
System Weight	~ 1 kg
Max. system framerate	30 Hz
System Latency	~100 ms


System description 2: New light weight system

HMD resolution	800x600 px, 2.25 arcmin/px
HMD contrast ratio	200:1
HMD field of view	32° (H) x 24° (V)
Color camera update rate	48 Hz
Color camera resolution	1024x768 px (progressive, digital)
Lens	M Mount 6mm
System Weight	~ .3 kg
Max. system frame rate	Difficult to say: Triggering of tracking and video not possible
System Latency	~100 ms


System description 3: New high resolution system

HMD resolution	1280x1024 px, 2.25 arcmin/px
HMD contrast ratio	200:1
HMD field of view	48° (H) x 36° (V)
Color camera update rate	30 Hz
Color camera resolution	1024x768 px (progressive, digital)
Lens	C/CS Mount 8mm
System Weight	~ 1 kg
Max. system frame rate	30 Hz
System Latency	~100 ms


System description 4: Experimental setup

HMD resolution	-
HMD contrast ratio	-
HMD field of view	-
Color camera update rate	200 Hz
Color camera resolution	640x480 px (progressive, digital)
Lens	C/CS Mount 6mm
System Weight	200g per camera
Max. system frame rate	Limited by display refresh rate
System Latency	<50 ms

Figure 3.7: The AR systems

CHAPTER
FOUR

ASSESSMENT

An important scientific innovation rarely makes its way by gradually winning over and converting its opponents: What does happen is that the opponents gradually die out.

- Max Planck (1858-1947)

This chapter describes assessment of medical AR. It includes new methods for assessment of AR systems as well as a description of assessment steps that have already been performed for the NARVIS. The experiments reveal the current bottlenecks of the system, which is regarded to be state-of-the-art. Therefore, any outcome can be considered as exemplary for current medical AR. Two new methods for objectively measuring two of these bottlenecks are introduced in this chapter.

After choosing components and integrating them into a system, the final task is the validation. As stated in the introduction (cf. section 1.2), medical AR is a disruptive technology. Thus, its validation is not as straightforward as that of sustaining technology, which can be compared to the previous level of development with established ways of benchmarking. These benchmarks display the performance of the key properties of the established technology, but may not sufficiently take the new properties into account. Furthermore, the direct comparison between an established technology and a new one biases the result in favor of the established. Teething problems of a new technology cannot be resolved in an experiment and the long experience and personal preferences of subjects cannot simply be imitated in an experiment. Last but not least, the whole infrastructure of the established technology, as for instance professional teaching and integration into a certain workflow, is in favor of the established technology. This implies that any such comparisons are biased, which means they can only reveal useful information if the new technology performs better. With the opposite result is unclear whether that result was due to bias or the actual advantages and disadvantages of the compared technologies.

This argument line is not meant to discourage comparisons between emerging medical Augmented Reality and established technologies. In fact, any new technology should be compared in order to find out principal differences – however, not necessarily to evaluate in terms of *worse* or *better*.

In order to be able to understand the potential of an upcoming technology it appears reasonable to evaluate the potential of a state-of-the art system. For this purpose, an extensive review of other existing systems (cf. chapter 1.1), their algorithms, and components (cf. chapter 2) has been performed in order to compose a state-of-the-art system (cf. chapter 3). The final goal of a validation of a human-computer interface is answering the question whether it is usable and what makes it usable.

4.1 Technical evaluation

Two new methods are presented here that concern two key properties of an AR system: latency and spatial accuracy. To complete the section about technical evaluation, experiments on end-to-end accuracy of the system are described.

4.1.1 Latency estimation

Note: The following work has already been published in [155].

Latency is a key property of a video see-through AR system since users' performance is strongly related to it. However, there is no standard way of latency measurement of an AR system in literature.

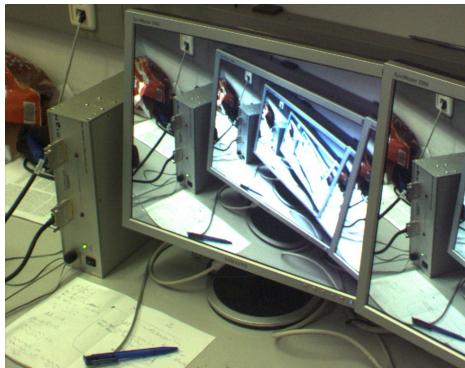


Figure 4.1: Camera feedback. We show that this way of encoding has an adequate accuracy for latency measurements. As the method allows for a series of automatic measurements, we propose to visualize the measurements in a histogram. This histogram reveals meaningful information about the system other than the mean value and standard deviation of the latency. The method has been tested on four different AR systems that use different camera technologies, resolutions and frame rates.

We have created a stable and comparable way of estimating the latency in a video see-through AR system. The latency is estimated by encoding the time in the image and decoding the time after camera feedback. We have encoded the time as a translation of a circle in the image. The cross ratio has been used as an image feature that is preserved in a projective transformation. This encoding allows for a simple but accurate way of decoding.

We show that this way of encoding has an adequate accuracy for latency measurements. As the method allows for a series of automatic measurements, we

4.1.1.1 Introduction

The purpose of AR systems is to augment the users with spatially registered data in real time. When studying the benefits of augmentation we need quantitative specifications of the AR system in order to distinguish effects of the system performance from those of human performance.

The response speed of a video see-through AR system has definite effects on the performance of users as shown by, for instance Lai et al. [89]. They examined the human

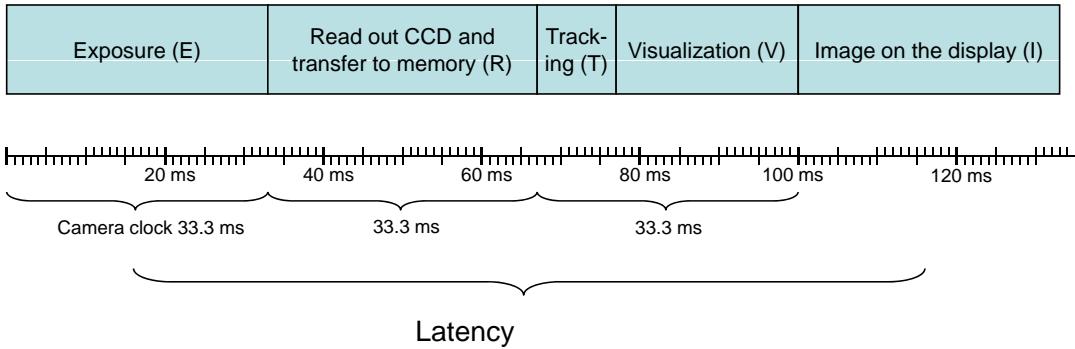


Figure 4.2: Exemplary timing of a video see-through system using a 30Hz camera.

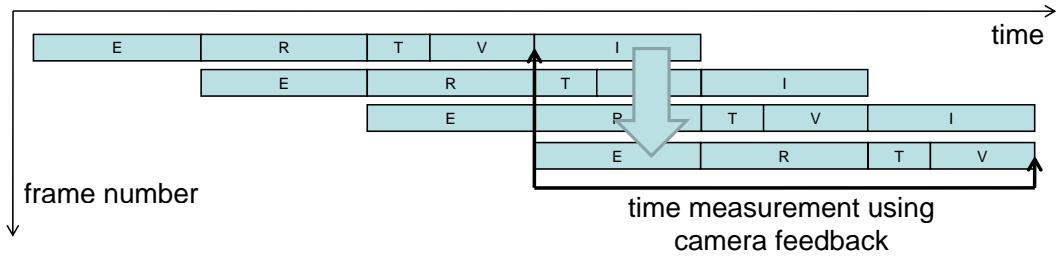


Figure 4.3: Concurrent behavior of continuous augmentation. See figure 4.2 for the explanation of abbreviating letters. Visual feedback transfers the encoded information of a former frame into the current one (big arrow). Visual feedback allows for time measurement of a full cycle (thin arrow).

performance using AR and VR systems with different frame rates. They experienced significant differences of user performance, depending on the frame rate. In our work regarding depth perception we realized significantly worse performance with regard to speed and accuracy for slow visualizations [12] (see section 4.2.4).

Ware et al. [185] show that latency has a negative effect on the performance in reaching experiments. The experiments match the theory of Fitt's law that relates the difficulty of a task to user performance speed. According to their findings, the effects of the frame rate can be attributed to latency.

In this paper, we regard the issues of *relative* latencies in video see-through systems as solved. We expect that real and virtual image parts originate from the same point of time solved by the method described in section 3.3.4.

The latency in an AR system is the temporal difference between an augmented image and instant of reality. Figure 4.2 depicts an exemplary timing of a video see-through system. Since the camera exposure requires a period of time rather than a point in time, motion blur might appear in the image. The object is interpreted to be in the middle of the blurred image at the middle of the exposure duration. Also the displayed image is perceived to temporally reside in the middle of its time slice.

The exposure time, image transfer, and the time of the image remaining on the display

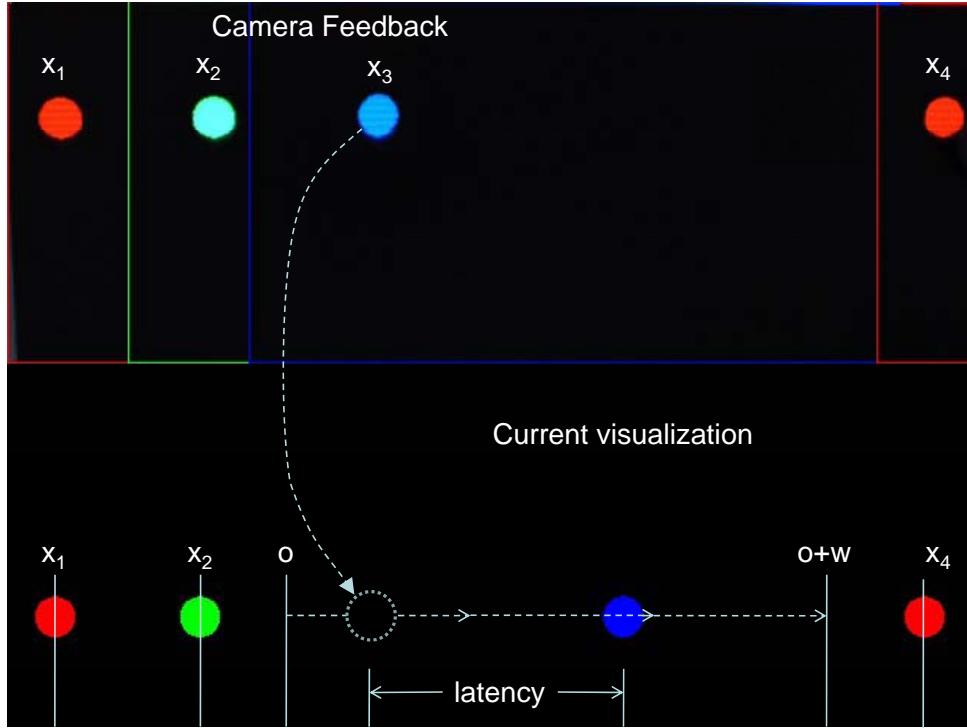


Figure 4.4: Current encoded time and feedback image.

can be obtained from the hardware. The time for tracking and visualization can be computed online. The sum of these times may yield be the correct latency, but it only constitutes a model. The sum only describes latency correctly if there are no other computations or unwanted idle times between the stages of exposure, data transfer, image generation and image display. These undesirable effects may extend latency significantly.

The complete latency can only be determined by exposing the system to the effect of latency and measure it. Jacobs et al. [74] use a pulse signal provider, a light-emitting diode (LED), a photo diode, and an oscilloscope for the measuring end-to-end latency, as they refer to it. The pulse signal triggers the LED and the oscilloscope. The photo diode is fixed on the display where the light of the LED is visualized. It is also connected to the oscilloscope. The latency can be obtained from the distance between the peaks triggered by the pulse signal and the photo diode.

We suggest a method for measuring latency using camera feedback. This method can be used in any video-based AR system without extra hardware and it provides an automatic tool for quantitative latency measurement.

An objective measurement of latency is crucial for evaluation of AR systems because latency influences users' performance significantly. Engineers of AR systems may use this measuring method in order to find out if the system matches their desired performance.

4.1.1.2 Method

In this section, we describe how to measure latency via visual feedback. This method assumes that there is no relative latency in the system. Therefore, camera latency is

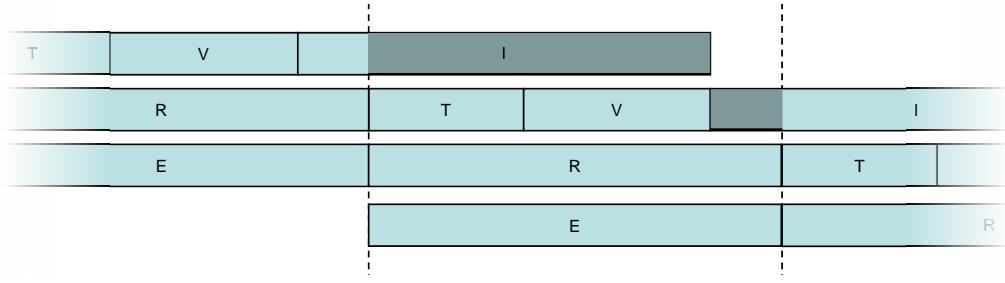


Figure 4.5: If the computation of tracking and visualization does not consume exactly a multiple of a camera cycle the camera is exposed to two frames in a single exposure.

interpreted as the total system latency.

Figure 4.2 depicts the timing of a video see-through system (as suggested by Sauer et al. [143] or any system that tracks using the image, such as systems using ARToolkit [78] and etc.). For the figure it is assumed that the tracking is triggered by the arrival of a new frame and the visualization directly follows the tracking.

Figure 4.3 shows the effect of camera feedback to such a system. The information of the visualization that is currently on the screen (I) is transferred to the currently recorded frame (E). The frames in between are not updated by feedback, because they are already in the process chain. Hence, the visual feedback can reveal the latency that users experience.

Figures 4.5, 4.6(a), and 4.6(b) depict what happens if the augmentation of the scene (V) does not end exactly when the exposure (E) begins: Two frames are visible in a single exposure. This is considered to be the common case. If the visualization has a higher update rate than the cameras a higher number of feedbacks may occur (see figure 4.6(b)).

4.1.1.2.1 Time encoding

Time is encoded into the image as four collinear circles where one circle changes its position according to the time with the function:

$$x_3 = (t \cdot w + o) \cdot X \quad (4.1)$$

with

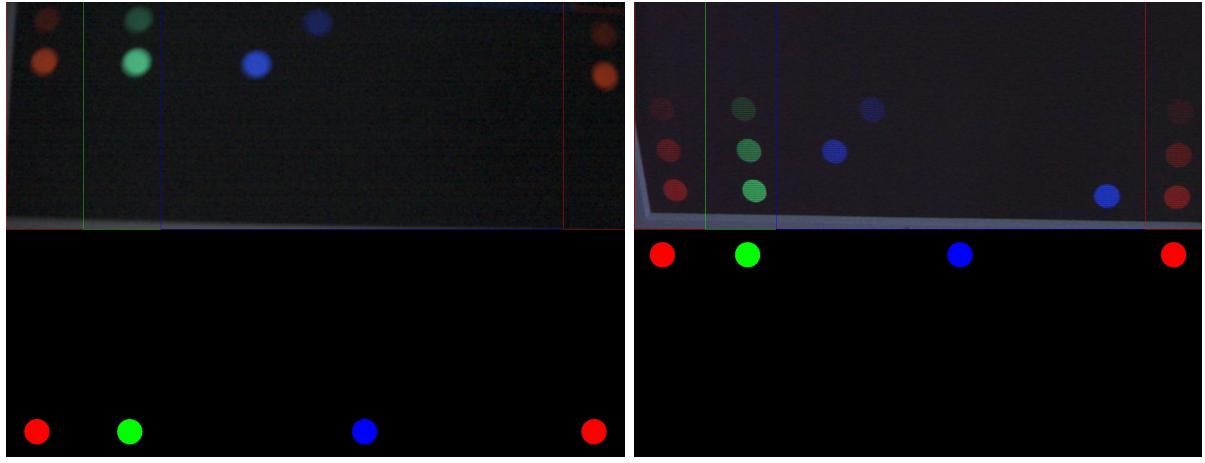
$$t := (\text{time} \bmod T)/T \in [0; 1[\quad (4.2)$$

where *time* is the absolute time, *T* is the interval time and *X* is the image resolution in x direction, and *x* is the position of the circle in the image (see figure 4.4). *w* and *o* are width and offset values to set the window in which the circle is supposed to move. The *x*-values *x*₁, *x*₂, and *x*₄ of the three other circles remain constant.

The *y*-value of each circle is calculated by

$$y_t = (n \bmod N)/Y \quad (4.3)$$

where *n* is the frame number, *N* – 1 is the expected latency measured in frames. The modulation in *y*-direction is necessary in order to avoid feedback of different frames on the same spot in the image (see figures 4.5 - 4.6(b)).



(a) Two frames visible during a single exposure. (b) Three frames visible during a single exposure of an interlaced camera.

Figure 4.6: Multiple feedback

4.1.1.2.2 Time decoding

For decoding the time from the camera feedback, the cross ratio λ of the four centroids (A, B, C, and D) of the circles is computed by

$$\lambda_{dec} = \frac{CA}{CB} : \frac{DA}{DB} \quad (4.4)$$

where AB denotes the signed distance ($AB = -BA$) from point A to B . Since the cross ratio is not affected by projective transformations, the cross ratio in the feedback image theoretically has the same value as in the encoding step:

$$\lambda_{enc} = \frac{(x_1 - x_3)(x_2 - x_4)}{(x_2 - x_3)(x_1 - x_4)} \quad (4.5)$$

Simple transformation leads to

$$m := \frac{(x_1 - x_4) \cdot \lambda_{dec}}{(x_2 - x_4)} = \frac{(x_1 - x_3)}{(x_2 - x_3)} \quad (4.6)$$

where m is given by x_1 , x_2 , x_4 , and λ_{dec} .

From equation 4.5 and 4.6 we obtain

$$x_3 = \frac{mx_2 - x_1}{1 - m} \quad (4.7)$$

with $m \neq 1$ if none of the points coincide.

With x_3 inserted into equation 4.1 we can easily compute

$$time_{image} \bmod T = \frac{x_3 - o}{w} XT \quad (4.8)$$

The latency l can be computed by

$$l = (time_{now} - time_{image}) \bmod T \text{ if } T > l \quad (4.9)$$

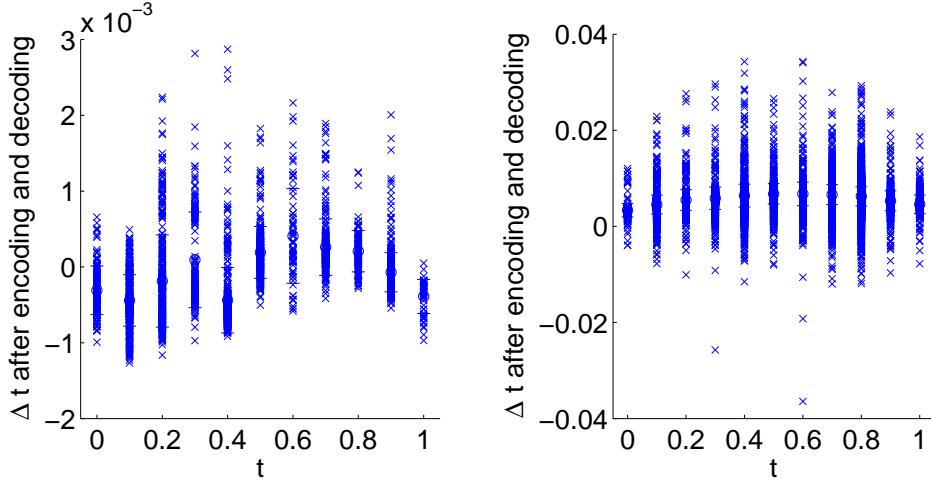


Figure 4.7: Experiment: Encoding/decoding error depending on given t . System 1 (left) and System 3 (right) have different camera resolutions

4.1.1.2.3 Computer vision

The computer vision algorithm has been chosen to extract the center of mass of circles. The center of mass of a circle is almost preserved in a projective transformation. The influence on the overall error is negligibly low if the camera view is nearly orthogonal to the display. We have used this kind of feature, because it can be easily reimplemented while providing subpixel accuracy.

As the cross ratio is maintained in projections, the camera position relative to the display is irrelevant, except for the fact that all four circles must be visible in the feedback. Any kind of distortions are not modeled in this approach. Camera distortions can be corrected with standard computer vision models. The display is assumed to be flat or at least locally flat in the range in which feedback is used.

The visualization of the circles has to be inserted after the visualization of the AR application because we intend to determine the time that elapses between the instant at which the light hits the sensor and the instant at which the augmented image is revealed.

Image revelation, which is done in OpenGL using the command `glSwapBuffers()`, works in virtually no time, since it is done by changing the pointer on the graphic card. This assumption is valid unless buffer swapping is synchronized with the vertical synchronization of the graphic card. Synchronizing buffer swaps in this way may be done in order to have only consistent frames. Otherwise, the top part of the image may originate from a different image than the bottom one, resulting in a perceivable vertical line. This effect does not visible in a software screen shot, but it is apparent in visual feedback.

4.1.1.3 Results

First we intend to validate the precision of our method since the cross ratio is not necessarily a computationally stable feature. In our experiments, we used values of

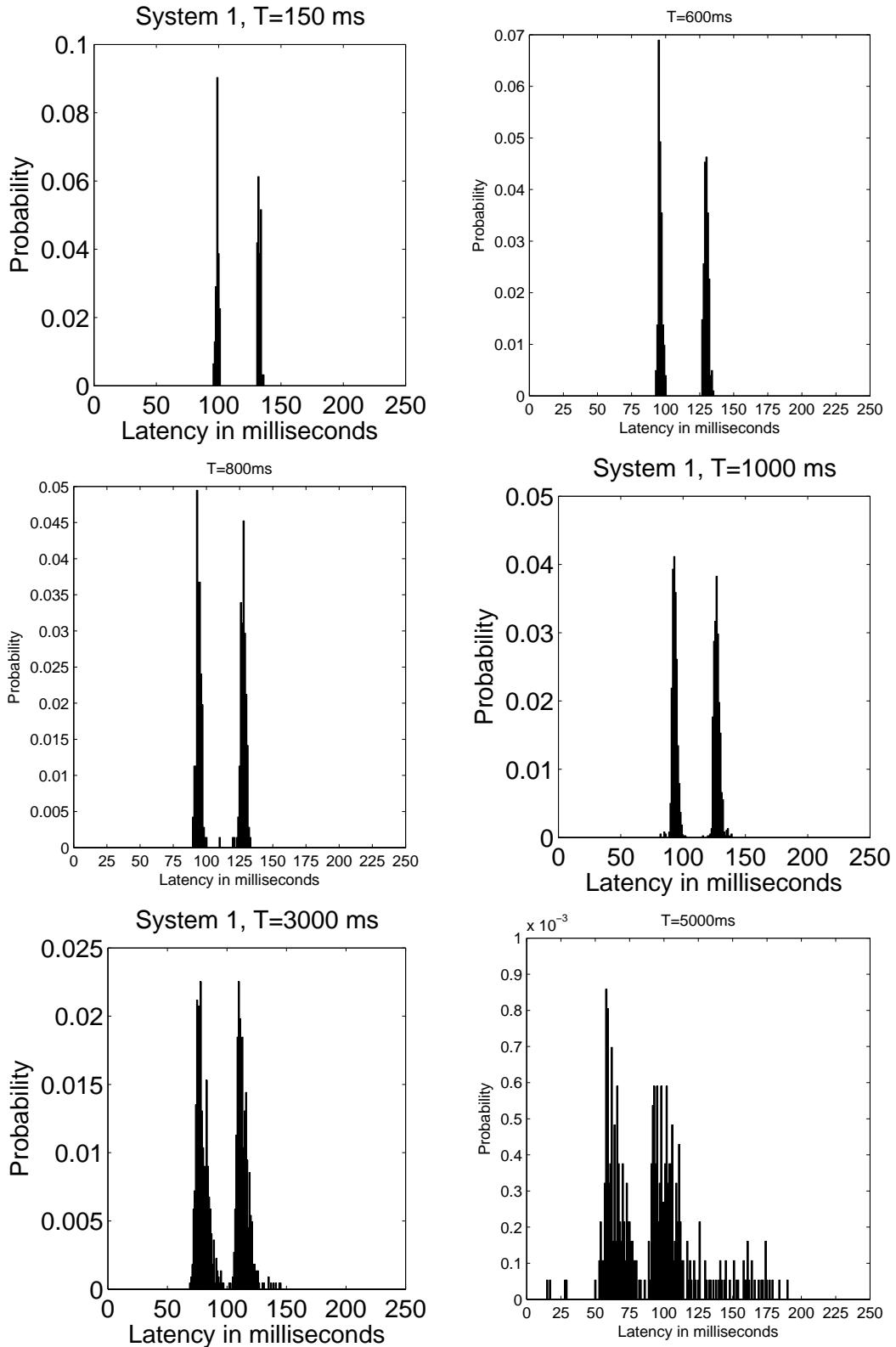


Figure 4.8: The latency graphs of system 1. Effects on precision due to varying T . System 1, $w = 0.5$

$T = 250ms$, $x_1 = 0.05X$, $x_2 = 0.2X$, $x_4 = 0.95X$ $w = 0.5$ and $o = 0.35$ for encoding unless stated otherwise. T is constrained to be larger than the expected latency because of equation 4.9. In this combination, using a VGA camera, a movement of one pixel represents $\frac{wX}{T} = 0.78125$ milliseconds.

For the actual precision validation, we set up two kinds of experiments. In the first experiment, we encode static values of t in the image and decode them from a feedback image in order to compare them with those in t . The measurements have a standard deviation of $2.12 \cdot 10^{-3}$ using the interlaced analog camera of system 1 and $3.94 \cdot 10^{-4}$ using a 1024×768 pixel digital camera (progressive scan) as employed in system 3 (see figure 4.7). If we use this scheme to encode values in a range of $T = 250ms$ we expect deviations of $0.5ms$ in system 1, which has the poorest imaging quality of the four systems. System 3 offers the best image quality. The second experiment (see figure 4.8) shows the impact on precision when varying the range T . The figure depicts the latency measurements of system 1 as a relative histogram. While the histogram does not change significantly for $T < 800ms$, errors are obvious for $T = 5000ms$.

The four different hardware setups of NARVIS were examined using this method (for a detailed system description see section 3.4.2). The visualization is kept as simple as possible in order to keep its contribution to the latency low. Only three registered arrows are drawn in order to show a coordinate system to make sure that tracking is performing correctly.

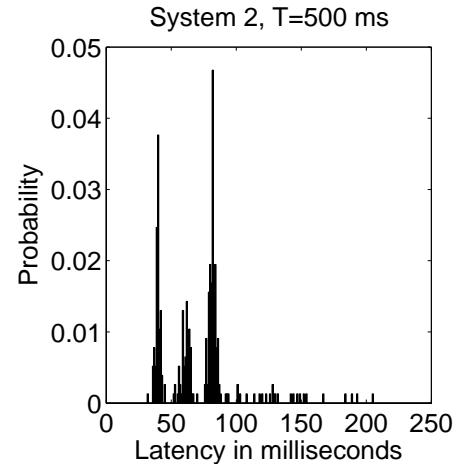


Figure 4.9: The latency graph of System 2. No synchronized tracking in the experiment.

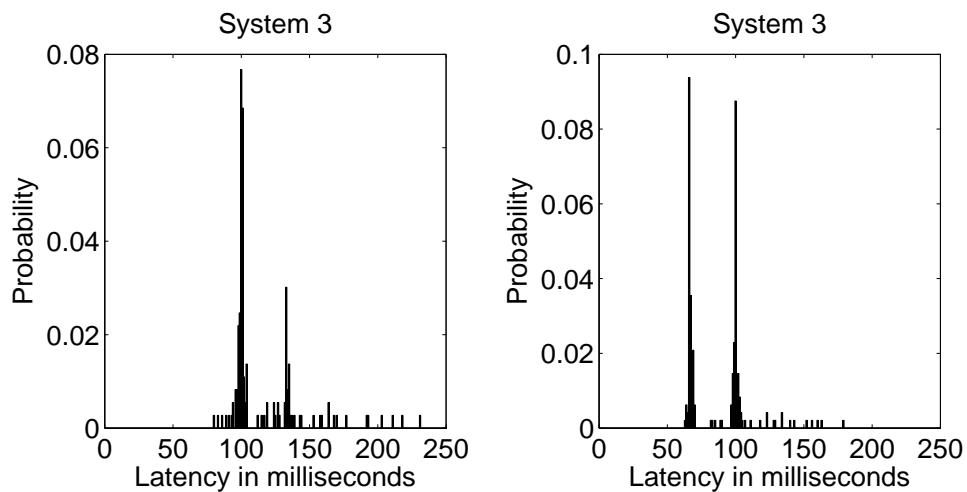


Figure 4.10: The latency graph of system 3. With synchronized tracking system(left) and without (right).

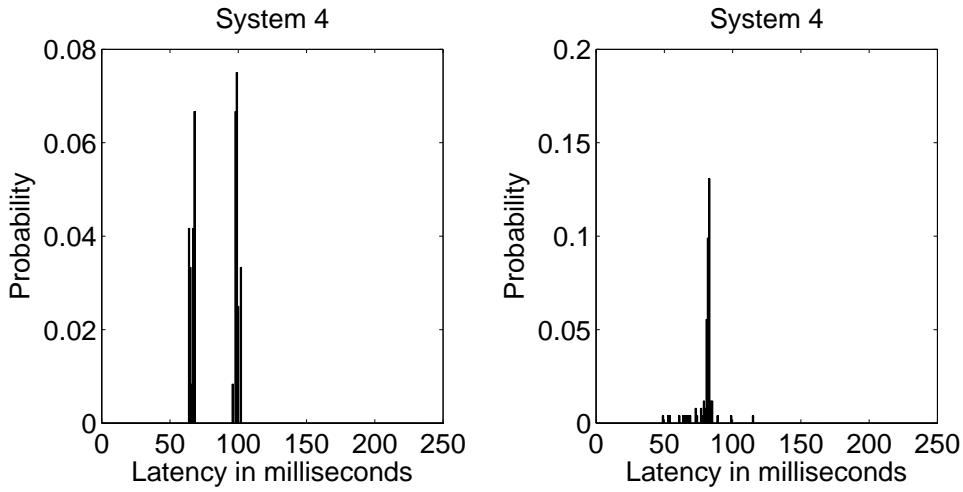


Figure 4.11: The latency graph of system 4. One experiment with a camera setting of $30Hz$ (left) and another with $200Hz$ (right). While the left figure shows expected values the right figure reveals suboptimal performance of the system.

4.1.1.4 Discussion

We interpret our experiments in as follows:

- The precision of $1ms$ appears to be sufficient as it is 1% of the latency of current state-of-the-art systems. It is also adequate for engineering purposes such as optimizing a system. Our experiments show that our method has a precision of at least $1ms$ using the poorest camera quality of our four systems.
- The choice of T has a direct influence on the accuracy of the measurement. It should be larger than the latency in order to avoid that the algorithm interprets a large latency as small due to the module function. A value of $500ms$ was found to be a reasonable compromise.
- Two different frames may be seen in a single exposure (see 4.5). The measurement method accounts for that by measuring encoded time of both frames. This results in the two peaks visible in the latency graphs. Effects of synchronization with other hardware can be studied easily with this method (see figure 4.10).
- The figures 4.9–4.11 show that latency is not necessarily constant in a system. It depends on different concurrent hardware devices, for instance the camera and the graphic card output. The visualization as a latency graph helps to detect the magnitude of outliers easily.
- Systems that do not operate optimally can be detected without additional hardware (see 4.11). Especially system 4 appears fast when operating. However, the $200Hz$ cameras promise a better latency than $80ms$.

4.1.1.5 Conclusion

We have presented a method to estimate the latency of a video see-through AR system. It does not require additional hardware for its measurements, and is able to automatically determine end-to-end latency. Precision is expected to be one millisecond or less, which is sufficient for the evaluation of an AR system.

The method has been used on four different AR systems. The measurement matches our theoretical considerations about latency of each system.

We recommend visualizing the latency of a system as a histogram, which is much more expressive in illustrating system performance than the mean value and the standard deviation.

For research and engineering, we consider latency measurement by camera feedback a valuable tool for validating system performance. Latency is a crucial factor for an AR system with regard to users' performance. We hope that the presented method can be a reference method for engineering a system with the maximum performance in terms of latency.

This form of latency measurement has to be integrated into the AR system. It cannot be delivered as a precompiled executable and generating a library would be a difficult task. Therefore, we sought for a method that can be easily reimplemented by other groups. A more accurate way of encoding the time into the image would be encoding the time as literals instead of a translation. A barcode could for instance, serve as a representation. The current encoding has the advantage that it may be evaluated visually without programming a computer vision algorithm.

We would like to point out that the method is not limited to video-based systems. Further investigations are planned to evaluate use of a camera for latency measurement between real and virtual for optical see-through technology.

4.1.2 Error Propagation

Note: The following work has been realized in collaboration with Martin Bauer. It already been published in [11].

Intraoperative guidance systems (IGS) require pose estimation for the patient and instruments. Most commercially available systems use optical tracking technology. The tracking system consists of n -ocular camera systems using linear CCD cameras or matrix cameras. The cameras detect the pose of the instruments by means of a rigidly attached set of fiducials. These fiducials are either light-emitting diodes or retroreflective objects reflecting light of flash lights that are attached to the tracking cameras. In this paper, we refer to a set of fiducials as *tracking target*.

The accuracy of such a system depends on a number of variables that interact in a non-obvious manner. The accuracy of the tracking system strongly depends on the geometry of the system. Furthermore, measurement technology, e.g. the cameras, computer vision and optimization algorithms, plays an important role, as do light conditions and physical properties of fiducials. The geometry includes the number of visible fiducials, the spatial arrangement of these fiducials, and the pose of the tracking cameras.

We model the tracking system error as follows:

IPE The image plane error (IPE) is the measurement error on the camera sensor. In our previous work we have shown that it is valid to assume that a variety of systematic and random errors can be modeled by a Gaussian error distribution in the image plane.

FLE The fiducial location error (FLE) represents the three-dimensional error of a single fiducial. It depends on the IPE and the geometrical setup of the cameras.

MTE The mean target error (MTE) represents the spatial error of a rigid tracking target in the centroid of the fiducials. It comprises rotational and translational parts. It depends on the FLE of each fiducial and the spatial arrangement of the tracking target.

TRE The target registration error (TRE) represents the actual error of a point in the tracking target coordinate system. The point of interest is generally not the centroid of fiducials, but a point on the attached object. The error is amplified due to the rotational part of the MTE.

The error of a system can be described as the root-mean-square (RMS) error, which is the standard deviation of a measurement. By definition, the RMS cannot provide the required information if there is one direction with a higher variance than the others. Therefore, it implicitly models isotropic and independent errors. If the error is expected to be anisotropic, it can be described by the covariance matrix of the distribution. This method of describing the error implicitly models the error with a zero-mean Gaussian distribution.

In current systems, the accuracy of a tracking system is specified statistically by moving fiducials through the specified measurement volume and computing the root-mean-square error [194]. This error is propagated to the point of interest according to the spatial marker arrangement. The point of interest may for instance be the tip of an instrument. Fitzpatrick et al. [51] describe a simple formula to predict the target registration error (TRE) from the fiducial location errors (FLE), given as a one-dimensional RMS. The RMS computation is provided by the tracking system manufacturer.

However, it is known that due to the underlying geometrical situation, the tracking errors have non-uniform error distributions throughout their measurement volumes [51, 83]. According to Khadem et al. [83] the errors of different commercial systems show a significant anisotropy.

Hoff et al. [72] model the TRE with covariances. They measure the root-mean-square deviation of the FLE and propagate it. In previous work, our group has proposed a model for the estimation of the fiducial location error from the camera geometry [9]. The FLE is modeled using covariances, since the error is anisotropic (see figure 4.13).

The presented work is not limited to multiple camera systems. It can be applied to single camera systems as well. The methodology makes particular sense in sensor networks as used for the NARVIS setup.

4.1.2.1 Methods

In our approach, the error and its propagation is modeled as a Gaussian distribution. The error is defined by its expected value and its covariance. The error propagation follows the formula $cov(\mathbf{Ax}) = \mathbf{A}cov(\mathbf{x})\mathbf{A}^T$ for affine transformations \mathbf{A} . The propagation of the expected value is straightforward. Thus, we focus on propagating the covariances. In case of a non-linear propagation of the error, we can use the first order approximation $cov(f(\mathbf{x})) = \mathbf{J}_f cov(\mathbf{x}) \mathbf{J}_f^T$ with \mathbf{J}_f as the Jacobian of f at \mathbf{x} , if the function is locally linear around \mathbf{x} .

The following steps have to be performed:

- The estimation of IPE (described in section 4.1.2.2.1). We create a two-dimensional covariance matrix for each camera.
- The propagation from IPE to FLE (described in section 4.1.2.2.2). We create a three-dimensional covariance matrix for each fiducial.
- The propagation from FLE to MTE (described in section 4.1.2.1.1). We create a six-dimensional covariance matrix for each target.
- The propagation from MTE to TRE is computed with the error propagation formula for affine transformations. We create a 3D covariance matrix for the point of interest.

4.1.2.1.1 From fiducial location error to tracking target error

We assume that all coordinates and covariances are given in target coordinates, i.e. $[R|t] = I$. Without loss of generality, the origin is at the centroid of the fiducial positions. This function maps the six-dimensional marker target error $[\Delta_R|\Delta_t]$ to the three-dimensional fiducial location error Δ_{t_c} at the point $q = (q_x, q_y, q_z)$.

$$\Delta_{t_c} = \Delta_R q + \Delta_t \quad (4.10)$$

We linearize this function around the zero-mean error Δ_t :

$$J_f(q) = \frac{\delta}{\delta(\Delta_t)} \Delta_{t_c} \Big|_{\Delta_t=0} = \begin{bmatrix} 1 & 0 & 0 & 0 & q_z & -q_y \\ 0 & 1 & 0 & -q_z & 0 & q_x \\ 0 & 0 & 1 & q_y & -q_x & 0 \end{bmatrix} \quad (4.11)$$

We can now stack the equations for all fiducials together in a single matrix

$$M = [J_f(q_1), \dots, J_f(q_n)]^T \quad (4.12)$$

and apply the backward propagation formula:

$$\mathbf{J}_{f^{-1}} cov(\mathbf{x}) \mathbf{J}_{f^{-1}}^T \quad (4.13)$$

which can be calculated more easily, as described by Hartley et al. [130]:

$$cov(f^{-1}(\mathbf{x})) = (\mathbf{J}_f^T cov(\mathbf{x})^{-1} \mathbf{J}_f)^{-1} \quad (4.14)$$

$$\Sigma_{\vec{c}} = \left(M^T \begin{bmatrix} \Sigma_{p_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \Sigma_{p_n} \end{bmatrix}^{-1} M \right)^{-1} = M^+ \begin{bmatrix} \Sigma_{p_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \Sigma_{p_n} \end{bmatrix} (M^T)^+ \quad (4.15)$$

where Σ_{p_i} is the covariance of the i -th fiducial in target coordinates and Σ_c represents the covariance matrix of the MTE at the centroid.

4.1.2.1.2 Visualization setup

The estimated errors are visualized in the video and in the following images in an augmented reality system using a second, independent tracking system. We visualize only the tracking error. The overall application error could be estimated by additional error propagation using our model, but this exceeds the scope of this work. Addition of errors and further propagation is well-described in the work of Hoff et al. [72].

In the images, the tracking error is visualized as the 95% confidence level of the estimated error amplified by 33.

4.1.2.2 Results

The error estimation in this paper is an extension of the estimation method introduced by Fitzpatrick et al. [51] and Hoff et al. [72]. Using the same assumptions as they did, our model mathematically provides the same results. However, these assumptions are too strict according to our computations, because the FLE has a non-uniform and anisotropic error.

Figure 4.12 shows the behavior of the different error propagations. This figure depicts the expected error of a four-camera setup in which the fiducial error is nearly isotropic. Therefore, the middle and the right image show almost the same expected error. The difference between the corresponding methods becomes obvious when one or more fiducials or cameras are occluded (see section 4.1.2.2.5 and 4.1.2.2.4).

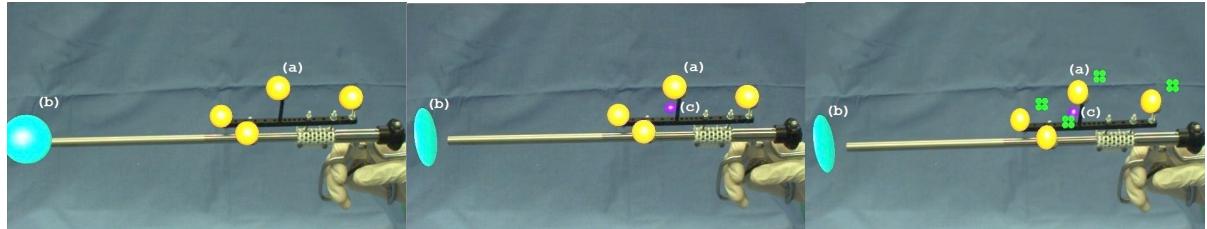


Figure 4.12: Estimation of RMS at point of interest using target RMS [51] (left), estimation of covariance using fiducial RMS [72] (middle), estimation of covariance using fiducial covariance (right) - covariance of fiducial (a), covariance at point of interest (b), covariance at center of fiducials (c)

4.1.2.2.1 Measurement of the image error

The error of an optical tracking system may have different sources: Internal calibration of cameras, imperfect lenses, accuracy of computer vision algorithms, and image blur. All of these can be modeled as a 2D error in the plane of the camera sensor.

In an approximation, we regard the error to be constant on the sensor of the camera for a certain size of the fiducial. For the estimation of errors in the sensor plane, we used covariance measurements of fiducials in the tracking space.

We calculated the noise in the sensor plane of each camera by finding the inverse propagation described in section 4.1.2.2.2 using numerical optimization, which is the Levenberg-Marquardt algorithm. The covariance has been constrained to be isotropic and uniform for a numerically stable solution. The measured standard deviation in our setup is approximately $\frac{1}{52}$ pixels for each fiducial.

4.1.2.2.2 From image error to fiducial error

As in the previous work of our group [9] we model this transformation using the geometry of the cameras.

We use an n -ocular system with pinhole cameras with intrinsic parameters \mathbf{K}_i and extrinsic parameters \mathbf{T}_i of the i -th camera detecting the same point \vec{x} at the location \vec{u} . We obtain the measurement function for the triangulation, a set of nonlinear camera equations p as the projection function:

$$\begin{aligned} p : \quad \vec{u}_1 &= \frac{1}{\rho_1} \mathbf{K}_1 \mathbf{T}_1 \vec{x} \\ &\vdots \\ \vec{u}_n &= \frac{1}{\rho_n} \mathbf{K}_n \mathbf{T}_n \vec{x} \end{aligned} \quad \text{using } \rho \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \mathbf{T} \mathbf{x} \quad (4.16)$$

ρ_i denotes the normalization factor needed for homogeneous coordinates. We use linear models throughout the error propagation, so we can safely assume a pinhole camera model here.

In order to compute the FLE, we build the Jacobian $\mathbf{J}_p = \frac{\delta p}{\delta \vec{x}}$ and apply the backward propagation formula.

$$\Sigma_x = \left(\mathbf{J}_p^T \begin{bmatrix} \Sigma_{u_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \Sigma_{u_n} \end{bmatrix}^{-1} \mathbf{J}_p \right)^+ \quad (4.17)$$

The resulting equations are analytically computed using a computer algebra system and exported to C code.

4.1.2.2.3 Inclusion of visibility data

Optical tracking systems have the inherent property that cameras need to see the fiducials to be able to compute a pose estimation. If the line of sight is blocked between a camera and a target the accuracy of measurement degrades. Figure 4.14 depicts the RMS in a plane of the same four-camera setup with one camera occluded in each image. The depicted plane is parallel to the plane that includes the four cameras at a distance of one meter. The error has been computed according to the proposed model in section 4.1.2.2.2.

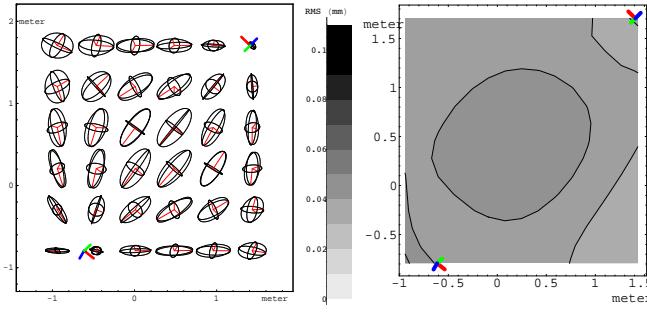


Figure 4.13: Predicted FLE accuracy of a two-camera setup. Covariance ellipsoids (left) and RMS error (right). RMS is close to constant while covariances reveal significantly changing directions of estimated errors. Coordinate systems depict camera positions.

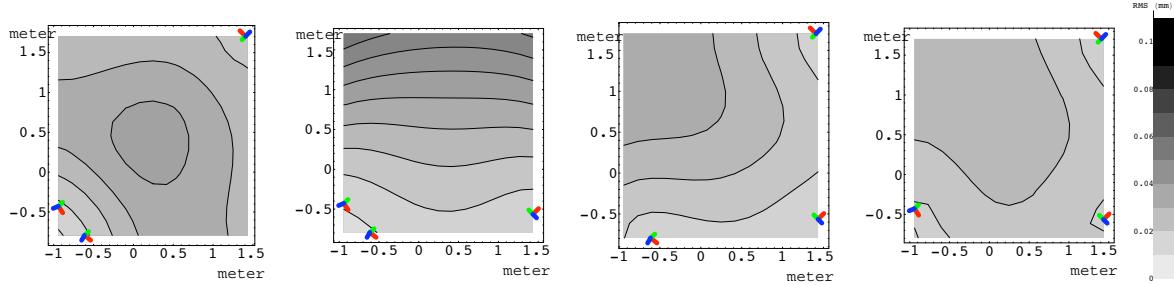


Figure 4.14: Occlusion of cameras changes the expected fiducial location error significantly: Coordinate systems depict camera positions. The gray level indicates the estimated root mean square error of a fiducial in a plane of 1m distance. Asymmetry is due to asymmetric camera orientation

The images clearly show that assuming a constant RMS of fiducials in the volume is not sufficient. It also shows that the visibility plays an important role with regard to the accuracy of each fiducial.

4.1.2.2.4 Occlusion of Single Fiducials

Current navigation systems request full visibility of each fiducial in each camera for the maintenance of the required accuracy. Some tracking systems already provide error estimates based on the work of Fitzpatrick et al. [51]. Using our proposed method, we can now provide more accurate error estimates. We can show that in typical configurations, the expected error will actually be larger than expected, while the average error estimate over the whole tracking space still stays the same. Figure 4.16 shows the effect on the estimation of the instrument tip when occluding one fiducial.

4.1.2.2.5 Occlusion of Cameras

Camera occlusion is a problem that occurs only in multi-camera setups. In current tracking systems in surgical navigation, the pose estimation is stopped in case of camera occlusion. The occlusion of cameras has a significant influence on tracking accuracy. (see figure 4.14 and 4.16).

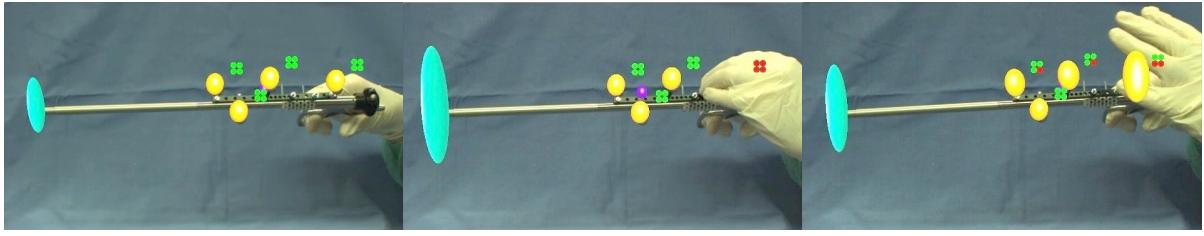


Figure 4.15: Effects of occlusion: No occlusion (left), fiducial occlusion of four cameras (middle), several fiducials occluded (right); each circle depicts the visibility status of the fiducial in each of the four cameras.

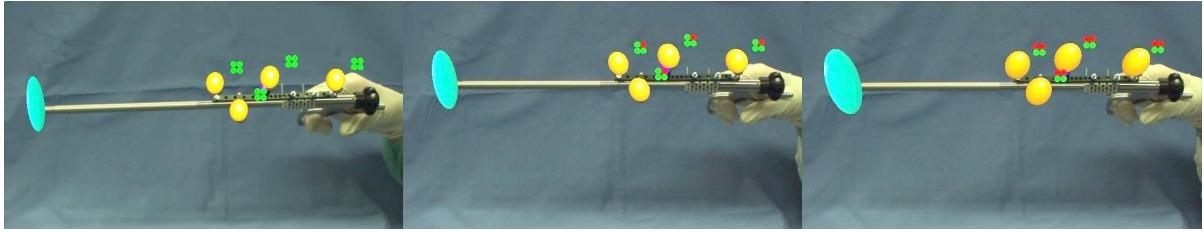


Figure 4.16: Effects of camera occlusion: no camera occluded (left), one camera occluded (middle), two cameras occluded (right)

Tracking systems in commercial medical applications need to be accuracy-certified. In order to achieve this certification, the manufacturer has to guarantee a certain level of accuracy. Currently, this is done by mounting the cameras in a mechanically rigid configuration and then calibrating and measuring the error in a predefined working volume. For this reason, up to now all commercially used tracking systems use a configuration with two matrix cameras or three linear cameras. Using multi-camera setups would provide redundant cameras to avoid occlusions.

Unfortunately, guaranteeing accuracy for flexible setups is a complex task even without taking the visibility into account.

The online error estimation presented in this work provides an important step towards accuracy-certified multi-camera setups. If we can guarantee a certain accuracy on the image plane this accuracy can be propagated to provide accuracy estimates at a point of interest. By providing many distributed tracking cameras in the operating room, the proposed method would allow tracking and navigation even if one or more camera views are occluded. The navigation could continue if the system can guarantee the required level of accuracy.

4.1.2.3 Conclusion

We have proposed a novel way of estimating the error in optical tracking systems. We model the FLE and the TRE as anisotropic non-uniform errors. We deduce them from the error in the sensor plane of the tracking camera. Errors are computed online based on the visibility of each fiducial. The effects can be appreciated best in the supplementary material.

The interactive propagation of errors in real time based on the geometry, facilitates a multitude of applications. Engineers designing tracking systems [35] have the opportunity to get direct feedback with regard to the final accuracy. Physicians may benefit from the online information in intraoperative visualization. The accuracy can be visualized in the navigation rather, as done in current approaches, visualizing any pose information in case accuracy sinks below a certain threshold.

Online estimation of errors based on visibility of fiducials allows for creating a multiple camera tracking setup for surgical navigation. In order to guarantee a certain accuracy, which is necessary for navigation systems, we compute the expected error based on the visibility. The proposed method enables online estimation of the final accuracy while one or more fiducials or camera views are occluded. This could allow computer-aided solutions to gain considerable flexibility. They could in fact continue to function even when occlusions occur, as long as such occlusions do not decrease the level of accuracy below the acceptable limits.

4.1.3 End-to-end accuracy

Technical validation of the system has been performed by NARVIS project partner SCR. The tests have been performed on system 1 (see section 3.4.2) using the calibration method described in section 3.4.1. In these experiments, the new software framework is not used yet. However it should not effect accuracy. The measurement precision of the tracking system has a submilimetric standard deviation [179].

In order to measure end-to-end accuracy, an experiment has been prepared that is impossible to perform without guidance in order to prevent benevolent support of the subject. Accuracy of the actual result is measured by a independent measurement system. This method takes into account calibration errors, measurement noise, perceptual errors, and execution errors by subjects.

Transparent plastic rings were targeted in a transparent jelly. The plastic rings are only distinguishable from the jelly in the imaging device. The subject had the task to move a needle tip to the center of the ring, which did not allow for any haptic feedback. Using the needle, a piece of metal was dropped to mark the reached destination point. This allowed comparison by means of an MRI scanner. The center of mass of segmented voxels are compared. The overall placement error, which also includes the jitter of the subject's hand is below 3mm [25], [181]. These findings have since been affirmed in further animal experiments. A second similar experiment was set up in using an opaque phantom and CT as the imaging hardware. An overall placement error of 3.5mm could be achieved [34].

4.2 Usability

The evaluation of usability is the final step in creating a new human-computer interface like medical augmented reality. Knowing limitations of the new interface and differences to classical interfaces help to improve the technology and find useful applications.

4.2.1 Clinical integrability

The system was presented to six surgeons in order to obtain a realistic insight of possibilities and limitations of the technology in an initial cadaver experiment. This experiment was set up by Jörg Traub. CT slices and volume rendering of a CT were displayed for pure visualization. The system has been well-received among surgeons, irrespective of whether they were project members or not. In principle, the system could fit to the environment of a surgery room. For sterility reasons, it is recommendable to attach the head-mounted display to a boom similar to that for the surgical illumination. Otherwise, a member of the surgical staff without sterile hands needs to mount and remove the head-mounted display during the surgery.

One of the results was the decision to realize a cadaver experiment for a navigated intramedullary nail (IM-nail) locking procedure. It is a frequently performed and image-guided orthopedic procedure where the result can easily be determined to be a success or not. Further descriptions will be given in the next section.

4.2.2 Navigation: 6D Guidance

The experiments described in section 4.1.3 do not only demonstrate the accuracy of the system, but also show that the system and its visualizations are capable navigating a surgeon to an arbitrary point in space with sufficient accuracy. For the spine surgery that is the chief focus of in the NARVIS project it not sufficient to navigate to a point. Placement of the pedicle screw is a more complicated task with six degrees of freedom. Since that laparoscopic instrument is rigid, it is not possible to navigate first to the entrance point and then in the desired direction. The direction has to be incorporated from the beginning of the guidance. This kind of guidance is required in the surgery tackled in the NARVIS project.

The second cadaver experiment was also set up as well by Jörg Traub. The intramedullary nail (IM nail) is used to align and stabilize fractures of long bones. It is inserted into the marrow canal of bones of the extremities. The nail is tightened to the bone with two screws that are driven through the bone and through a hole in the nail. The experiment is an example for 6D navigation, since the entry point has to be passed with the correct direction of motion. The IM nail locking has been chosen for the experiment, despite the target region differing from the spine. However, both are image-guided orthopedic interventions that require similar accuracy. Success can be verified easily by studying whether the screws traverse the hole.

Four IM nail screws were placed using a navigated drill and screw driver. In each of the cases the surgeon were confident that he traversed the hole in the nail. The devastating result was that all of the screws were in fact ill-positioned (see figure 4.17). Accuracy was validated by back projection of an additional known fiducials during the experiment. The failure could therefore not be attributed to accuracy. The result led us to the conclusion that the perceived position might differ from the position in the system. Experiments on depth perception followed, including a suggestion for visualization of six-dimensional guidance (see section 2.3.3.2).

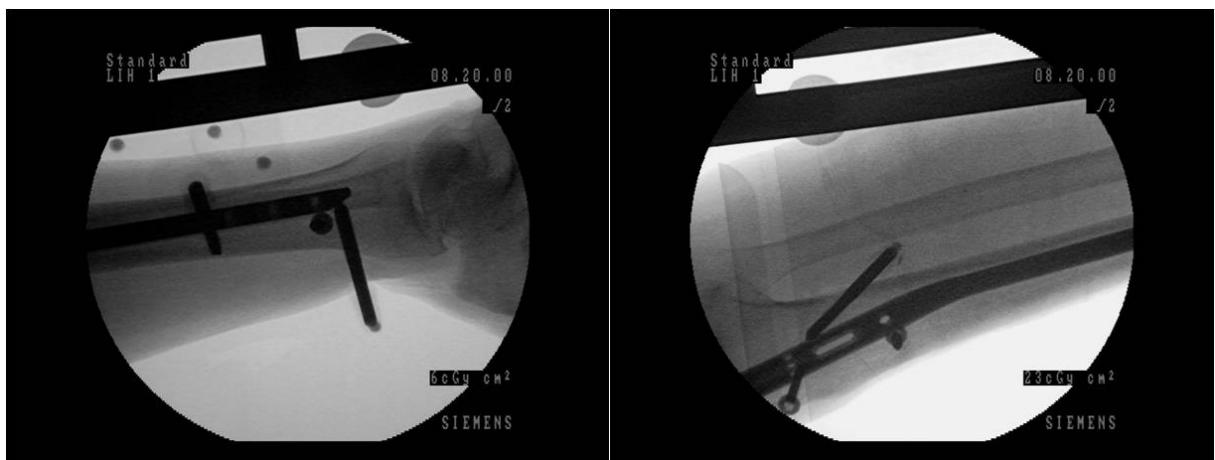
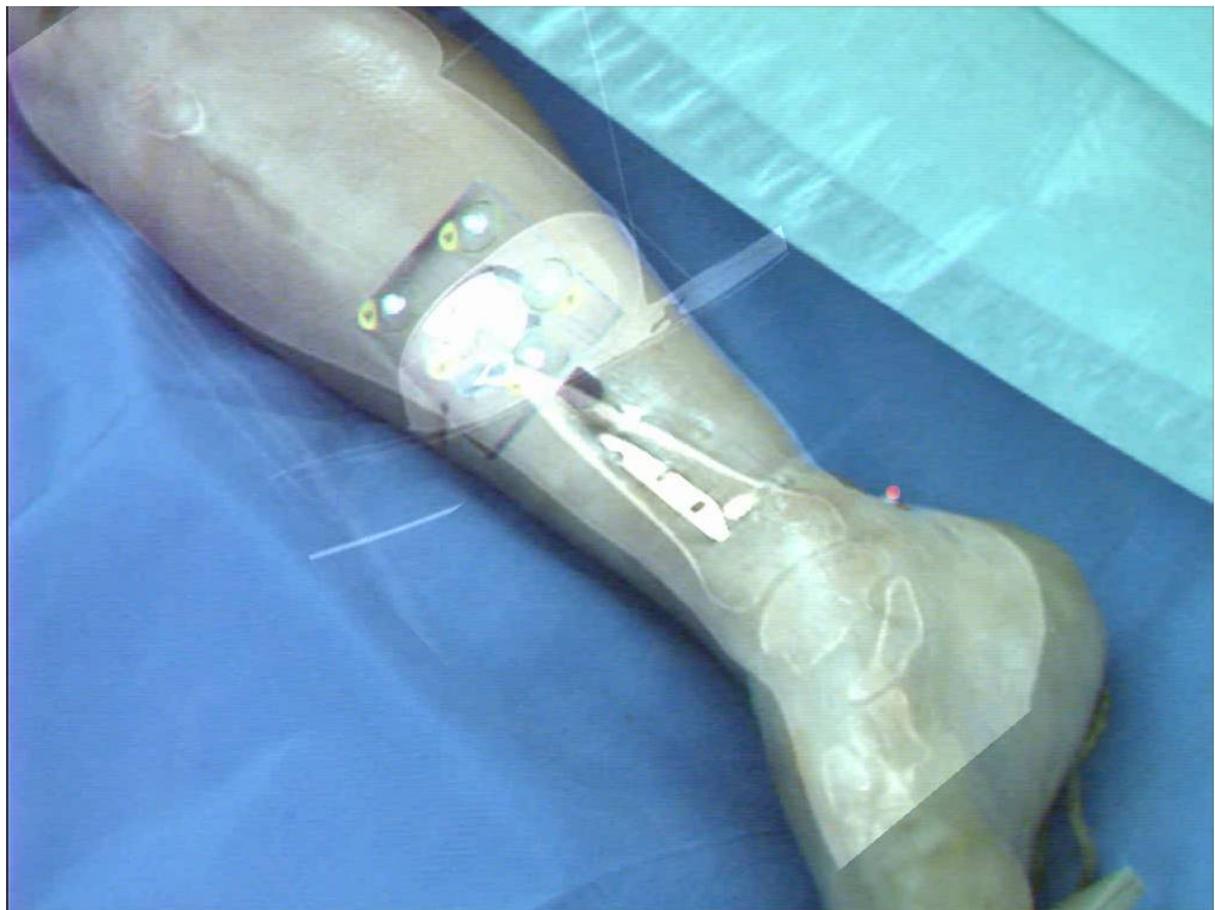


Figure 4.17: Cadaver experiment: Augmented slice rendering (top), Result: Screws miss the holes of the IM nails (bottom) - Courtesy of Jörg Traub

4.2.3 Navigation: Classic interface versus AR interface

The experiments of Wacker et al. [181] show that three-dimensional positioning of a biopsy needle can be improved in comparison to the conventional approach by means of a video see-through AR system. The authors report that the resulting accuracy was comparable to reported robotic approaches, while performance was significantly faster than that of the robotic approaches.

This leads directly to the question whether AR navigation to a certain point in space could be done as effectively with a classical navigation system.

A drill experiment was set up in order to compare different kinds of visualization for navigation. All experiments used the same input data of the tracking system and registered CT data. All visualizations were displayed using the same AR system. One visualization provided the three orthogonal views on a virtual screen where the center of each is defined by the drill position. The other provided registered in-situ visualization of the data as a slice rendering through the drill axis and as a volume rendering.

The results suggest that the orthogonal slices of the classical navigation perform at least as good as the other visualizations in terms of perceived accuracy. However, the surgeons performed faster with in-situ visualizations, which we interpret as being due to an improved overview on the scene [22].

In a second series of drill experiments, the virtual classic navigation visualization in the AR system was replaced [24] by a real monitor displaying a classical navigation interface. This experiment was intended to find out if the head-mounted display reduces the performance of the classic navigation interface of the first experiment. The results show no significant difference.

4.2.4 Depth Perception

The IM nail locking experiment showed an insufficient end-to-end accuracy for the six-dimensional navigation task, but this could not be attributed to system accuracy. This led to the assumption that the visualization and its perception might be the reason for the errors.

Other groups reported on the effects of wrong depth perception due to inappropriate visualization [6], [76]. More information about depth perception in AR systems is given in section 2.3.1.

Note: The following work has already been realized together with Christoph Bichlmeier. It has been published in [12]. The paper describes and evaluates well-known as well as novel visualization techniques that are designed to overcome misleading depth perception of superimposed virtual images on the real view.

We invited 20 surgeons to evaluate seven different visualization techniques using a head-mounted display (HMD). The evaluation has been divided into two parts. In the first part, the depth perception of each visualization type is evaluated quantitatively. In the second part, the visualization types are evaluated qualitatively using a questionnaire pertaining user friendliness and intuitiveness. This evaluation with a relevant number of surgeons using a state-of-the-art system is meant to guide future research and development on medical augmented reality.

4.2.4.1 Methods

We have compared seven different visualization modes relative to each other with the same hardware setup. In the following paragraphs, we will describe our hardware setup, the evaluation procedure and the visualization modes that we have evaluated. The technology in use is NARVIS system 1 (see section 3.4.2). Video see-through technology offers a broader range of visualization possibilities for our experiments than optical see-through, since virtual objects may be displayed opaque. An off-the-shelf computer was used to render 3D graphics, compute and include tracking data, and to synchronize and combine imagery of virtual and real entities. The computer used was an Intel Xeon(TM), CPU 3,20 GHz, 1,80 GB RAM, NVIDIA Quadro FX 3400/4400.

4.2.4.1.1 Description of the evaluation procedure

The evaluation of the visualization methods was divided into two parts. In the first part, we wanted to evaluate which visualization offered the most reliable depth perception. In order to get quantitative results, we let the participants fulfill a task on a phantom and subsequently measure the accuracy of the performance and the time required to fulfill the task.

In the second part, we surveyed user acceptance. Participants were asked to fill out a questionnaire regarding the usability of the system. This questionnaire was to be filled out directly after the first part of the evaluation. Thus, we expected the experience with the AR system to be very present and pristine. The group of participants consisted of 20 surgeons of our partner hospital.

4.2.4.1.2 Description of the tests

All participants had to fulfill the same task with different visualization techniques. The task and the visualization modes can be viewed in the video. The task consists of a pointer that has to be moved to a spot inside a body phantom. The body phantom consists of a plastic torso with a spinal cord model inside. The data model had been recorded with a CT scanner before.

During the test, each participant had to move a pointer to randomly located spots on the surface of the spine of the phantom (see figure 4.18). Each participant had to find twenty points for each visualization. For the tests, the model of the spine had been taken out of the body phantom in order to not provide any haptic feedback. The participants had to rely on their vision alone. The phantom guaranteed that the participants did not have a direct view to the point of interest, which is also generally the case for minimally invasive surgery. The participants have been asked to move the pointer to the target spot and indicate when they were done. The spot had been marked with a small black point in order not to interfere with the current visualization (figure 4.18).

Each participant was given two minutes to get used to the HMD and AR visualization. After this, the participants were asked to fulfill the tasks described above. The average time for the whole test was 16.4 minutes. This exceeds the expected usage time of the visualization system in the operating room.

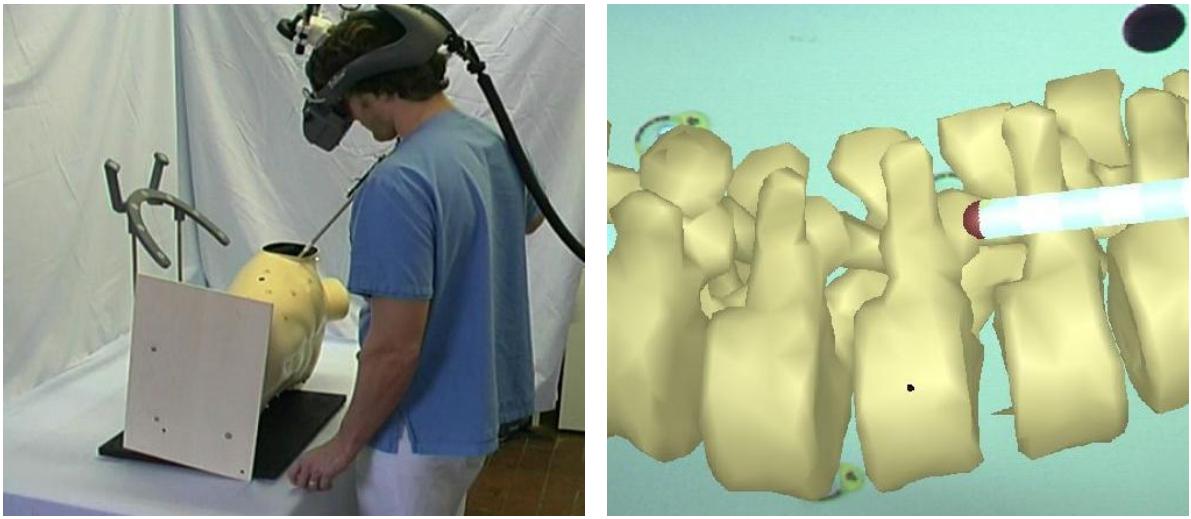


Figure 4.18: Augmented surgical instrument serves as the pointer device. The small black point is visualized. The surgeons were asked to touch the point with its tip in the virtual position.

4.2.4.1.3 Evaluated visualization modes

The following visualization modes were tested in random order in order to average out effects of learning and exhaustion. The table in figure 4.20 contains the visualizations in the same order as figure 4.19 does. Explanations of these follow this order as well. The visualization speed is denoted in brackets in frames per second.

Triangle mesh. The triangle mesh (see figure 4.19.4) is a representation of the surface of the bone structure in the scene. The surface has been segmented from the CT scan before the evaluation. The surface is stored in the computer as a list of triangles. In this mode only the edges of the triangles are displayed. Edges at greater distance to the viewer are displayed in a darker shading than closer ones in order to give a strong depth impression.

Surface rendering. The surface rendering (see figure 4.19.1,2,3,6) is, like the triangle mesh, a representation of the surface of the bone structure in the scene. The surface has also been segmented from the CT scan. The surface is visualized with untextured, but shaded solid triangles.

Volume rendering. Volume rendering (see figure 4.19.5,7) represents the whole volume rather than a surface. The data need not be segmented nor prepared. Each voxel is rendered with a certain transparency according to its value. The transfer function that relates the value to its color and transparency has been chosen in a way that bone structure in the spine model is emphasized. The rendering is performed with the support of 3D texture hardware as suggested by Hastreiter et al. [67]

Glass effect. This effect is only applied to the surface of the skin, but not to the vertebrae (see figure 4.19.6). The skin is rendered transparently and achromatically. Only reflections of a virtual light source on the skin are rendered. Hence, the skin looks similar to glass. The idea is to provide a visual impression that the skin is in front of the bone model, since the reflections of the glass skin occlude slightly the bone model or influence its color in certain areas.

Virtual window. This effect is also only applied to the skin (see figure 4.19.3,5). In this visualization add-on, we define a region on the skin which we call the virtual window. The visualization of the bone can only be seen through the virtual window, but not from outside. The window enables the effective depth cues *occlusion* and *motion parallax* because its frame partially occludes the spinal column in the background. The projections on the retina of these stationary objects move relative to each other, which is caused by observer movement.

4.2.4.2 Results

We have evaluated an overall number of 2398 spatial points that had to be touched in the model. We would like to point out that this evaluation is not intended to determine the accuracy of the system. Especially the exaggerated length of the pointer (64 cm) and its crude tip with a diameter of 9 mm were designed to provoke interaction errors. A shorter pointer that is peaked at the end yields far more accurate results, but it would emphasize errors of the system as well as unwanted movements (like tremor), which we would like to disregard.

The volume rendering visualization proved to have a performance that can be displayed in a current PC-based AR system with a large data volume. However, other kinds of volume visualizations that need prior segmentation showed a faster and more precise interaction. Still, the surgeons liked the volume rendering and reported a good perception of depth (see figure 4.20).

The triangle mesh mode did not perform as well as expected. A reason might be that the visualized model has a complex structure of many self-occluding surfaces that confuse the viewer when rendered with transparent triangles.

Figure 4.21 (top right) shows the performance in the course of time. The participants were able to work with the video see-through HMD without degrading performance during the whole experiment that took 16.4 minutes on average. The graphs show no indication of exhaustion during the test. The accuracy and speed of interaction improved over course of the whole experiment. At the beginning of the experiment, the participants trade accuracy for interaction speed (see figure 4.21, top right). We interpret this as an indicator for getting used to the test and the unfamiliar system.

Figure 4.21 (top left) shows the performance of each visualization mode. The numbers are in the same order as in figure 4.19 and 4.20. The different visualization modes have a clear impact on the interaction precision and speed. In our experiments, the frame rate seemed to be an important factor for the performance. The participants complained about slow visualization modes and showed a slower and less precise performance using these. The graph shows the average error as well as the error split up into x, y and z-axes, the latter of which is parallel to the viewing direction. The fast modes using surface rendering clearly show better results than the fast modes using triangle mesh. They also perform better than the slow modes using volume rendering or surface rendering. Therefore, we suggest using the two best visualization modes for interacting with a three-dimensional model in medical AR: These are the transparent surface rendering and the modes comprising the virtual window. It would be exciting to find out, if a combination of these could further improve the performance.

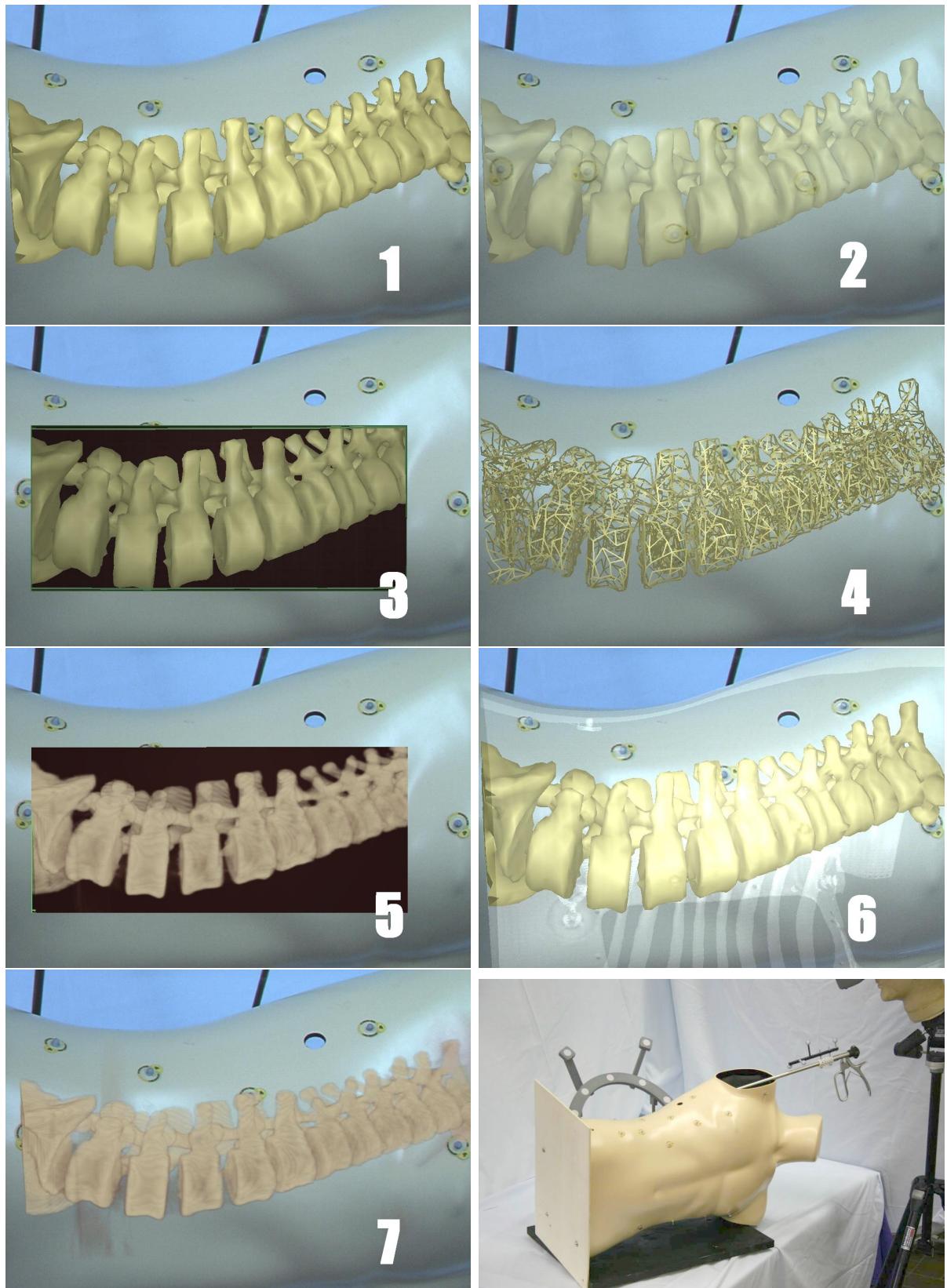


Figure 4.19: The seven evaluated visualizations; bottom right: The setup

Rendering mode	Depth perception	Effectiveness	Perception (1 = good, 5 = bad)	System frame-rate
	0	+	2,2381	30 fps
Surface rendering transparent, superimposed	+	+	2,5238	30 fps
Surface rendering through virtual window on skin	+	+	2,4762	30 fps
Triangle mesh	-	+	3,4211	30 fps
Volume rendering model through virtual window on skin	-	-	2,2857	9-10 fps
Surface rendering with glass effect for skin	-	-	2,8571	7-9 fps
Volume rendering superimposed	-	-	2,381	5-6 fps

Figure 4.20: Table of tested visualizations, including simplified results

Figure 4.21 (bottom left) shows the performance of each individual surgeon. Comparing the graphs depicting speed and accuracy, we can see that these do not correlate with each other. That means that the success depends on the skills and not on the speed of the performance.

4.2.4.3 Conclusion

We conclude that the kind of visualization makes a substantial difference in terms of interaction effectiveness. These findings do however not regard the question whether a visualization offers the desired medical information as this is application-dependent. We have only been interested in the depth perception of different visualizations for stereoscopic augmented reality in a medical context. Due to hardware developments in the past five years, we can offer new visualization techniques that have been compared to the classic ones. One of the new visualization types could outperform the other tested visualizations in terms of depth perception and effectiveness. This supports our thesis that visualization issues of augmented reality have to be better understood to take full advantage of its power.

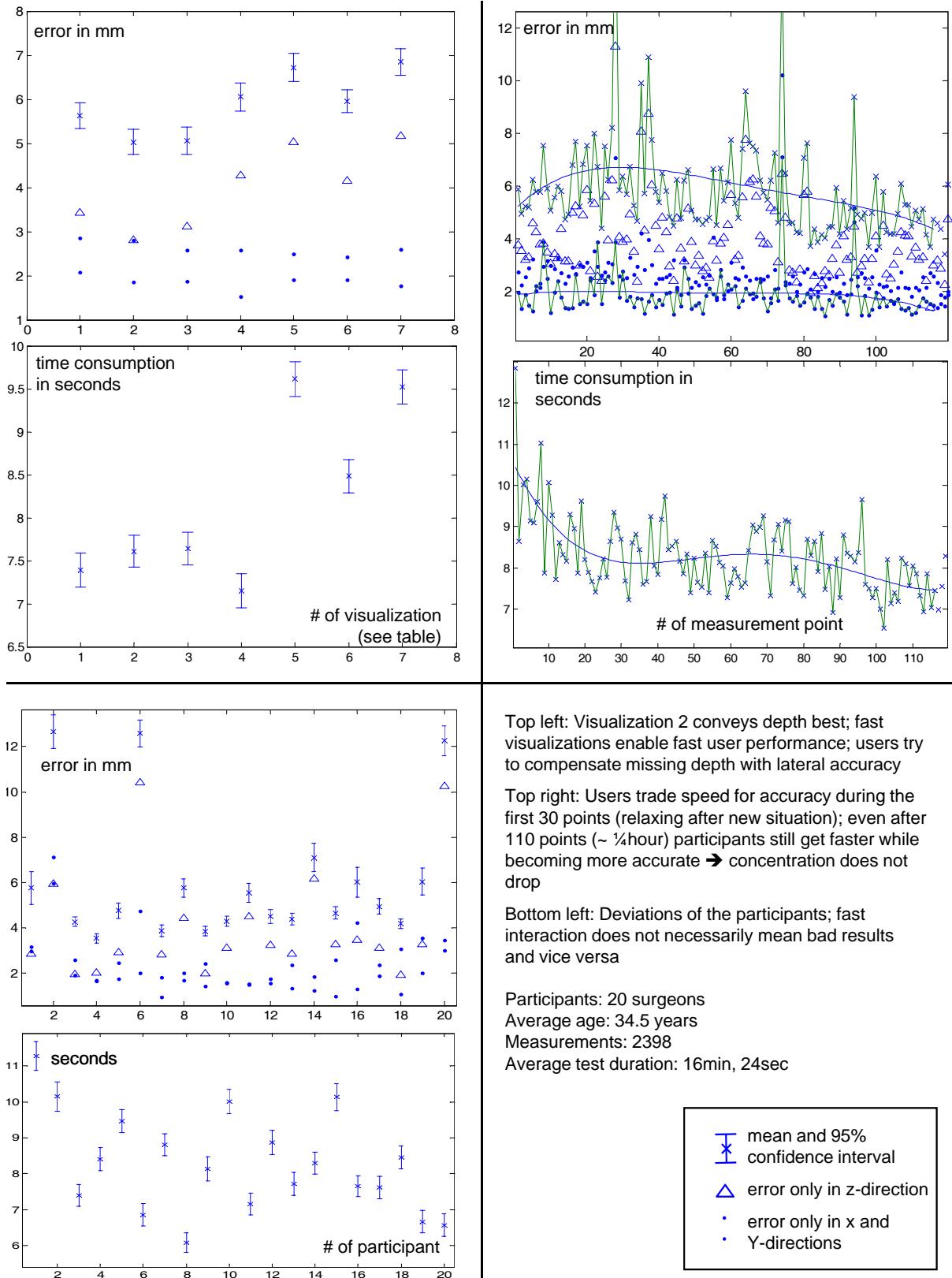


Figure 4.21: Figures of the evaluation

CHAPTER
FIVE

CONCLUSION

The thesis portraits the whole line of research of medical augmented reality summarizing possible hardware designs (chapter 1.1), necessary software components for AR systems (chapter 2), and integration of these components to a compelling setup (chapter 3), and validation of an AR system (chapter 4). It addresses bottlenecks of medical AR in each chapter.

The component spatial registration seems to be well-understood by the community. The known methods provide sufficient accuracy for several medical applications.

Temporal registration has been introduced in this thesis as a potentially important component of medical AR. The results of the presented methods are promising and the topic is currently investigated in a much deeper way as the scope of the thesis allowed for.

The optimal visualization of information in an AR system is a challenging and critical issue that depends on the information to visualize. We could show that the way of data presentation influences the depth perception and give suggestions for compelling AR visualizations.

Interaction with medical AR systems is a not well-explored topic. In the author's opinion it is not critical if the exact state of the surgery is detected by means of temporal registration. Such a system could be entirely controlled with an extremely limited number of buttons and implicit controls like the head pose and instrument location.

The software for integration of the components presented in the thesis offers an effective structure to model state-of-the-art AR systems. Several systems using different technologies are already implemented with it. It offers easy-to-use mechanisms for synchronization of data, means of run-time speed assessment, and efficient programming interfaces for rapid prototyping of applications while maintaining fast performance.

The thesis offers two new methods for assessing AR systems. Means of latency estimation and dynamic error propagation can be valuable tools in the arsenal of developers of medical AR systems.

REFERENCES

- [1] R Aggarwal, S Undre, K Moorthy, C Vincent, and A Darzi. The simulated operating theatre: comprehensive training for surgical teams. *Qual Saf Health Care*, 13(1):27–32, 2004.
- [2] Seyed-Ahmad Ahmadi, Tobias Sielhorst, Ralf Stauder, Martin Horn, Hubertus Feussner, and Nassir Navab. Recovery of surgical workflow without explicit models. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 420–428, 2006.
- [3] A. Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison-Wesley Professional, 2001.
- [4] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [5] Ronald T. Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21:34–47, 2001.
- [6] Michael Bajura, Henry Fuchs, and Ryutarou Ohbuchi. Merging virtual objects with the real world: seeing ultrasound imagery within the patient. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 203–210. ACM Press, 1992.
- [7] Michael Bajura and Ulrich Neumann. Dynamic registration correction in video-based augmented reality systems. *IEEE Comput. Graph. Appl.*, 15(5):52–60, 1995.
- [8] Martin Bauer, Bernd Bruegge, Gudrun Klinker, Asa MacWilliams, Thomas Reicher, Stefan Riß, Christian Sandor, and Martin Wagner. Design of a component-based augmented reality framework. In *International Symposium on Augmented Reality ISAR*, 2001.
- [9] Martin Bauer, Michael Schlegel, Daniel Pustka, Nassir Navab, and Gudrun Klinker. Predicting and estimating the accuracy of vision-based optical tracking systems. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, pages 43–51, Santa Barbara (CA), USA, October 2006.

- [10] Klaus-Peter Beier, James A. Freer, Howard Levine, Timothy A. Pletcher, Warren Russel, David J. Treloar, Dag K. J. E. von Lubitz, William Wilkerson, and Eric Wolf. An immersive virtual reality platform for medical education: Introduction to the medical readiness trainer. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000.
- [11] Rudolf Beisse, Michael Potulski, and Volker Bühren. Endoscopic techniques for the management of spinal trauma. *European Journal of Trauma*, 25(6):275–291, 2001.
- [12] Richard E. Bellman and Stuart E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, 1962.
- [13] G. Berci and K. A. Forde. History of endoscopy - what lessons have we learned from the past? *Surgical Endoscopy*, 14:5–15, 2002.
- [14] JW Berger and DS Shin. Computer-vision-enabled augmented reality fundus biomicroscopy. *Ophthalmology*, 106(10):1935–41, 1999.
- [15] 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 *ISMAR 2005*, pages 188–189, October 2005.
- [16] C. Bichlmeier, M. Rustaee, and S.M. Heining nad N. Navab. Virtually extended surgical drilling device: Virtual mirror for navigated spine surgery. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2007.
- [17] Christoph Bichlmeier, Tobias Sielhorst, and Nassir Navab. The tangible virtual mirror: New visualization paradigm for navigated surgery. In *International Workshop for Augmented Environments for Medical Imaging and Computer-aided Surgery (AMI-ARCS)*, Copenhagen, Denmark, Oct. 2006. MICCAI Society.
- [18] Frank A. Biocca and J. P. Rolland. Virtual eyes can rearrange your body: Adaptation to visual displacement in see-through, head-mounted displays. *Presence: Teleoperators and Virtual Env.*, 7:262–277, 1998.
- [19] Wolfgang Birkfellner, Michael Figl, Klaus Huber, Franz Watzinger, Felix Wanschitz, Johann Hummel, Rudolf Hanel, Wolfgang Greimel, Peter Homolka, Rolf Ewers, and Helmar Bergmann. A head-mounted operating binocular for augmented reality visualization in medicine - design and initial evaluation. *IEEE Trans. Med. Imag.*, 21(8):991–997, 2002.
- [20] Wolfgang Birkfellner, Klaus Huber, Franz Watzinger, Michael Figl, Felix Wanschitz, Rudolf Hanel, Dietmar Rafolt, Rolf Ewers, and Helmar Bergmann. Development of the varioscope ar – a see-through hmd for computer-aided surgery. In *Proc. IEEE and ACM Int'l Symp. on Augmented Reality*, pages 54–59. IEEE Computer Society Press, 2000.

-
- [21] Mike Blackwell, Constantinos Nikou, Anthony M. Di Gioia, and Takeo Kanade. An image overlay system for medical data visualization. In III William M. Wells, Alan C. F. Colchester, and Scott L. Delp, editors, *Proceedings of the First International Conference of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1496 of *Lecture Notes in Computer Science*, pages 232–240, Cambridge, MA, USA, October 1998. Springer-Verlag.
 - [22] Tobias Blum, Tobias Sielhorst, and Nassir Navab. Advanced augmented reality feedback for teaching 3d tool manipulation. In *New Technology Frontiers in Minimally Invasive Therapies*, chapter 25, pages 223–236. Lupiensis Biomedical Publications, Leece, Italy, 2007.
 - [23] Aaron F. Bobick and Andrew D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(12):1325–1337, 2005.
 - [24] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.
 - [25] Ivan Bricault, Gilbert Ferretti, and Philippe Cinquin. Registration of real and ct-derived virtual bronchoscopic images to assist transbronchial biopsy. *IEEE Trans. Med. Imag.*, 17(5):703–714, 1998.
 - [26] Frederick P. Brooks. The computer scientist as toolsmith ii. *Communications of the ACM*, 39(3):61–68, 1996.
 - [27] Darius Burschka, Ming Li, Russell Taylor, and Gregory D. Hager. Scale-invariant registration of monocular endoscopic images to ct-scans for sinus surgery. In C. Barillot, D.R. Haynor, and P. Hellier, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 3217 of *Lecture Notes in Computer Science*, pages 413–421. Springer-Verlag, 2004.
 - [28] Ozan Cakmakci and Jannick Rolland. Head-worn displays: A review. *Journal of Display Technology*, 2(3):199–216, September 2006.
 - [29] CJ Calvano, ME Moran, LD Tackett, PP Reddy, KE Boyle, and MM Pankratov. New visualization techniques for in utero surgery: amnioscopy with a three-dimensional head-mounted display and a computer-controlled endoscope. *J Endourol*, 12(5):407–10, 1998.
 - [30] T.P. Caudell and D.V. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing technology augmented reality. In *Proceedings of Hawaii International Conference on System Sciences*, 1992.
 - [31] Clayton M. Christensen. *The Innovator's Dilemma - When New Technologies Cause Great Firms to Fail*. Harvard Business School Press, 1998.

- [32] Kevin Cleary, Ho Young Chung, and Seong K. Mun. Or 2020: The operating room of the future. *Laparoendoscopic and Advanced Surgical Techniques*, 15(5):495–500, 2005.
- [33] James E. Cutting and Peter M. Vishton. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In W. Epstein & S. Rogers (Eds.), *Perception of Space and Motion*, pages 69–117, 1995.
- [34] M. Das, F. Sauer, U.J. Schoepf, A. Khamene, S.K. Vogt, S. Schaller, R. Kikinis, E. vanSonnenberg, and S.G. Silverman. Augmented reality visualization for ct-guided interventions: System description, feasibility, and initial evaluation in an abdominal phantom. *Radiology*, 240(1):230, 2006.
- [35] L. Davis, E. Clarkson, and JP Rolland. Predicting accuracy in pose estimation for marker-based tracking. pages 28–35, 2003.
- [36] Stijn De Buck, Johan Van Cleynenbreuge, Indra Geys, Thomas Koninckx, Philippe R. Koninck, and Paul Suetens. A system to support laparoscopic surgery by augmented reality visualization. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 691–698, 2001.
- [37] D. Dey, DG Gobbi, PJ Slomka, KJM Surry, and TM Peters. Automatic fusion of freehand endoscopic brain images to three-dimensional surfaces: creating stereoscopic panoramas. *IEEE Trans. Med. Imag.*, 21(1):23–30, 2002.
- [38] D. Dey, P.J. Slomka, D.G. Gobbi, and T.M. Peters. Mixed reality merging of endoscopic images and 3-d surfaces. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 796–803. Springer-Verlag, 2000.
- [39] A. Dosis, F. Bello, K. Moorthy, Y. Munz, D. Gillies, and A. Darzi. Real-time synchronization of kinematic and video data for the comprehensive assessment of surgical skills. In *Medicine Meets Virtual Reality (MMVR)*, 2004.
- [40] David Drascic and Paul Milgram. Perceptual issues in augmented reality. *SPIE Volume 2653: Stereoscopic Displays and Virtual Reality Systems*, 2653:123–134, 1996.
- [41] James Ellsmere, Jeffrey Stoll, David W. Rattner, David Brooks, Robert Kane, William M. Wells, III, Ron Kikinis, and Kirby Vosburgh. A navigation system for augmenting laparoscopic ultrasound. In Randy E. Ellis and Terry M. Peters, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, pages 184–191. Springer-Verlag, 2003.
- [42] Steven Feiner, Blair Macintyre, and Dorée Seligmann. Knowledge-based augmented reality. *Commun. ACM*, 36(7):53–62, 1993.

- [43] Marco Feuerstein, Thomas Mussack, Sandro M. Heining, and Nassir Navab. Registration-free laparoscope augmentation for intra-operative liver resection planning. In Kevin R. Cleary and Michael I. Miga, editors, *Medical Imaging 2007: Visualization and Image-Guided Procedures*, Proceedings of SPIE, San Diego, California, USA, February 2007.
- [44] Marco Feuerstein, Thomas Mussack, Sandro M. Heining, and Nassir Navab. Intra-operative laparoscope augmentation for port placement and resection planning in minimally invasive liver resection. *To appear in IEEE Trans. Med. Imag.*, 2008.
- [45] Marco Feuerstein, Tobias Reichl, Jakob Vogel, Armin Schneider, Hubertus Feussner, and Nassir Navab. Magneto-optic tracking of a flexible laparoscopic ultrasound transducer for laparoscope augmentation. In N. Ayache, S. Ourselin, and A. Maeder, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 4791 of *Lecture Notes in Computer Science*, pages 458–466, Brisbane, Australia, October/November 2007. Springer-Verlag.
- [46] Marco Feuerstein, Stephen M. Wildhirt, Robert Bauernschmitt, and Nassir Navab. Automatic patient registration for port placement in minimally invasive endoscopic surgery. In James S. Duncan and Guido Gerig, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 3750 of *Lecture Notes in Computer Science*, pages 287–294, Palm Springs, CA, USA, September 2005. Springer-Verlag.
- [47] Mark Fiala. Arttag, a fiducial marker system using digital techniques. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596, 2005.
- [48] Gabor Fichtinger, Anton Deguet, Ken Masamune, Emese Balogh, Gregory S. Fischer, Herve Mathieu, Russell H. Taylor, S. James Zinreich, and Laura M. Fayad. Image overlay guidance for needle insertion in ct scanner. *IEEE Transactions on Biomedical Engineering*, 52(8):1415–1424, 2005.
- [49] M. Figl, C. Ede, J. Hummel, F. Wanschitz, R. Ewers, H. Bergmann, and W. Birkfellner. A fully automated calibration method for an optical see-through head-mounted operating microscope with variable zoom and focus. *IEEE Trans. Med. Imag.*, 24(11):1492–1499, 2005.
- [50] Gregory S. Fischer, Anton Deguet, Daniel Schlattman, Russell Taylor, Laura Fayad, S. James Zinreich, and Gabor Fichtinger. Mri image overlay: Applications to arthrography needle insertion. In *Medicine Meets Virtual Reality (MMVR) 14*, 2006.
- [51] J. Michael Fitzpatrick, Jay B. West, and Calvin R. Maurer, Jr. Predicting error in rigid-body point-based registration. *IEEE Trans. Med. Imag.*, 14(5):694–702, 1998.
- [52] W. Freysinger, A.R. Gunkel, and W.F. Thumfart. Image-guided endoscopic ent surgery. *European archives of Otorhinolaryngology*, 254(7):343–346, 1997.

- [53] Eric M. Friets, John W. Strohbehn, John F. Hatch, and David W. Roberts. A frameless stereotaxic operating microscope for neurosurgery. *IEEE Transactions on Biomedical Engineering*, 36(6), 1989.
- [54] E.W. Fritsch. Navigation in spinal surgery using fluoroscopy. In J. B. Stiehl, W. H. Konermann, and R. G. A. Haaker, editors, *Navigation and Robotics in Total Joint and Spine Surgery*, chapter 68, pages 487–494. Springer, 2004.
- [55] Henry Fuchs, Mark A. Livingston, Ramesh Raskar, D'nardo Colucci, Kurtis Keller, Andrei State, Jessica R. Crawford, Paul Rademacher, Samuel H. Drake, and Anthony A. Meyer. Augmented reality visualization for laparoscopic surgery. In III William M. Wells, Alan C. F. Colchester, and Scott L. Delp, editors, *Proceedings of the First International Conference of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1496 of *Lecture Notes in Computer Science*, pages 934–943, Cambridge, MA, USA, October 1998. Springer-Verlag.
- [56] A.L. Fuhrmann, R. Splechtna, and J. Prikryl. Comprehensive calibration and registration procedures for augmented reality. pages 219–228, 2001.
- [57] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley, 1995.
- [58] T. Gasser, A. Kneip, A. Binding, A. Prader, and L. Molinari. The dynamics of linear growth in distance, velocity and acceleration. *Annals of Human Biology*, 18(3):187–205, 1991.
- [59] J. Geerling, U. Berlemann, B. Frericks, M. Kfuri, T. Hüfner, and C. Krettek. Pedicle screw placement. In J. B. Stiehl, W. H. Konermann, and R. G. A. Haaker, editors, *Navigation and Robotics in Total Joint and Spine Surgery*, chapter 67, pages 481–486. Springer, 2004.
- [60] N. Glossop and Z. Wang. Laser projection augmented reality system for computer assisted surgery. In Randy E. Ellis and Terry M. Peters, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 2879 of *Lecture Notes in Computer Science*, pages 239–246. Springer-Verlag, 2003.
- [61] Gernot Goebbel, Klaus Troche, Matthias Braun, Ana Ivanovic, Armin Grab, Kai von Lübtow, Robert Sader, Florian Zeilhofer, Klaus Albrecht, and Karin Praxmarer. Arsysstricorder – development of an augmented reality system for intraoperative navigation in maxillofacial surgery. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, 2003.
- [62] W. E. L. Grimson, T. Lozano-Perez, W. M. Wells, III, G. J. Ettinger, S. J. White, and R. Kikinis. An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. *IEEE Trans. Med. Imag.*, 15(2):129–140, 1996.

- [63] P. A. Grützner, A. Hebecker, H. Waelti, B. Vock, L.-P. Nolte, and A. Wentzensen. Clinical study for registration-free 3d-navigation with the siremobil iso-c^{3D} mobile c-arm. *electromedica*, 71(1):7–16, 2003.
- [64] Robert Grzeszczuk, Shao Chin, Rebecca Fahrig, Hamid Abassi, Hilary Holz, John Adler Daniel Kim1, and Ramin Shahidi. A fluoroscopic x-ray registration process for three-dimensional surgical navigation. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science. Springer-Verlag, 2000.
- [65] Dimitrios Gunopoulos. Discovering similar multidimensional trajectories. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, page 673. IEEE Computer Society, 2002.
- [66] G.B. Hanna, SM Shimi, and A. Cuschieri. Task performance in endoscopic surgery is influenced by location of the image display. *Ann Surg*, 227(4):481–4, 1998.
- [67] Peter Hastreiter, Christof Rezk-Salama, Bernd Tomandl, K. Eberhardt, and Thomas Ertl. Fast analysis of intracranial aneurysms based on interactive direct volume rendering and cta. In III William M. Wells, Alan C. F. Colchester, and Scott L. Delp, editors, *Proceedings of the First International Conference of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1496 of *Lecture Notes in Computer Science*, pages 660–669, Cambridge, MA, USA, October 1998. Springer-Verlag.
- [68] J.F. Hatch and J.W. Strohbehn D.W.Roberts. Reference-display system for the integration of ct scanning and the operating microscope. In *Proceedings of the Eleventh Annual Northeast Bioengineering Conference*, 1985.
- [69] Janne Heikkilä and Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1112. IEEE Computer Society, 1997.
- [70] Sandro-Michael Heining, Philipp Stefan, Latifa Omary, Stefan Wiesner, Tobias Sielhorst, Nassir Navab, Frank Sauer, Ekkehard Euler, Wolf Mutschler, and Joerg Traub. Evaluation of an in-situ visualization system for navigated trauma surgery. *Journal of Biomechanics*, 39(Supplement 1):209, 2006.
- [71] C Herfarth. 'lean' surgery through changes in surgical workflow. *British Journal of Surgery*, 90(5):513–514, 2003.
- [72] W. A. Hoff and T. L. Vincent. Analysis of head pose accuracy in augmented reality. *IEEE Trans. Visualization and Computer Graphics*, 6, 2000.
- [73] R.L. Holloway. Registration error analysis for augmented reality. *Presence: Teleoperators and Virtual Env.*, 6(4):413–432, 1997.

- [74] Marco C. Jacobs, Mark A. Livingston, and Andrei State. Managing latency in complex augmented reality systems. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 49–54. ACM Press, 1997.
- [75] Rohen Johannes W and Chihiro Yokochi. *Anatomie des Menschen*. Schattauer, 2nd edition edition, 1988.
- [76] Laura G Johnson, Philip Edwards, and David Hawkes. Surface transparency makes stereo overlays unpredictable: The implications for augmented reality. In J. D. Westwood, editor, *Medicine Meets Virtual Reality (MMVR)*, volume 94 of *Studies in Health Technology and Informatics*, pages 131–136. IOS Press, 2002.
- [77] Athanassios Kassidas, John F. MacGregor, and Paul A. Taylor. Synchronization of batch trajectories using dynamic time warping. *AICHE Journal*, 44(4):864–875, 1998.
- [78] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the Second IEEE and ACM International Workshop on Augmented Reality*, 1999.
- [79] Hannes Kaufmann and Dieter Schmalstieg. Mathematics and geometry education with collaborative augmented reality. In *SIGGRAPH 2002 Conference Abstracts and Applications*, pages 37–41, 2002.
- [80] T. Kawamata, H. Iseki, T. Shibasaki, and T. Hori. Endoscopic augmented reality navigation system for endonasal transsphenoidal surgery to treat pituitary tumors: technical note. *Neurosurgery*, 50(6):1393–1397, 2002.
- [81] P. J. Kelly, G.J. Alker, and S. Goerss. Computer-assisted stereotactic laser microsurgery for the treatment of intracranial neoplasms. *Neurosurgery*, 10:324–331, 1982.
- [82] Rasool Khadem, Michael R. Bax, Jeremy A. Johnson, Eric P. Wilkinson, and Ramin Shahidi. Endoscope calibration and accuracy testing for 3d/2d image registration. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 1361–1362, 2001.
- [83] Rasool Khadem, Clement C. Yeh, Mohammad Sadeghi-Tehrani, Michael R. Bax, Jeremy A. Johnson, Jacqueline Nerney Welch, Eric P. Wilkinson, and Ramin Shahidi. Comparative tracking error analysis of five different optical tracking systems. *Computer Aided Surgery*, 5(2):98–107, June 2000.
- [84] A. P. King, P. J. Edwards, C. R. Maurer, Jr., D. A. de Cunha, D. J. Hawkes, D. L. G. Hill, R. P. Gaston, M. R. Fenlon, A. J. Strong, C. L. Chandler, A. Richards, and M. J. Gleeson. Design and evaluation of a system for microscope-assisted guided interventions. *IEEE Trans. Med. Imag.*, 19(11):1082–1093, 2000.

-
- [85] Tassilo Klein, Joerg Traub, Hubert Hautmann, Alireza Ahmadian, and Nassir Navab. Fiducial free registration procedure for navigated bronchoscopy. 2007. To appear at Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI).
 - [86] E. M. Kolasinski, S. L. Goldberg, and J. H. Hiller. Simulator sickness in virtual environments. Technical report, U.S. Army Research Institute – Simulator Systems Research Unit, 1995. Technical Report 1027.
 - [87] Jens Krüger and Rüdiger Westermann. Acceleration Techniques for GPU-based Volume Rendering. In *Proceedings IEEE Visualization 2003*, 2003.
 - [88] S. Krüger, F. Vogt, W. Hohenberger, D. Paulus, H. Niemann, and C. H. Schick. Evaluation of computer-assisted image enhancement in minimal invasive endoscopic surgery. *Methods Inf Med*, 43(4):362–366, 2004.
 - [89] W.Y. Lai and H.B.L. Duh. Effects of frame rate for visualization of dynamic quantitative information in a head-mounted display. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 7, 2004.
 - [90] F. Langlotz. Pedicle screw implatation using the disos template system. In J. B. Stiehl, W. H. Konermann, and R. G. A. Haaker, editors, *Navigation and Robotics in Total Joint and Spine Surgery*, chapter 1, pages 501–505. Springer, 2004.
 - [91] F. Langlotz. Where do we go from here? In J. B. Stiehl, W. H. Konermann, and R. G. A. Haaker, editors, *Navigation and Robotics in Total Joint and Spine Surgery*, chapter 1, pages 509–513. Springer, 2004.
 - [92] WY Lau. History of Endoscopic and Laparoscopic Surgery. *World Journal of Surgery*, 21(4):444–453, 1997.
 - [93] M. Levoy. Display of surfaces from volume data. *Computer Graphics and Applications, IEEE*, 8(3):29–37, 1988.
 - [94] Chuanjun Li, Peng Zhai, Si-Qing Zheng, and Balakrishnan Prabhakaran. Segmentation and recognition of multi-attribute motion sequences. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 836–843. ACM Press, 2004.
 - [95] Hong Li and Michael Greenspan. Multi-scale gesture recognition from time-varying contours. *Tenth IEEE International Conference on Computer Vision (ICCV)*, 1:236–243, 2005.
 - [96] H. Liao, N. Hata, S. Nakajima, M. Iwahara, I. Sakuma, and T. Dohi. Surgical navigation by autostereoscopic image overlay of integral videography. *Information Technology in Biomedicine, IEEE Transactions on*, 8(2):114–121, 2004.

- [97] H. Liao, S. Nakajima, M. Iwahara, E. Kobayashi, I. Sakuma, N. Yahagi, and T. Dohi. Intra-operative Real-Time 3-D Information Display System based on Integral Videography. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 392–400. Springer-Verlag, 2001.
- [98] Henry C. Lin, Izhak Shafran, Todd E. Murphy, Allison M. Okamura, David D. Yuh, and Gregory D. Hager. Automatic detection and segmentation of robot-assisted surgical motions. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 3749, pages 802–810. Springer, 2005.
- [99] G.S. Litynski. Endoscopic Surgery: The History, the Pioneers. *World Journal of Surgery*, 23(8):745–753, 1999.
- [100] W. Lorensen, H. Cline, C. Nafis, R. Kikinis, D. Altobelli, L. Gleason, G.E. Co, and NY Schenectady. Enhancing reality in the operating room. In *IEEE Conference on Visualization*, pages 410–415, 1993.
- [101] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press.
- [102] Blair MacIntyre, Maribeth Gandy, Jay Bolter, Steven Dow, and Brendan Hannigan. Dart: The designer's augmented reality toolkit. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, pages 329–339, Tokyo, Japan, October 7-10 2003.
- [103] Wendy Mackay, Gilles Velay, Kathy Carter, Chaoying Ma, and Daniele Pagani. Augmenting reality: Adding computational dimensions to paper. *Commun. ACM*, 36(7):96–97, 1993.
- [104] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, March 1998.
- [105] G. Marti, V. Bettschart, J.-S. Billiard, and C. Baur. Hybrid method for both calibration and registration of an endoscope with an active optical tracker. In *Computer Assisted Radiology and Surgery*, 2004.
- [106] Ken Masamune, Yoshitaka Masutani, Susumu Nakajima, Ichiro Sakuma, Takeyoshi Dohi, Hiroshi Iseki, and Kintomo Takakura. Three-dimensional slice image overlay system with accurate depth perception for surgery. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 1935 of *Lecture Notes in Computer Science*, pages 395–402. Springer-Verlag, October 2000.
- [107] Y. Masutani, M. Iwahara, O. Samuta, Y. Nishi, N. Suzuki, M. Suzuki, T. Dohi, H. Iseki, and K. Takakura. Development of integral photography-based enhanced reality visualization system for surgical support. *Proc. of ISCAS*, 95:16–17, 1995.

-
- [108] Calvin R. Maurer, Jr., J. Michael Fitzpatrick, Matthew Y. Wang, Jr. Robert L. Galloway, Robert J. Maciunas, and George S. Allen. Registration of head volume images using implantable fiducial markers. *IEEE Trans. Med. Imag.*, 16(4):447–462, 1997.
 - [109] Scott Meyers. *More Effective C++: 35 New Ways to Improve Your Programs and Designs*. Addison-Wesley, 1st edition, 1995.
 - [110] Scott Meyers. *Effective C++: 50 Specific Ways to Improve Your Programs and Design*. Addison-Wesley, 2nd edition, 1997.
 - [111] Scott Meyers. *Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library*. Addison-Wesley, 1st edition, 2001.
 - [112] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. In *IEICE Trans. Information Systems*, pages 1321–1329, 1994.
 - [113] Kensaku Mori, Daisuke Deguchi, Kenta Akiyama, Takayuki Kitasaka, Calvin R. Maurer, Jr., Yasuhito Suenaga, Hirotugu Takabatake, Masaki Mori, , and Hiroshi Natori. Hybrid bronchoscope tracking using a magnetic tracking sensor and image registration. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 543–550. Springer-Verlag, 2005.
 - [114] Kensaku Mori, Daisuke Deguchi, Jun ichi Hasegawa, Yasuhito Suenaga, Jun ichiro Toriwaki, Hirotugu Takabatake, and Hiroshi Natori. A method for tracking the camera motion of real endoscope by epipolar geometry analysis and virtual endoscopy system. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 2208 of *Lecture Notes in Computer Science*, pages 1–8. Springer-Verlag, 2001.
 - [115] Fabien Mourgues and Ève Coste-Manière. Flexible calibration of actuated stereoscopic endoscope for overlay in robot assisted surgery. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 25–34, 2002.
 - [116] Mario E. Munich and Pietro Perona. Camera-based id verification by signature tracking. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume I*, pages 782–796. Springer-Verlag, 1998.
 - [117] Martin J. Murphy. An automatic six-degree-of-freedom image registration algorithm for image-guided frameless stereotaxic radiosurgery. *Medical Physics*, 24(6):857–866, 1997.
 - [118] Nassir Navab, Marco Feuerstein, and Christoph Bichlmeier. Laparoscopic virtual mirror - new interaction paradigm for monitor based augmented reality. In *Virtual Reality*, pages 43–50, Charlotte, North Carolina, USA, March 2007.
 - [119] Nassir Navab, Matthias Mitschke, and Ali Bani-Hashemi. Merging visible and invisible: Two camera-augmented mobile C-arm (CAMC) applications. In *Proc.*

- IEEE and ACM Int'l Workshop on Augmented Reality*, pages 134–141, San Francisco, CA, USA, 1999.
- [120] S.A. Nicolau, L. Goffin, and L. Soler. A low cost and accurate guidance system for laparoscopic surgery: Validation on an abdominal phantom. In *ACM Symposium on Virtual Reality Software and Technology*, pages 124–133, November 2005.
 - [121] Stéphane Nicolau, Xavier Pennec, Luc Soler, and Nicholas Ayache. An accuracy certified augmented reality system for therapy guidance. In *Proc. of the 8th European Conference on Computer Vision (ECCV 04)*, volume 3023 of *Lecture Notes in Computer Science*, pages 79–91, Prague, May 2004. Springer-Verlag.
 - [122] Takuya Nojima, Dairoku Sekiguchi, Masahiko Inami, and Susumu Tachi. The smarttool: A system for augmented reality of haptics. In *IEEE Virtual Reality Conference*, 2002.
 - [123] Tobias Obst, Reiner Burgkart, Eugen Ruckhäberle, and Robert Riener. The delivery simulator: A new application of medical vr. In *Proceedings of Medicine Meets Virtual Reality 12*, 2004.
 - [124] Angela Olinger and Ulrich Hildebrandt. *Endoskopische Wirbelsäulenchirurgie - thorakal, transperitoneal, retroperitoneal*. Springer, 2000.
 - [125] TM Peters, CJ Henri, P. Munger, AM Takahashi, AC Evans, B. Davey, and A. Olivier. Integration of stereoscopic DSA and 3D MRI for image-guided neurosurgery. *Comput Med Imaging Graph*, 18(4):289–99, 1994.
 - [126] Andreas Petersik, Bernhard Pflessner, Ulf Tiede, Karl-Heinz Höhne, and Rudolf Leuwer. Realistic haptic interaction in volume sculpting for surgery simulation. In *Proceedings of Surgery Simulation and Soft Tissue Modeling (IS4TM)*, pages 194–202, 2004.
 - [127] P.J. Edwards, Dr. D.L.G. Hill, Dr. D.J. Hawkes, and Dr. A.C.F. Colchester. Neurosurgical guidance using the stereo microscope. In *Proceedings of Computer Vision, Virtual Reality and Robotics in Medicine, First International Conference, CVRMed'95*, 1995.
 - [128] Simon Prince, Adrian David Cheok, Farzam Farbiy, Todd Williamson, Nik Johnson, Mark Billinghurst, and Hirokazu Kato. 3d live: Real time captured content for mixed reality. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, 2002.
 - [129] Bernhard Reitinger, Alexander Bornik, Reinhard Beichel, and Dieter Schmalstieg. Liver Surgery Planning Using Virtual Reality. *IEEE Computer Graphics and Applications*, 26(6):36–47, 2006.
 - [130] A. Zisserman R.I. Hartley. *Multiple View Geometry in Computer Vision*. Cambridge University Pres, 2000.

- [131] G.E. Riccio. An ecological theory of motion and postural instability. *Ecological Psychology*, 3(3):195–240, 1991.
- [132] Stefan Riedl. Modern operating room management in the workflow of surgery. spectrum of tasks and challenges of the future. *Der Chirurg*, 73(2):105–110, 2002.
- [133] D.W. Roberts, J.W. Strohbehn, J.F. Hatch, W. Murray, and H. Kettenberger. A frameless stereotaxic integration of computerized tomographic imaging and the operating microscope. *J Neurosurg*, 65(4):545–9, 1986.
- [134] Jannick P. Rolland and Henry Fuchs. Optical versus video see-through head-mounted displays in medical visualization. *Presence*, 9:287–309, 2000.
- [135] J.P. Rolland, L. Davis, and Y. Baillot. A survey of tracking technology for virtual environments. pages 67–112. 2001.
- [136] J Rosen, M Solazzo, B Hannaford, and M Sinanan. Task decomposition of laparoscopic surgery for objective evaluation of surgical residents’ learning curve using hidden markov model. *Comput Aided Surg.*, 7(1):49–61, 2002.
- [137] R.Rao and M. Singrakhia. *Fundamentals in Spine Surgery*, chapter 65, pages 467–476. Springer, 2004.
- [138] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.*, 26(1):43–49, 1978.
- [139] Marilyn C. Salzman, Chris Dede, R. Bowen Loftin, and Jim Chen. A model for understanding how virtual reality aids complex conceptual learning. *Presence: Teleoperators and Virtual Env.*, 8(3):293–316, 1999.
- [140] T. Sasama, T. Ochi, S. Tamura, N. Sugano, Y. Sato, Y. Momoi, T. Koyama, Y. Nakajima, I. Sakuma, M.G. Fujie, et al. A novel laser guidance system for alignment of linear surgical tools: Its principles and performance evaluation as a man-machine system. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 125–132. Springer-Verlag London, UK, 2002.
- [141] Yoshinobu Sato, Masahiko Nakamoto, Yasuhiro Tamaki, Toshihiko Sasama, Isao Sakita, Yoshikazu Nakajima, Morito Monden, and Shinichi Tamura. Image guidance of breast cancer surgery using 3-d ultrasound images and augmented reality visualization. *IEEE Trans. Med. Imag.*, 17(5), 1998.
- [142] Frank Sauer, Ali Khamene, and Sebastian Vogt. An augmented reality navigation system with a single-camera tracker: System design and needle biopsy phantom trial. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 2489, pages 116–124, 2002.

- [143] Frank Sauer, Fabian Wenzel, Sebastian Vogt, Yiyang Tao, Yakup Genc, and Ali Bani-Hashemi. Augmented workspace: designing an ar testbed. In *Proc. IEEE and ACM Int'l Symp. on Augmented Reality*, pages 47–53, 2000.
- [144] M. Scheuering, A. Schenk, A. Schneider, B. Preim, and G. Greiner. Intraoperative augmented reality for minimally invasive liver interventions. In *Medical Imaging 2003: Visualization, Image-Guided Procedures, and Display*, Proceedings of SPIE, 2003.
- [145] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavari, L. Miguel Encarnacao, and Michael Gervautz Werner Purgathofer. The studierstube augmented reality project. *Presence: Teleoperators and Virtual Env.*, 11(1):33–54, 2002.
- [146] J. Schmidt, F. Vogt, and H. Niemann. Robust hand-eye calibration of an endoscopic surgery robot using dual quaternions. In B. Michaelis and G. Krell, editors, *Pattern Recognition, 25th DAGM Symposium*, volume 2781 of *Lecture Notes in Computer Science*, pages 548–556, Berlin, Heidelberg, New York, 2003. Springer-Verlag.
- [147] M. Scholz, W. Konen, S. Tombrock, B. Fricke, and L. Adams. Development of an endoscopic navigation system based on digital image processing. *Computer Aided Surgery*, 3:134–143, 1998.
- [148] Ramin Shahidi, Michael R. Bax, Calvin R. Maurer, Jr., Jeremy A. Johnson, Eric P. Wilkinson, Bai Wang, Jay B. West, Martin J. Citardi, Kim H. Manwaring, and Rasool Khadem. Implementation, calibration and accuracy testing of an image-enhanced endoscopy system. *IEEE Trans. Med. Imag.*, 21(12):1524–1535, 2002.
- [149] Ramin Shahidi, Bai Wang, Marc Epitaux, Robert Grzeszczuk, and John Adler. Volumetric image guidance via a stereotactic endoscope. In III William M. Wells, Alan C. F. Colchester, and Scott L. Delp, editors, *Proceedings of the First International Conference of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1496 of *Lecture Notes in Computer Science*, pages 241–252, Cambridge, MA, USA, October 1998. Springer-Verlag.
- [150] Tobias Sielhorst. High accuracy tracking for medical augmented reality. Master's thesis, Technische Universität München, 2003.
- [151] Tobias Sielhorst, Martin A. Bauer, Oliver Wenisch, Gudrun Klinker, and Nassir Navab. Online estimation of the target registration error for n-ocular optical tracking systems. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2007. To appear at Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI).
- [152] Tobias Sielhorst, Christoph Bichlmeier, Sandro Michael Heining, and Nassir Navab. Depth perception a major issue in medical ar: Evaluation study by twenty surgeons. In R. Larsen, M. Nielsen, and J. Sporring, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, 2006.

- [153] Tobias Sielhorst, Tobias Blum, and Nassir Navab. Synchronizing 3d movements for quantitative comparison and simultaneous visualization of actions. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, 2005.
- [154] Tobias Sielhorst, Tobias Obst, Rainer Burgkart, Robert Riener, and Nassir Navab. An augmented reality delivery simulator for medical training. In *International Workshop on Augmented Environments for Medical Imaging - MICCAI Satellite Workshop*, 2004.
- [155] Tobias Sielhorst, Wu Sa, Ali Khamene, Frank Sauer, and Nassir Navab. Measurement of absolute latency for video see through augmented reality. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, 2007.
- [156] Tobias Sielhorst, Ralf Stauder, Martin Horn, Thomas Mussack, Armin Schneider, Hubertus Feussner, and Nassir Navab. Simultaneous replay of automatically synchronized videos of surgeries for feedback and visual assessment. *International Journal of Computer Assisted Radiology and Surgery, Supplement 1*, 2:433–434, June 2007.
- [157] Tobias Sielhorst, Jörg Traub, and Nassir Navab. The ar apprenticeship: Replication and omnidirectional viewing of subtle movements. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, 2004.
- [158] Warren E. Smith, Nimish Vakil, and Seth A. Maislin. Correction of distortion in endoscope images. *IEEE Trans. Med. Imag.*, 11(1):117–122, 1992.
- [159] Andrei State, Mark A. Livingston, William F. Garrett, Gentaro Hirota, Mary C. Whitton, Etta D. Pisano, and Henry Fuchs. Technologies for augmented reality systems: realizing ultrasound-guided needle biopsies. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 439–446, New York, NY, USA, 1996. ACM Press.
- [160] James D. Stefansic, Alan J. Herline, Yu Shyr, William C. Chapman, J. Michael Fitzpatrick, and Robert L. Galloway. Registration of physical space to laparoscopic image space for use in minimally invasive hepatic surgery. *IEEE Transactions on Medical Imaging*, 19(10):1012–1023, 2000.
- [161] H. Steinhaus. Sur la localisation au moyen des rayons x. *Comptes Rendus de L'Academie des Science*, 206:1473–5, 1938.
- [162] G.D. Stetten, V.S. Chib, and R.J. Tamburo. Tomographic reflection to merge ultrasound images with direct vision. In *IEEE Proceedings of the Applied Imagery Pattern Recognition (AIPR) annual workshop*, pages 200–205, 2000.
- [163] George D. Stetten and Vikram S. Chib. Overlaying ultrasound images on direct vision. *Journal Ultrasound in Medicine*, 20:235–240, 2001.
- [164] J.B. Stiehl, W.H. Konermann, and R.G. Haaker, editors. *Navigation and Robotics in Total Joint and Spine Surgery*. Springer, 2004.

- [165] Danail Stoyanov, Ara Darzi, and Guang Zhong Yang. Laparoscope self-calibration for robotic assisted minimally invasive surgery. In J. Duncan and G. Gerig, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 3750 of *Lecture Notes in Computer Science*, pages 114–121. Springer-Verlag, 2005.
- [166] G. Strauß, M. Fischer, J. Meixensberger, V. Falk, C. Trantakis, D. Winkler, F. Bootz, O. Burgert, A. Dietz, and H.U. Lemke. Bestimmung der effizienz von intraoperativer technologie. *HNO*, 54(7):528–535, 2006.
- [167] Bjarne Stroustrup. *The C++ Programming Language 3rd Edition*. Addison Wesley, 1997.
- [168] Ivan Sutherland. A head-mounted three dimensional display. In *Proceedings of the Fall Joint Computer Conference*, pages 757–764, 1968.
- [169] Ramesh U. Thoranaghatte, Guoyan Zheng, Frank Langlotz, and Lutz-Peter Nolte. Endoscope-based hybrid navigation system for minimally invasive ventral spine surgeries. *Computer Aided Surgery*, 10(5/6):351–356, September/November 2005.
- [170] J. Traub, H. Heibel, P. Dressel, S.M. Heining, R. Graumann, and N. Navab. A multi-view opto-xray imaging system: Development and first application in trauma surgery. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2007.
- [171] Joerg Traub, Philipp Stefan, Sandro-Michael Heining, Tobias Sielhorst, Christian Riquarts, Ekkehard Euler, and Nassir Navab. Hybrid navigation interface for orthopedic and trauma surgery. In R. Larsen, M. Nielsen, and J. Sporring, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 4190 of *Lecture Notes in Computer Science*, pages 373–380, Copenhagen, Denmark, Oct. 2006. MICCAI Society, Springer.
- [172] Joerg Traub, Philipp Stefan, Sandro-Michael Heining, Tobias Sielhorst, Christian Riquarts, Ekkehard Euler, and Nassir Navab. Towards a hybrid navigation interface: Comparison of a slice based navigation system with in-situ visualization. In Guang-Zhong Yang, Tianzi Jiang, Dinggang Shen, Lixu Gu, and Jie Yang, editors, *Medical Imaging and Augmented Reality*, volume 4091 of *Lecture Notes in Computer Science*, pages 179–186, Shanghai, China, Aug. 2006. Springer.
- [173] Roger Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.
- [174] Mihran Tuceryan, Douglas S. Greer, Ross T. Whitaker, David E. Breen, Chris Crampton, Eric Rose, and Klaus H. Ahlers. Calibration requirements and procedures for a monitor-based augmented reality system. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):255–273, September 1995.

- [175] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(4):376–380, 1991.
- [176] Allen Van Gelder and Kwansik Kim. Direct volume rendering with shading via three-dimensional textures. pages 23–30, 1996.
- [177] Florian Vogt. *Augmented Light Field Visualization and Real-Time Image Enhancement for Computer Assisted Endoscopic Surgery*. PhD thesis, Universität Erlangen-Nürnberg, 2005.
- [178] Sebastian Vogt, Ali Khamene, Frank Sauer, Andreas Keil, and Heinrich Niemann. A high performance AR system for medical applications. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, pages 270–271, October 2003.
- [179] Sebastian Vogt, Ali Khamene, Frank Sauer, and Heinrich Niemann. Single camera tracking of marker clusters: Multiparameter cluster optimization and experimental verification. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 127–136, 2002.
- [180] Sebastian Vogt, Frank Wacker, Ali Khamene, Daniel R. Elgort, Tobias Sielhorst, Heinrich Niemann, Jeff Duerk, Jonathan Lewin, and Frank Sauer. Augmented reality system for mr-guided interventions: Phantom studies and first animal test. In *Proceedings of SPIE, Volume 5367, Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display*, 2004.
- [181] Frank K. Wacker, Sebastian Vogt, Ali Khamene, John A. Jesberger, Sherif G. Nour, Daniel R. Elgort, Frank Sauer, Jeffrey L. Duerk, and Jonathan S. Lewin. An augmented reality system for mr image - guided needle biopsy: Initial results in a swine model. *Radiology*, 238(2):497–504, 2006.
- [182] Kongming Wang and Theo Gasser. Alignment of curves by dynamic time warping. *Annals of Statistics*, 25(3):1251–1276, 1997.
- [183] Kongming Wang and Theo Gasser. Synchronizing sample curves nonparametrically. *Annals of Statistics*, 27(2):439–460, 1999.
- [184] Matthew Y. Wang, Calvin R. Maurer, Jr., J. Michael Fitzpatrick, and Robert J. Maciunas. An automatic technique for finding and localizing externally attached markers in ct and mr volume images of the head. *IEEE Trans. Biomed. Eng.*, 43(6):627–637, 1996.
- [185] Colin Ware and Ravin Balakrishnan. Reaching for Objects in VR Displays: Lag and Frame Rate. *ACM Transactions on Computer-Human Interaction*, 1(4):331–356, 1994.
- [186] I. Wegner, M. Vetter, M. Schoebinger, I. Wolf, and H.P. Meinzer. Development of a navigation system for endoluminal brachytherapy in human lungs. In Kevin R. Cleary and Robert L. Galloway, Jr., editors, *Medical Imaging 2006: Visualization*,

Image-Guided Procedures, and Display, volume 6141 of *Proceedings of SPIE*, pages 23–30, March 2006.

- [187] Pierre Wellner, Wendy Mackay, and Rich Gold. Computer augmented environments: Back to the real world. *Commun. ACM*, 36(7):24–26, 1993.
- [188] Thomas Wendler, Marco Feuerstein, Jörg Traub, Tobias Lasser, Jakob Vogel, Farhad Daghighian, Sibylle Ziegler, and Nassir Navab. Real-time fusion of ultrasound and gamma probe for navigated localization of liver metastases. In N. Ayache, S. Ourselin, and A. Maeder, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 4792 of *Lecture Notes in Computer Science*, pages 252–260, Brisbane, Australia, October/November 2007. Springer-Verlag.
- [189] Thomas Wendler, Alexander Hartl, Tobias Lasser, Joerg Traub, Farhad Daghighian, Sibylle Ziegler, and Nassir Navab. Towards intra-operative 3d nuclear imaging: reconstruction of 3d radioactive distributions using tracked gamma probes. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2007. To appear at Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI).
- [190] Thomas Wendler, Joerg Traub, Sibylle Ziegler, and Nassir Navab. Navigated three dimensional beta probe for optimal cancer resection. In Rasmus Larsen, Mads Nielsen, and Jon Sporring, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 4190 of *Lecture Notes in Computer Science*, pages 561–569, Copenhagen, Denmark, Oct. 2006. MICCAI Society, Springer.
- [191] Christian Wengert, Philippe Cattin, John M. Duff, Charles Baur, and Gábor Székely. Markerless endoscopic registration and referencing. In R. Larsen, M. Nielsen, and J. Sporring, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, 2006.
- [192] S. Wesarg, E.A. Firle, B. Schwald, H. Seibert, P. Zogal, and S. Roeddiger. Accuracy of needle implantation in brachytherapy using a medical ar system: a phantom study. In *Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display. Edited by Galloway, Robert L., Jr. Proceedings of the SPIE*, volume 5367, pages 341–352, 2004.
- [193] James D. Westwood, Randy S. Haluck, Helene M. Hoffman, Greg T. Mogel, Roger Philips, and Richard A. Robb. *Medicine Meets Virtual Reality 12 - Building a Better You: The Next Tools for Medical Education, Diagnosis, and Care*, volume 12. IOS Press, 2004.
- [194] A.D. Wiles, D.G. Thompson, and D.D. Frantz. Accuracy assessment and interpretation for optical tracking systems. *SPIE Medical Imaging*, 5367, 2004.
- [195] Reg Willson. *Modeling and Calibration of Automated Zoom Lenses*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 1994.

- [196] T. Yamaguchi, M. Nakamoto, Y. Sato, K. Konishi, M. Hashizume, N. Sugano, H. Yoshikawa, and S. Tamura. Development of a camera model and calibration procedure for oblique-viewing endoscopes. *Computer Aided Surgery*, 9(5):203–214, 2004.
- [197] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.

AUTHORED AND COAUTHORED PUBLICATIONS

- [1] Seyed-Ahmad Ahmadi, Tobias Sielhorst, Ralf Stauder, Martin Horn, Hubertus Feussner, and Nassir Navab. Recovery of surgical workflow without explicit models. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 420–428, 2006.
- [2] Christoph Bichlmeier, Tobias Sielhorst, Sandro Michael Heining, and Nassir Navab. Improving depth perception in medical ar: A virtual vision panel to the inside of the patient. In *Bildverarbeitung für die Medizin (BVM)*. Springer, Mar. 2007.
- [3] Christoph Bichlmeier, Tobias Sielhorst, and Nassir Navab. The tangible virtual mirror: New visualization paradigm for navigated surgery. In *AMIARCS - The Tangible Virtual Mirror: New Visualization Paradigm for Navigated Surgery*, Copenhagen, Denmark, Oct. 2006.
- [4] Tobias Blum, Tobias Sielhorst, and Nassir Navab. Advanced augmented reality feedback for teaching 3d tool manipulation. In *New Technology Frontiers in Minimally Invasive Therapies*, chapter 25, pages 223–236. Lupiensis Biomedical Publications, Leece, Italy, 2007.
- [5] Sandro-Michael Heining, Philipp Stefan, Latifa Omari, Stefan Wiesner, Tobias Sielhorst, Nassir Navab, Frank Sauer, Ekkehard Euler, Wolf Mutschler, and Joerg Traub. Evaluation of an in-situ visualization system for navigated trauma surgery. *Journal of Biomechanics*, 39(Supplement 1):209, 2006.
- [6] A. Khamene, S. Vogt, F. Azar, T. Sielhorst, and F. Sauer. Local 3D reconstruction and augmented reality visualization of freehand ultrasound for needle biopsy procedures. In Randy E. Ellis and Terry M. Peters, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, pages 344–355. Springer-Verlag, 2003.
- [7] Gudrun Klinker, Hesam Najafi, Tobias Sielhorst, Fabian Sturm, Florian Echtler, Mustafa Isik, Wolfgang Wein, and Christian Truebswetter. Fixit: An approach towards assisting workers in diagnosing machine malfunctions. In *Proc. of the International Workshop exploring the Design and Engineering of Mixed Reality Systems - MIXER 2004, Funchal, Madeira, CEUR Workshop Proceedings*, 2004.

- [8] Nassir Navab, Joerg Traub, Tobias Sielhorst, Marco Feuerstein, and Christoph Bichlmeier. Action- and workflow-driven augmented reality for computer-aided medical procedures. *IEEE Computer Graphics and Applications*, 27(5):10–14, September/October 2007.
- [9] T. Sielhorst, S.-A. Ahmadi, R. Stauder, M. Horn, and H. Feussner and N. Navab. Towards automatic recovery of surgical workflow. In *5. Jahrestagung der Deutschen Gesellschaft für Computer-und Roboter-Assistierte Chirurgie (CURAC 2006)*, Hannover, Germany, September 2006 2006.
- [10] T. Sielhorst and N. Navab. Ein augmented reality system zur medizinischen ausbildung. In *3. Jahrestagung der Deutschen Gesellschaft für Computer-und Roboter-Assistierte Chirurgie (CURAC 2004)*, München, Germany, September 2004 2004.
- [11] Tobias Sielhorst, Martin A. Bauer, and Nassir Navab. Online estimation of the target registration error for n-ocular optical tracking systems. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2007. To appear at Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI).
- [12] Tobias Sielhorst, Christoph Bichlmeier, Sandro Michael Heining, and Nassir Navab. Depth perception a major issue in medical ar: Evaluation study by twenty surgeons. In R. Larsen, M. Nielsen, and J. Sporring, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, 2006.
- [13] Tobias Sielhorst, Tobias Blum, and Nassir Navab. Synchronizing 3d movements for quantitative comparison and simultaneous visualization of actions. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, 2005.
- [14] Tobias Sielhorst, Marco Feuerstein, Joerg Traub, Oliver Kutter, and Nassir Navab. Campar: A software framework guaranteeing quality for medical augmented reality. *International Journal of Computer Assisted Radiology and Surgery*, 1(Supplement 1):29–30, June 2006.
- [15] Tobias Sielhorst and Alexander Horsch. An efficient wavelet compression of ct and mrt images in an online textbook. In *Bildverarbeitung für die Medizin (BVM)*, 2002.
- [16] Tobias Sielhorst and Nassir Navab. Advanced visualization in the operating room. *Med and Lab Engineering Magazine*, 6(72):6–7, April 2007.
- [17] Tobias Sielhorst, Tobias Obst, Rainer Burgkart, Robert Riener, and Nassir Navab. An augmented reality delivery simulator for medical training. In *International Workshop on Augmented Environments for Medical Imaging - MICCAI Satellite Workshop*, 2004.
- [18] Tobias Sielhorst, Ralf Stauder, Martin Horn, Thomas Mussack, Armin Schneider, Hubertus Feussner, and Nassir Navab. Simultaneous replay of automatically synchronized videos of surgeries for feedback and visual assessment. *International Journal of Computer Assisted Radiology and Surgery, Supplement 1*, 2:433–434, June 2007.

- [19] Tobias Sielhorst, Jörg Traub, and Nassir Navab. The ar apprenticeship: Replication and omnidirectional viewing of subtle movements. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR)*, 2004.
- [20] Philipp Stefan, Joerg Traub, Sandro-Michael Heining, Christian Riquarts, Tobias Sielhorst, Ekkehard Euler, and Nassir Navab. Hybrid navigation interface: a comparative study. In H. Handels H.-P. Meinzer T. Tolxdorf A. Horsch, T. M. Deserno, editor, *Bildverarbeitung für die Medizin (BVM)*, pages 81–86. Springer-Verlag, 25-27 March 2007.
- [21] N. Navab T. Sielhorst, T. Blum. Quantitative comparison of movements for medical training. In *4. Jahrestagung der Deutschen Gesellschaft für Computer-und Roboter-Assistierte Chirurgie (CURAC 2005)*, Berlin, Germany, September 2005 2005.
- [22] Joerg Traub, Philipp Stefan, Sandro-Michael Heining, Tobias Sielhorst, Christian Riquarts, Ekkehard Euler, and Nassir Navab. Hybrid navigation interface for orthopedic and trauma surgery. In R. Larsen, M. Nielsen, and J. Sporring, editors, *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 4190 of *Lecture Notes in Computer Science*, pages 373–380, Copenhagen, Denmark, Oct. 2006. MICCAI Society, Springer.
- [23] Joerg Traub, Philipp Stefan, Sandro-Michael Heining, Tobias Sielhorst, Christian Riquarts, Ekkehard Euler, and Nassir Navab. Stereoscopic augmented reality navigation for trauma surgery: cadaver experiment and usability study. *International Journal of Computer Assisted Radiology and Surgery*, 1(Supplement 1):30–31, June 2006.
- [24] Joerg Traub, Philipp Stefan, Sandro-Michael Heining, Tobias Sielhorst, Christian Riquarts, Ekkehard Euler, and Nassir Navab. Towards a hybrid navigation interface: Comparison of a slice based navigation system with in-situ visualization. In Guang-Zhong Yang, Tianzi Jiang, Dinggang Shen, Lixu Gu, and Jie Yang, editors, *Medical Imaging and Augmented Reality*, volume 4091 of *Lecture Notes in Computer Science*, pages 179–186, Shanghai, China, Aug. 2006. Springer.
- [25] Sebastian Vogt, Frank Wacker, Ali Khamene, Daniel R. Elgort, Tobias Sielhorst, Heinrich Niemann, Jeff Duerk, Jonathan Lewin, and Frank Sauer. Augmented reality system for mr-guided interventions: Phantom studies and first animal test. In *Proceedings of SPIE, Volume 5367, Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display*, 2004.