

Augmented Reality for Exterior Construction Applications

Gudrun Klinker (1), Didier Stricker (2), Dirk Reiners (2)

(1) Technische Universität München,

Lehrstuhl für angewandte Softwaretechnologie

Arcisstr. 21, D-80290 München

(2) Fraunhofer Projektgruppe für Augmented Reality am ZGDV,

Rundeturmstr. 6, D-64283 Darmstadt, Germany

Abstract

Augmented reality (AR) constitutes a very promising new user interface concept for many applications. In this chapter, we pay particular attention to developing AR technology for exterior construction applications, augmenting video sequences from construction sites with information stored in models. Such augmentations can tremendously benefit several business processes common to many construction projects.

We are focussing on two approaches to augment the reality of construction sites. The first one augments video sequences of large outdoor sceneries with detailed models of prestigious new architectures, such as TV towers and bridges. Since such video sequences are very complex, we currently pre-record the sequences and employ off-line, interactive techniques. The second approach operates on live video streams. To achieve robust real-time performance, we need to use simplified, “engineered” scenes. In particular, we place highly visible markers at precisely measured locations to aid the tracking process.

1 Introduction

Augmented reality (AR) constitutes a very promising new user interface concept for many applications. Currently, we pay particular attention to developing AR technology for exterior construction applications. In the context of the European CICC-project [10], we develop and evaluate the potential of AR in a series of pilot projects, augmenting video sequences from construction sites with information stored in models. Such augmentations can tremendously benefit several processes common to many construction projects.

- **Design and Marketing:** Creating a design and evaluating it for function and esthetics, and showing a customer what a new structure will look like in its final setting. AR provides the unique opportunity to integrate the design into the real world context.
- **During Construction:** Visualization whether an actual structure is built in accordance with the design; quick update of work plans after a design change; visualization of consequences of potential design changes before they are agreed upon.
- **Maintenance and Renovation:** Visualization of hidden information (wires, pipes, beams in a wall); visualization of non-graphical information (heat and pressure of pipes, maintenance schedules and records); visualization of potential redesigns (interior, exterior) to evaluate their compatibility with existing structures, and placement of new structures onto/into preexisting buildings.

Some of these benefits can also be partially achieved with other graphical approaches, such as Virtual Reality presentations. The level of realism, however, that can potentially be achieved with AR-systems far surpasses VR which seems to asymptotically narrow the gap between synthetic models and the real world (see Figure 1). AR, on the other hand, starts with the real world, augmenting it as little or much as is deemed suitable for the task at hand [35].¹

¹ Sources of all graphical material are listed at the end of this chapter.

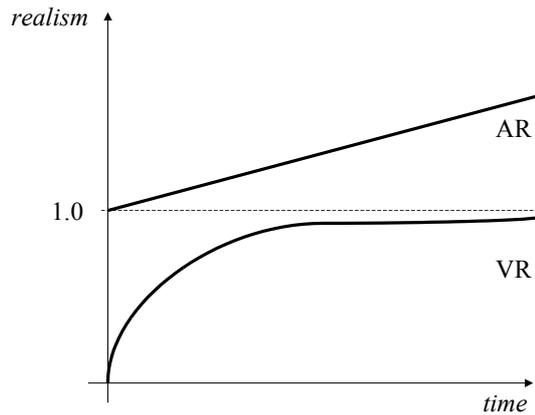


Figure 1: Potential Realism of AR vs. VR approaches

This gain in realism is coupled with a potential gain in speed since the real environment doesn't have to be rendered but merely mixed with a (much smaller) virtual model. The price to pay is the effort to strive for perfect alignment between the real and the virtual world. Assuming that this can be achieved satisfactorily in real-time in the foreseeable future, the overall speed gain by not having to synthesize the real world will be considerable.

1.1 AR challenges and chances in exterior construction applications

Exterior construction applications impose very demanding challenges on the robustness and usability of evolving AR technologies.

- First of all, the mere size of a large construction project (e.g., a bridge, a tower, a shopping mall or an airport) is overwhelming. The synthetic data is huge and needs special processing technology. To present such wealth of information in real time, the data needs to be reduced and simplified. Concepts such as level of detail and relevance need to be developed with respect to the task at hand. Furthermore, much information is currently only represented in two dimensions. Tools to translate it into a three-dimensional context are necessary. To access all data when and wherever necessary, the system depends on a very good computer infrastructure, including fast and mobile networks, computers, and data repositories.
- Second, the size of not just the synthetic data model but also of the real site imposes problems. Users manoeuvre in a very large space. Some AR devices, such as magnetic trackers or overhead surveillance cameras are rather geared towards indoor applications and unlikely to operate well under such conditions.

GPS, on the other hand, optical tracking techniques and inertial sensors have the potential to fare well. But they must be able to cope with situations, when only partial information or only a local view of the entire construction site is available. Exterior construction scenarios thus require more tracking skills than what is currently shown in table-top demonstrations [55, 60, 65, 68].

- Third, AR applications require a very accurate model of the current site (a *reality model*) both to determine the current camera position and to augment the current view realistically with synthetic information (the *virtual model*). Realistic immersion of virtual objects into a real scene requires that the virtual objects behave in physically plausible manners. They occlude or are occluded by real objects, they are not able to move through other objects, and they cast shadows on other objects. To this end, AR-systems need geometrically precise descriptions of the real environment. Yet, construction environments are not well structured. Natural objects such as rivers, hills, trees, and also heaps of earth or construction supplies are scattered around the site. Typically, no exact detailed 3D information of such objects exists making it difficult to generate a precise model of the site. Even worse, construction sites are in a permanent state of change. Buildings and landscapes are demolished, new ones are constructed. People and construction equipment move about, and the overall conditions depend on the weather and seasons. AR applications thus need to identify suitable approaches for generating and dynamically maintaining appropriate models of the real environment.
- It is also important to decide upon the appropriate level of realism with which virtual objects are rendered into the real world. For safety reasons, construction workers need to have and maintain a very clear understanding of the real objects and safety hazards around them. Virtual objects must not decrease people's awareness of danger, e.g., by perfectly adding virtual floors and walls to the bare wireframe of beams of the next floor being built in a high-rise.

In other situations, however, the highest level of realism is highly desirable, e.g., when visualizing whether a designed object will integrate well into an existing landscape.
- In addition to augmenting reality, exterior construction scenarios also need tools to diminish reality, since in most cases, objects and landscapes are removed or changed before new ones are built. Thus, techniques for synthetically removing real objects from the incoming video input stream need to be developed.

Despite such challenges, exterior construction is a very suitable application area for AR. Construction, in its very nature, is very much a three-dimensional activity. Business practices and work habits are all oriented towards the design, comprehension, visualization and realization of 3D plans. Workers are used to graphical descriptions

such as 2D plots. Much information is already represented and communicated in graphical form. Thus, new graphical user interfaces like AR fit very naturally into current work practices.

Furthermore, gathering high-precision geodesic measurements of selected points on a construction site and marking them in suitable ways is a well-established practice. Large construction sites use a wealth of high precision equipment such as theodolites, differential GPS and laser pointers, that AR can build upon. *Engineering the environment* to suit the current capabilities of the technology is acceptable within limits. Thus, AR can begin by building applications that simplify many of the general challenges, adapting the construction site to suit their skills. Over time more sophisticated and general approaches can be developed.

1.2 Our approach

Figure 2 illustrates our current framework for augmenting images of the real world with virtual objects. The AR-viewer takes four kinds of input (shown in the darker, rounded rectangles): virtual object models to be visualized or rendered, a photo or an image sequence to which the virtual objects are added, camera positions to facilitate seamless integration, and a reality model to enable physically correct coexistence of virtual and real objects.

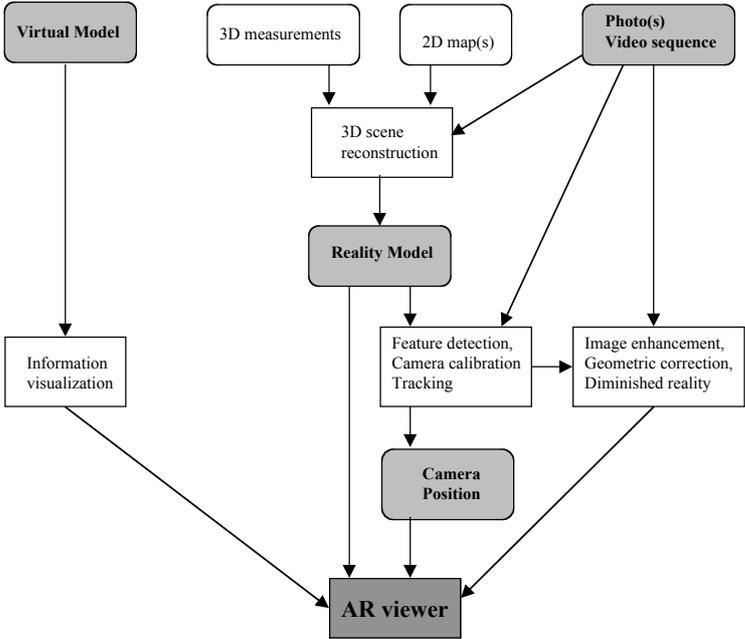


Figure 2: Conceptual framework of an AR-system.

The white, sharp rectangles show our information processing tools: We use a wealth of both commonly available and special purpose visualization and graphical rendering schemes to present the synthetic (virtual) information (section 4). We have also developed various interactive 3D scene reconstruction techniques to generate reality models from available information, such as photos, maps or 3D measurements (section 2). Third, we use various interactive or automatic techniques to calibrate and track cameras for live or prerecorded image sequences, using features that are specified in the reality model (section 3). Fourth, we are aware of the principle need to synthetically alter the image prior to its display, e.g., to correct for lens distortions or to remove objects from the scene. Yet, we haven't approached the subject in much depth yet (section 5). All original or processed information flows into the AR-viewer where the final three-dimensional integration of real and virtual objects is generated (section 4), ready for the user interact with in various ways (section 6).

Within this framework, we are focussing on two approaches to augment the reality of construction sites.

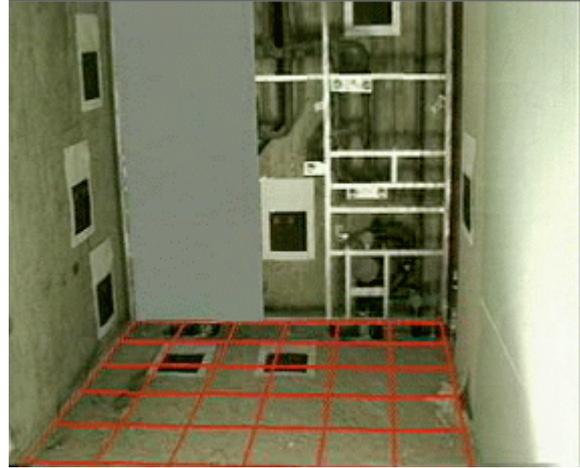
- The first approach augments video sequences of large outdoor sceneries with detailed models of prestigious new architectures, such as TV towers and bridges that will be built to ring in the new milleneum (see Figure 3a). Since such video sequences are very complex, we currently pre-record the sequences and employ off-line, interactive calibration techniques to determine camera positions. Given all calibrations, the augmentation of the images with the virtual object is performed live, i.e., the virtual model can be altered and transformed while it is being seen in the video sequence.
- The second approach operates on live video streams, calibrating and augmenting images as they come in. To achieve robust real-time performance, we need to use simplified, "engineered" scenes. In particular, we place highly visible markers at precisely measured locations to aid the tracking process (Figure 3b).

As indicated by these figures and example applications, we focus on only a subset of the challenges posed by exterior construction scenarios. We currently use rather pragmatic simplified or semi-interactive approaches, expecting that future developments will provide more automatic and general solutions.

Furthermore, we focus on the graphical aspects of AR. A full AR system also requires sound and other multi-media interfaces, as well as a complex computer and network infrastructure, such as distributed, mobile, wireless and wearable computing, to make relevant information and processing power available where the user happens to go [43,59]. Such aspects of AR are discussed in other chapters of this book.



(a) Virtual bridge across a real river.



(b) Virtual wall and grid in a real room.

Figure 3: Interactive vs. automatic video augmentation.

1.3 Related work and current state of the art

The first chapters of this book and recent surveys have provided excellent general overviews of AR and its young history [3,6,44]. We will focus here on topics closely related to our approach.

For the interactive augmentation of landscapes (Figure 3a), our approach is most closely related to photomontaging jobs performed by professional photo labs. Selected still photos are augmented to illustrate to the public how new construction projects, such as placing the Munich railroad station underground, will improve the city. The process is currently very tedious and time consuming; the results are individual, unalterable augmentations of individual photos. Using our approach, the understanding (calibration) of the image is decoupled from its augmentation. Thus, augmentations can be altered at interactive speed; innumerable different versions of the virtual objects can be integrated into the photo. Furthermore, the approach works for entire video loops; not just for still photos.

For the automatic augmentation of live video streams (Figure 3b), several research groups have begun exploiting the use of special targets in the scene for optical tracking [4,31,33,41,42,48,59,60,66]. Experiments indicate that optical approaches provide higher precision than non-optical ones. Ideally, they are combined in hybrid approaches with other, non-optical techniques [60]. Yet, since each group uses different targets and equipment, it

is unclear which approach works best. So far, no standardized test scene has been shared between several groups.

Little research has focussed so far on architectural applications. Feiner's group is exploring approaches to improve the construction, inspection and renovation of architectural structures with AR [68], focussing on space frame constructions. Bajura and Neumann augment a toy house with a virtual antenna and an annotation arrow [4]. Debevec et al, as well as Faugeras et al, have proposed technologies to semi-automatically generate architectural models from images [12,14].

2 Reality models

In order to perfectly mix virtual and real objects, AR-systems need to calibrate cameras and other sensing and display equipment so that the virtual objects are rendered from the same vantage point as the real objects. Realistic immersion of virtual objects into a real scene further requires that the virtual objects behave in physically plausible manners, i.e.: they occlude or are occluded by real objects, they are not able to move through other objects, and they are shadowed or indirectly illuminated by other objects while also casting shadows and mirror images themselves.

For optical camera calibration and to enforce physical interaction constraints between real and virtual objects, augmented reality systems need to have a precise description of the physical scene: a *reality model*.

2.1 Required complexity of reality models

AR reality models don't need to be as complex as, for example, VR models. VR models are expected to synthetically provide a realistic immersive impression of reality. Thus, the description of photometric reflection properties and material textures is crucial. AR, on the other hand, can rely on live optical input to provide a very high sense of realism. The reality model only needs to indicate geometric properties, such as easily identifiable

landmarks in the scene for camera calibration, and surface shapes for occlusion handling and shadowing between real and virtual objects.

However, AR reality models have to be much more precise than VR models. Since an immersive VR system cuts users off from reality, users can only gain a qualitative impression whether or not the objects are modelled correctly. In AR, on the other hand, users have an immediate quantitative appreciation of the extent of mismatches between the reality model and the live video input from the real scene.

A reality model has to track and adapt to changes in the real world. The need and frequency of model updates depends on the application. Many durable large-scale structures will remain in place during the entire construction work. Thus, such components need to be modeled only once. Using existing CAD-models or semi-automatic modeling techniques to generate such models may be sufficient. Other aspects of construction sites are more variable: trucks and cranes move, trees lose their leaves in winter time, and buildings under construction slowly develop. Such gradual changes need to be mirrored in the reality model at an appropriate pace. In some applications, daily or real-time scene changes may also have to be modelled, e.g: when virtual objects have to be integrated into scenes with moving people, material or equipment.

Reality modelling and *reality tracking* are very complex and demanding tasks. Currently they cannot be achieved in real-time in a general way. The subsequent sections present and discuss several approaches to generate reality models.

2.2 Use of existing models

The most straightforward approach to acquiring 3D scene descriptions is to use existing geometric models, such CAD data, output from GIS systems, and maps (see Figure 5). When such models are available, they constitute the easiest approach towards integrating virtual objects into the real world. Yet, this approach is not always pursuable for a number of reasons.

- In many applications, reality models are not commercially available. For example, interior restoration of old buildings typically needs to operate without pre-existing CAD data.

- The data points in a commercial model don't necessarily coincide with well visible features in images; quite the opposite is true: geodesic measurements generally are indicated by small, barely visible marks in the ground.
- Available models are not complete. Real physical objects typically show more detail than is represented in the models. Furthermore, scene models cannot fully anticipate the occurrence of non-stationary objects, such as coffee mugs on tables and cars or cranes on construction sites.
- The system needs to account for the changing appearances of existing objects, such as buildings under construction or engines that are partially disassembled.

When users see objects in the scene, they expect the virtual objects to interact with them correctly, independently of whether they are new to the scene or whether they have been there for a long time (i.e.: have already been included in a reality model). Thus, it currently is often necessary to create and update reality models explicitly for the AR application.

2.3 Manual approach

The manual approach involves obtaining 3D measurements within the real world, using measuring tapes, theodolites, GPS-operated laser pointers, information from GIS systems, etc. Such 3D points are entered into a small model, which in turn can be used to calibrate and track a camera by tracking the corresponding image points. Figure 4 shows a thus generated reality model of our "tracking laboratory", a room with several carefully measured targets on its walls.

The approach is intuitive and works well for very sparse reality models. Yet, it is prohibitively expensive to measure thousands of points this way. Furthermore, the approach depends upon availability of professionals and special equipment. Thus, models cannot be expected to be obtainable on short notice.

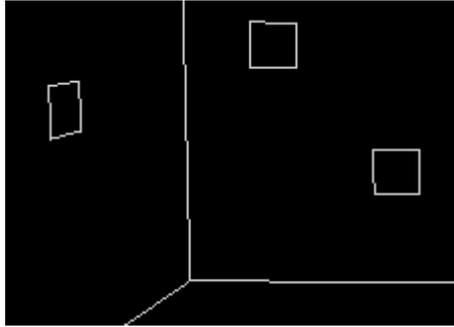


Figure 4: Example and use of a manually created reality model.

2.4 Interactive approach

As an alternative, reality models can be generated with interactive graphical tools.

We have developed a system, InCal, which begins with a very sparse, initial reality model of a landscape or cityscape, using externally provided information such as the known position and height of a few buildings, electric power poles and bridge pillars. More information, such as the course of rivers and streets, is measured from two-dimensional maps and inserted at zero height into the model. From this model, we generate an initial camera calibration for a few site photos, interactively indicating how features in the image relate to the model.

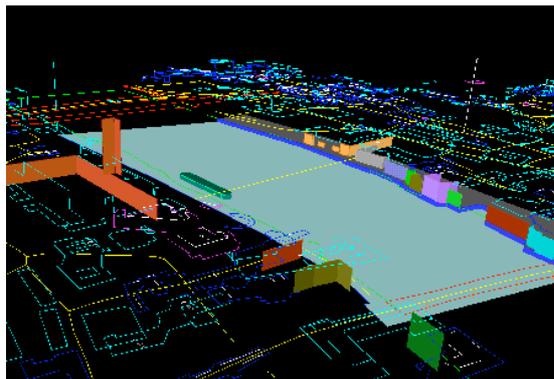
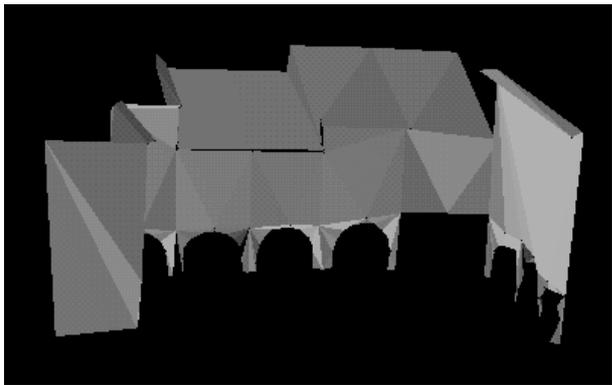


Figure 5: Final interactive model superimposed on the commercial model of the city of London.

Once an image has been successfully calibrated (3.2), the model is overlaid on the image, showing good alignment of the image features with the model features. Models of new structures in the landscape, such as houses or hills can then be entered into the reality model, using their two-dimensional position in the city map and estimating their height from their alignment with the image. Figure 5 shows the final model, compared with a commercially available model of the same area.

2.5 Towards automatically generated models

Computer vision techniques are designed to automatically acquire three-dimensional scene descriptions from image data. Much research is currently under way, exploring various schemes to optically reconstruct a scene from multiple images, such as structure from motion [1,14,63,67], (extended) stereo vision [12,26,28,29,45], and photogrammetric techniques [21,23].



(a) Geometric model.



(b) Enhanced with texture maps.



(c) 2D picture of the plaza, augmented with a Ferrari.FFi

Figure 6: Automatically generated model of the arcades of Valbonne.

In the context of the European Realise project [22] and its successor, Cumuli [11], we explore to what extent automatically generated scene models can support AR and VR applications. In collaboration with INRIA and Lund University, we are developing and testing tools to semi-automatically generate descriptions of complex landscapes and cityscapes, such as parts of London along the Thames. Using epipolar relationships between features seen in several images from different unknown vantage points, geometric constraints on architectural structures, as well as city maps, the tools will help determine a set of progressively more precise projective, affine and finally euclidean properties of points in the three-dimensional scene.

Figure 6 shows the final result of the Realise project, a reconstructed model of the Arcades of Valbonne. Figure 6a shows the reconstructed model. In Figure 6b, the model has been enhanced by mapping textures from the original image data onto the surfaces. Figure 6c illustrates how the original image data can be augmented with synthetic objects, such as a Ferrari, once the images have been analyzed with the Realise system.

2.6 Range data models

An interesting alternative to motion-based scene recognition are approaches using three-dimensional range sensors, undertaken in efforts such as the RESOLV project [53]. RESOLV is developing a mobile robot (Figure 7a) to conduct a 3D survey of a building, including capturing the appearance of the visible surfaces. A portable unit is taken around the environment that is to be captured. The unit includes a spatial camera constituted of a scanning laser rangefinder for capturing the 3D structure of the surroundings and a video camera for adding the textures.

The environment is scanned from a number of capture positions and reconstructed into a model, unifying measurements from all viewing positions. Surfaces are recognised by processing the range data and are textured from the camera images. By combining what is seen from neighbouring capture positions, surfaces that would be occluded from one position are recorded. The spatial camera travels from one capture position to another either on a trolley (Figure 7) or an autonomous vehicle. The environment is reconstructed as the robot progresses and each new position is registered with previous ones using key points in the surroundings. The partial reconstruction is used to determine future capture positions.



(a)



(b)

Figure 7: a) RESOLV trolley b)Automatically generated model of part of the interior of the Royal Institute of Chartered Surveyors, London.

Figure 7b shows a model taken from the interior of the Royal Institute of Chartered Surveyors, London. Of particular note is the fire extinguisher in one corner, and the curved surface of the pillar, showing the usefulness and accuracy of the laser/video combination.

Currently, the robot is optimised for human scale applications such as indoor refurbishment or maintenance tasks, and not yet designed for capturing external landscapes. Both the trolley shown in the picture and the autonomous vehicle are designed to support capture at two heights - eye level when sitting and standing. The size of the unit is comparable to that of a person to ensure that it can be taken to all the places where people are likely to pause when looking around a building. The data is held in a form suitable for CAD systems and for viewing on a WWW browser -- which is also a suitable format for reality models of AR applications.

3 Camera calibration

One of the key issues of AR is the proper alignment of the virtual world with the real world. Such alignment requires that the view onto the real world be accurately determined and simulated for the virtual scene. Since we concentrate in our approach on viewing the real world through a TV-camera, proper alignment means calibrating

a camera, determining 5 internal (focal length (f_x, f_y) , center (c_x, c_y) , aspect ratio a) and 6 external parameters (position (x, y, z) and orientation (r_x, r_y, r_z)). We do not yet account for lens distortions.

We are experimenting with two different approaches, depending on various application scenarios. Both approaches proceed in two steps, a) the determination of suitable feature points in an image and the establishment of proper correspondences between such 2D image features and 3D scene features in the reality model, and b) the computation of the current camera parameters according to the matches. These two steps, as well as sensor fusion concepts and precision and stability issues are now discussed.

3.1 Correspondence between the Reality Model and 2D image points

3.1.1 Live automatic mapping of specific targets

To achieve live camera tracking performance, feature detection has to operate fast and automatically. The automatic detection of three-dimensional objects in images is a long standing research area in computer vision. So far it cannot be achieved in a general way in real time. To provide fast and robust optical tracking performance, it is common practice in AR applications to simplify the computer vision problem by placing special, easily detectable target patterns into the real world [4,31,41,42,59,60,66]. For example, Neumann et al. use circular tracking targets [48]. State et al. use concentric multi-colored disks [60]. Bajura and Neumann [4] track bright red LEDs which are significantly brighter than the other objects in the environment and thus can be easily detected. Starner et al. search the environment for visual tags consisting of two red squares bounding a binary pattern of green squares to identify objects of special interest [59]. It is also very common in the car industry to attach special black-and-yellow circular patterns with an internal cross to cars to evaluate car crash tests.

In our approach, we use black squares attached to a planar object with sufficient contrast. In order to uniquely identify each square independently of the current field of view, the black squares contain a labeling region, consisting of 2 rows with 4 positions (bits) for smaller red squares (see Figure 8). Using a binary encoding scheme, we can define up to 256 different targets, each of which can be matched against a 3D target in a reality

model. In a particular image, any set of two or more targets from the model suffice for our tracking system to work.

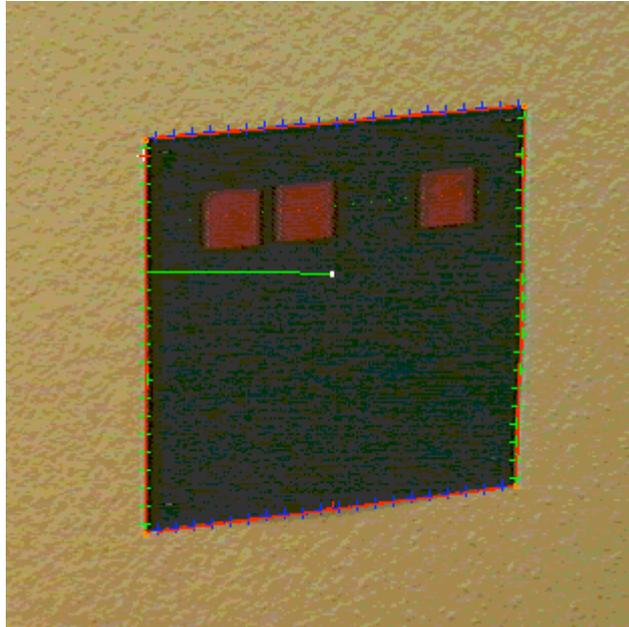


Figure 8: Processed image of a black target square 13 (binary label 1101)

At startup time, the optical tracker does not yet have any indication of the viewing direction. Thus, the entire image has to be searched quickly for targets. In a subsampled image, we begin by searching for candidate "blobs", scanning sample lines for strong bright-to-dark and subsequent dark-to-bright transitions. We then follow the contours of each blob. We classify the edge pixels according to their gradient direction as belonging to one of four edge classes, and fit straight lines to the edges of each class. The intersections of neighboring lines determine the corner points of candidate squares. The algorithm then examines the labeling area within each square, correlating the pixels along sampling lines of the first and second 4-bit row with the ideal binary signal of numbers 0 through 15. The number producing the highest correlation with the image is selected, and the candidate square label is compared against the list of 3D squares in the reality model. If the label exists and is assigned to exactly one image square, a match between the 2D and 3D square is established. To account for camera rolls by more than 90 degrees, we apply the same labelling test along all 4 edges of the square, selecting the labelling with the best match.

Subsequent images do not have to be searched from scratch to find the squares. Rather, tracking algorithms can predict the approximate location of squares from their locations in previous images (see section 3.4). To

determine the exact position of each square, we find strong image gradients in the vicinity of their predicted edge positions. We fit lines to edge pixels, and intersect them to determine the corner points of the squares in the new image.

This technique is fast (23 frames per second on an SGI O2, 14 frames on an Indy with an R5000 processor) and robust over quite a range of moderately fast camera motions. Typically, we move the camera on a tripod with wheels. Yet, we can also track an IndyCam in our hand. When the camera moves very fast or is jerked, the predicted square positions are incorrect, and/or the image exhibits motion blurring. Under such circumstances, the square redetection algorithm fails, and the square detection system is reinitialized in a third of a second.

3.1.2 Interactive mapping of arbitrary targets

Although automatic target tracking can be demonstrated to operate well in small-scale, engineered environments, it imposes significant restrictions on applications. In many cases, special targets cannot be placed in the scene, or the size of the scene is so large that targets at some distance are barely visible in the images. Thus, more general feature detection approaches need to be investigated. Yet, it will take time for them to mature and to become fast enough for real-time AR applications.

On the other hand, quite a few scenarios for exterior construction applications such as project acquisition and design efforts can already benefit from much slower, off-line, photo and video film augmentation. Under such circumstances, we can employ human help and analyze much more general scenes without special targets.

Our interactive calibration system, InCal, provides a user interface to calibrate and track camera positions in images. It superimposes a 3D reality model on an image which the user can move and rotate interactively. Furthermore, the user can interactively indicate correspondences between three-dimensional model features and pixels in the image (see Figure 9). Such correspondences are then used to automatically compute the current camera position. When the virtual camera is set to the same position, the reality model "snaps into alignment" with the image (see Figure 10).

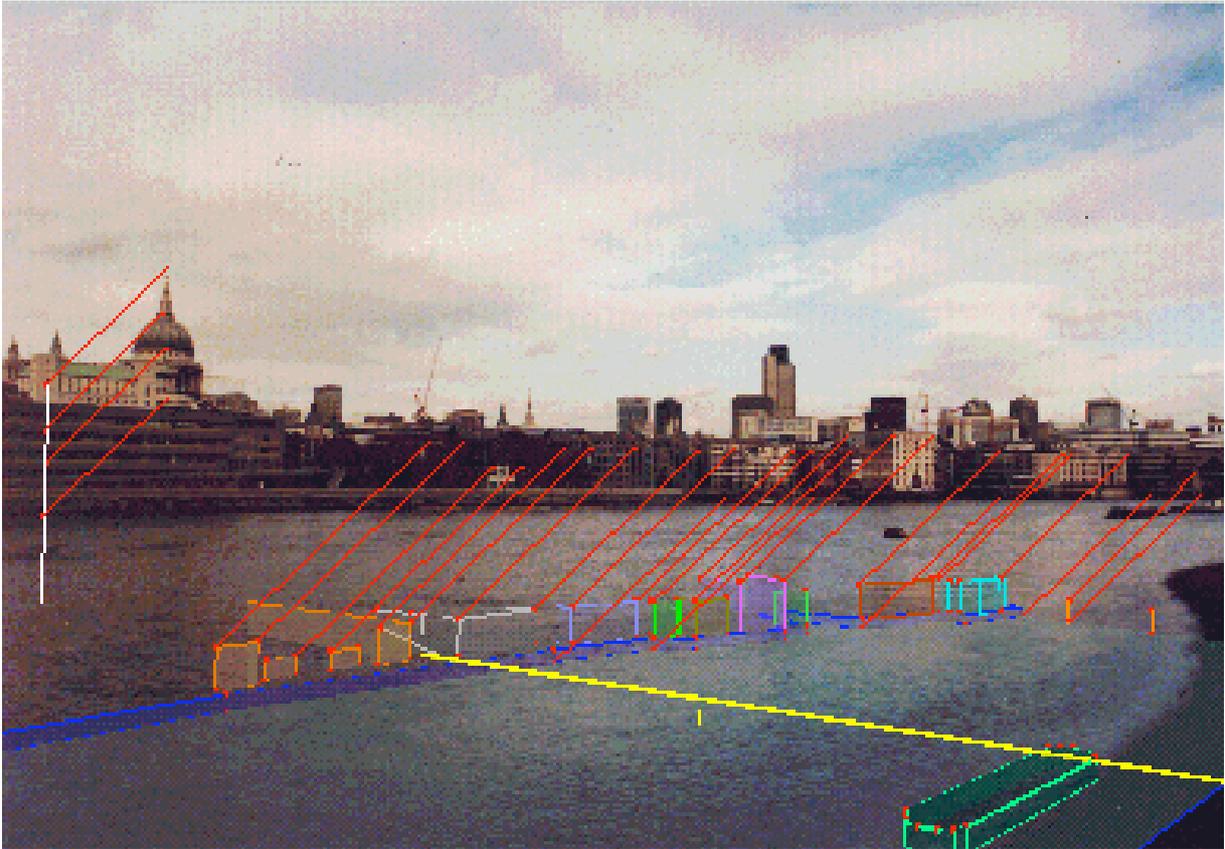


Figure 9: Mapping of selected 3D model features to 2D image features.

When calibrating image sequences, InCal exploits inter-frame coherence to automatically propose feature locations in new images from their locations in previous images. For user-scalable rectangular template areas around each feature in the current image, InCal uses normalized cross correlation to determine with sub-pixel precision the best match between a template area in the current image and pixels within a search window of the next image. Such automatic feature tracking substantially helps the user in working through a long image sequence. Many features are well detected. Occasional mismatches can be corrected interactively.

3.2 Calibration

Once correspondences between 3D scene features and 2D image features have been established, they can be used to determine the camera position. Many camera calibration algorithms have been developed in various computer vision research efforts (see [64,72] for reviews). The principle problem is well understood: Given a number of matches between 2D and 3D points, compute the camera viewing parameters that minimize the distance between

the image points and the projected position of their matched 3D world features. Since the system of equations is not linear, much effort has been spent investigating various approaches for finding stable solutions.

We have worked with two approaches in particular, one developed by Wenig [65,71] and one by Tsai [64,72]. Weng's algorithm computes all 11 intrinsic and extrinsic camera parameters. It works only for non-coplanar arrangements of features in the real world. Tsai's algorithm consists of a collection of calibration routines, geared towards computing different subsets of the camera parameters. Thus, different versions of the algorithm compute either all parameters or only the extrinsic ones, assuming that the focal length, center and aspect ratio are known. The simplified version works also with coplanar 3D feature arrangements. The overall approach begins by adjusting the camera rotation parameters to reduce the misalignment between the world and the image as much as possible. Next, camera translation is explored as a means to further reduce the alignment error. Finally, all six parameters are jointly reconsidered and optimized, using a non-linear least squares routine to minimize the error.

From Tsai's collection of calibration tools, we have distilled a number of further simplified approaches, assuming that in real applications even some of the extrinsic camera parameters can often be approximated by other means. In our approach, either of the first two steps of Tsai's algorithm (estimation of rotation and translation) can be skipped assuming externally provided data to initialize the non-linear least squares optimization. The result is a much more stable system that can be adapted quickly to incorporate various external sources of information.

3.3 Precision and repeatability of calibration results

Calibration precision is a key issue in AR since it determines the quality and credibility of augmented video [27].

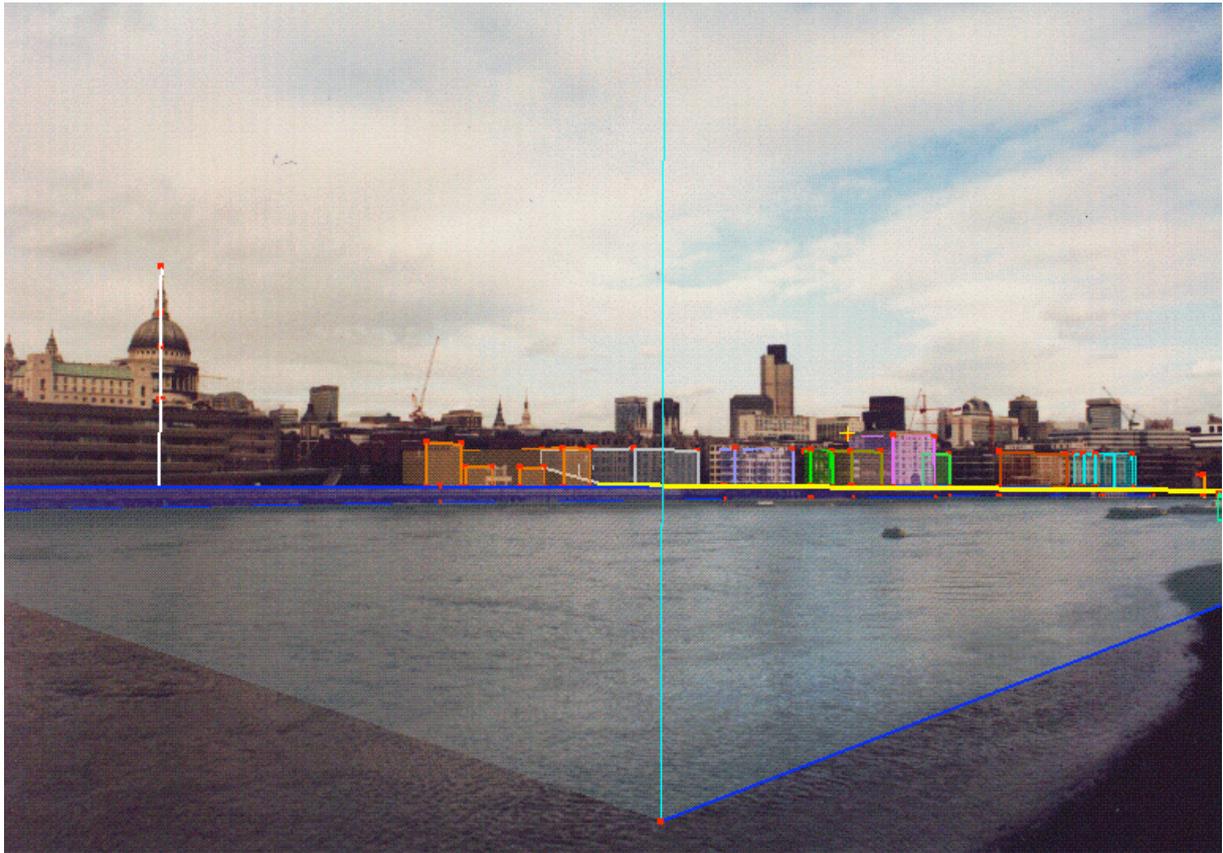
We have been able to successfully calibrate many live and pre-recorded video sequences. Even for the case of complex landscapes, such as from the river Wear in Sunderland, UK, we have been able to interactively calibrate sequences of hundreds of images nearly automatically within a few hours (e.g., the sequence containing Figure 13).

Yet, calibration algorithms are inherently sensitive to noise and to specific properties of the reality model, such as nearly planar or linear groupings of 3D features. We will now report on the more challenging cases. Figure 10 and Figure 11 illustrate the difficulties we had augmenting photos of the Thames river shore with a new footbridge. For this particular scene, most of the visible houses are nicely aligned along the river. Thus, most targets, such as house corners lie approximately within a plane. Considering a distance of approximately 400 meters between the camera and the houses, the depth provided within house facades and even the depth of individual houses cannot supply good 3-dimensional depth cues for any calibration algorithm.

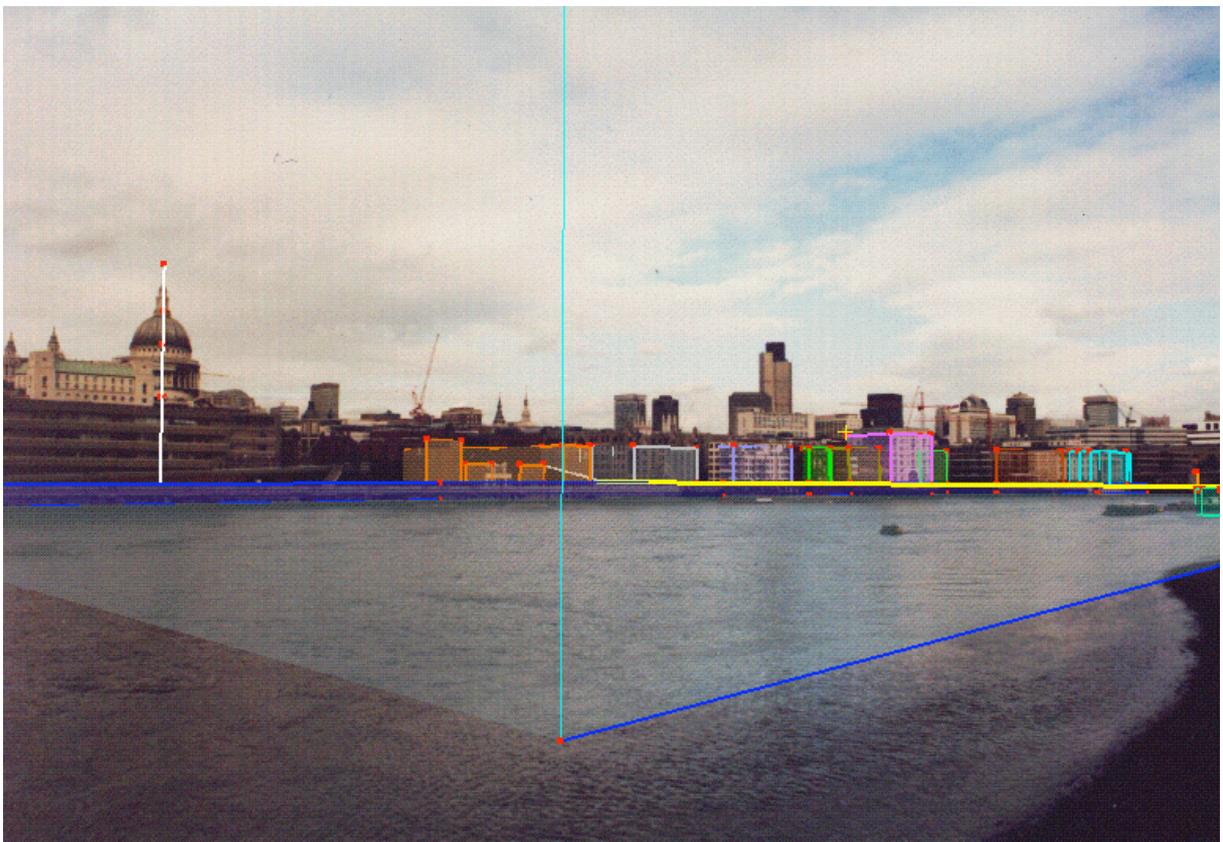
Figure 10 shows that at such camera distance and at the given image resolution slightly different 2D-to-3D mappings -- as the result of imprecise user input -- have a dramatic effect on the calibration result. In particular, there is a trade-off between the focal length of the camera and the distance of objects from the camera. Small mismatches of features along the line of sight can dramatically change the inferred focal length, altering the perspective appearance (vanishing points) of virtual objects without greatly misaligning their silhouette in the image. Between Figure 10a and b, one image feature at the back side of one of the houses is moved up by one pixel. The result is a significant calibration change: the near side of the river shore moves dramatically inward into the river when the feature is moved down.

The calibration results also depend on the particular algorithm that was selected. Since all algorithms use different heuristic assumptions for prioritizing the non-linear optimization scheme, they fail in different ways when confronted with poor mapping data. Figure 11 shows that -- in this particular case -- the Weng algorithm performed worse than the Tsai algorithm (compare with Figure 10a). In other cases, the opposite was the case.

These examples demonstrate that real exterior construction applications impose very hard requirements on AR and, in particular, on optical camera calibration. In contrast to artificially created demonstrations shown in laboratory settings to exhibit the general concepts of an approach, real applications provide challenging side constraints. For example, due to the fact that we are visualizing a new bridge, a river keeps us from getting closer to the buildings we use for camera calibration. In our work, we are thus emphasizing pragmatic concepts to cope with such real problems.



(a) Tsai: Feature at [537,270]



(b) Tsai: Feature at [537,269]

Figure 10: Different results of full Tsai calibration, with one of the mapped features having been moved by one pixel.

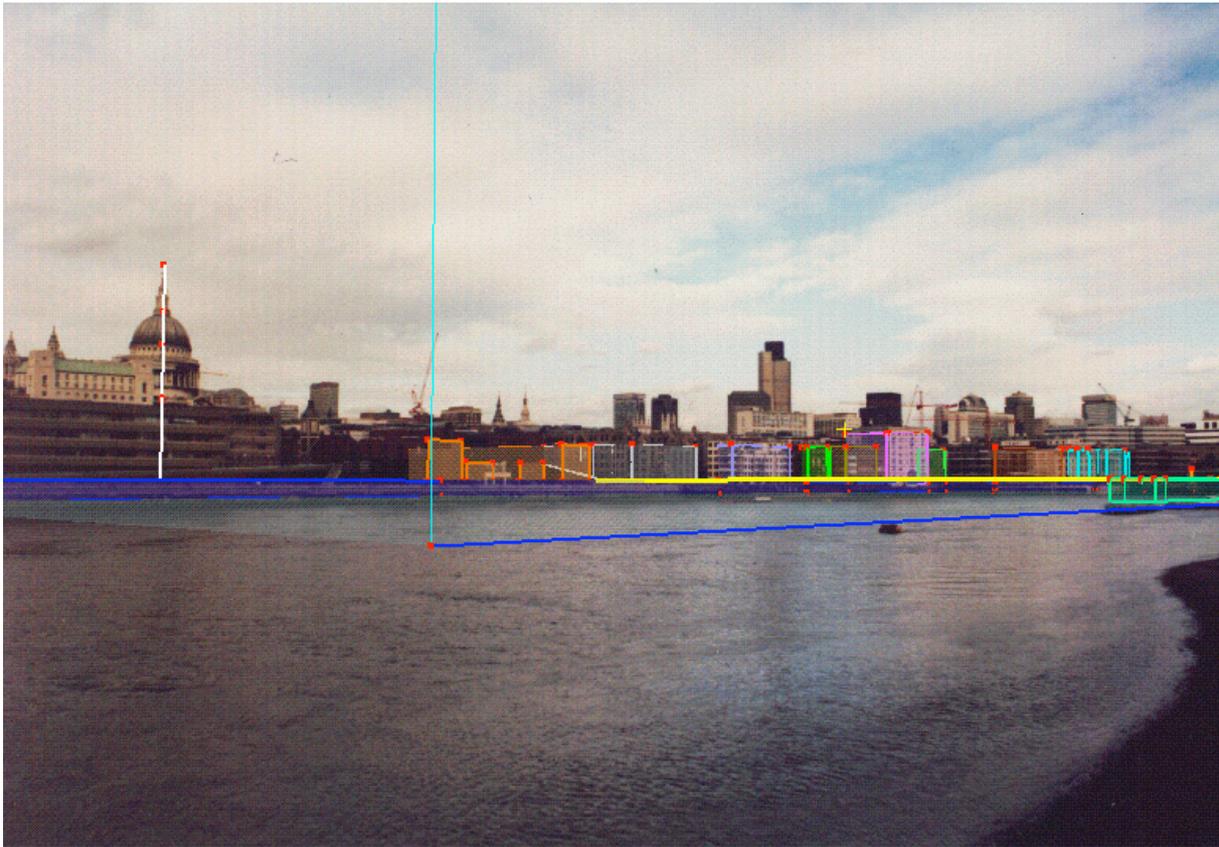


Figure 11: Results of Weng calibration after automatic correction of feature mappings (10 iterations).

- Considering current calibration instability, the need for good reality models becomes evident. They need to be very precise. Furthermore, they should cover targets in a widely spread three-dimensional volume. For example, the inclusion of distant high rises and power poles, or some targets on the near side of the river would greatly stabilize the results. Finally, the targets need to be easily detectable and precisely locatable in image data. Tips of power poles have proven very suitable for this purpose.
- To help users correctly identify image features, InCal automatically investigates which image feature currently has the largest influence on a calibration misalignment. By moving that feature by one pixel up, down, left, or right, a new calibration generates a much smaller mismatch between image features and projected scene features. Figure 11 shows the results of Weng's algorithm after 10 iterations.
- Another pragmatically useful concept suggests exploiting as much externally available information as possible. Thus, we exploit the flexibility of Tsai's calibration system. Using interactively provided data on the camera's focal length, center and aspect ratio, Tsai's algorithm computes only the external camera position and orientation parameters, as shown in Figure 12. The algorithm can be constrained even further

by providing approximate camera location and orientation information, as is done in the tracking systems being discussed next.

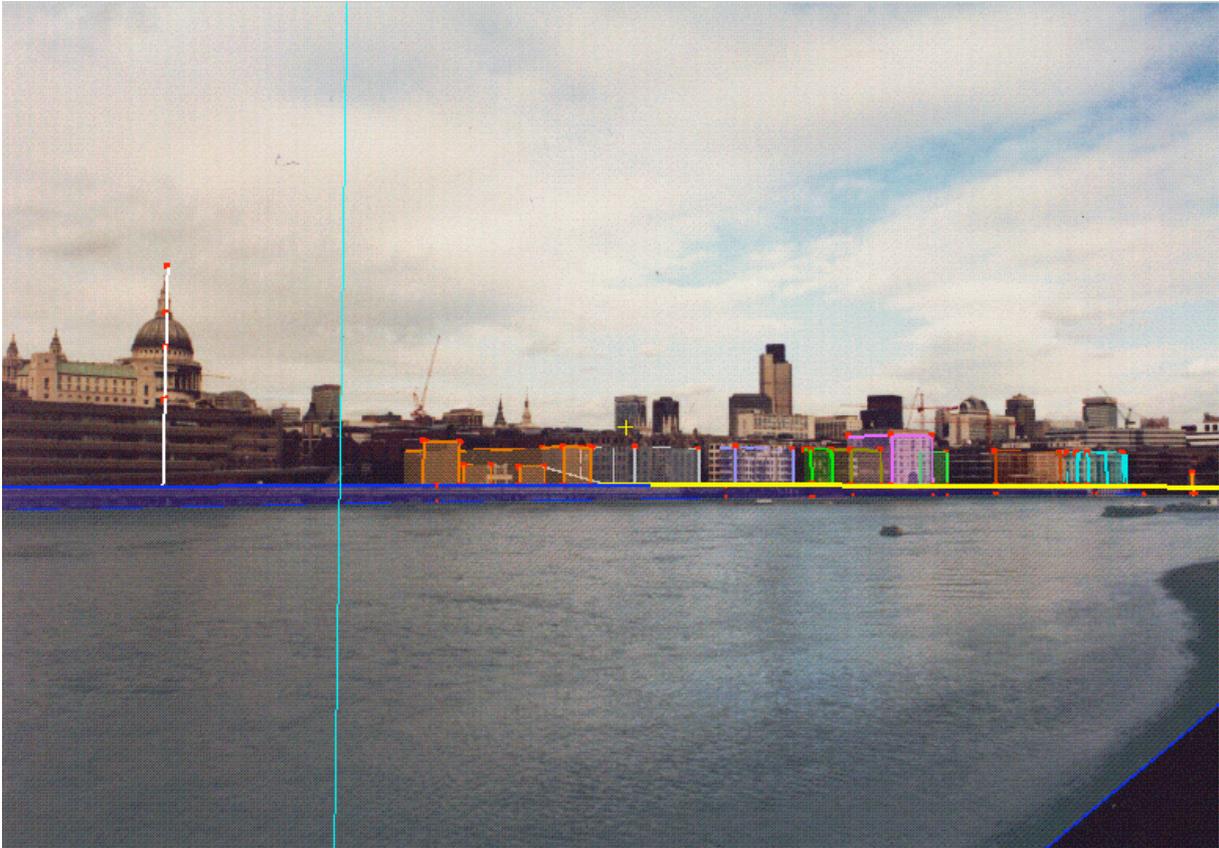


Figure 12: Tsai calibration using a fixed focal angle of 42 degrees.

3.4 Tracking

When live video streams (image sequences) rather than individual images are augmented, camera tracking becomes an issue. Due to the numerical instability of calibration routines, it is not advisable to recalibrate each image from scratch. The result would be a rather bumpy camera path. Rather, camera motion must be modeled as part of the parameter estimation process, influencing and stabilizing the system.

Kalman filtering is a well-established technology to stabilize the estimation of camera motion [5,13] and user motion [2]. With D. Koller, we have developed a three-dimensional camera motion model, which accounts for camera velocity and acceleration [29,31,30]. With this motion model, physical camera motion can be calculated and tracked after the first two images have been taken, predicting the real-world camera trajectory for subsequent

images. According to the predicted next camera position, the system determines local search areas in the image where the black squares should be. From their actually determined locations in the image a corrective term is calculated to influence the camera motion model. We have used this Kalman filtering approach in live tracking demonstrations, operating at about 10 frames per second on an SGI Indy workstation (with R5000 processor). Currently, the system tends to adjust only slowly to changing camera motion -- due to its built-in technology to assume smooth camera motion. When faced with abrupt camera jerks, the algorithm tends to continue moving the virtual camera in a steady direction for a while before reversing to account for the jerk. As a result, virtual objects tend to have a swinging behavior in the scene and the black square targets are frequently lost such that the tracker needs be reinitialized. Similar observations have been reported by Lowe and by Ravela et al. [38,51].

As an alternative, we have begun exploring simpler, more direct tracking schemes. Using the extended tool box of Tsai's calibration routines, we use calibration parameters from the previous image to initialize the next calibration. This works particularly well under circumstances when we know that the camera motion is constrained. When the camera is known to be on a tripod, we can use the same camera position throughout the entire image sequence, recalculating only the rotational components. But even when the camera is not stationary, we have been able to obtain very good and stable calibrations by assuming that -- at operating speeds of about 23 frames per second -- the camera hasn't moved much in between consecutive frames. We thus initialize the non-linear least squares motion estimation routine with the previous motion parameters, allowing them to resettle according to the updated matching data. When the camera is rotated too fast, a complete recalibration is achieved in a third of a second.

We use the same technology both for the live demonstrations and for the slower, interactive calibration of complex cityscapes. The system works very well for medium-speed camera motions. We are able demonstrate the system with a hand-held IndyCam. Virtual objects are much more stable within the real scene; the characteristic swinging of the Kalman filtering approach is not observed.

3.5 Tracking stability

Tracking stability is a key issue in AR. Virtual objects must be precisely positioned in the picture and keep their position over time despite camera motion and noise.

In our demonstrations generating high-quality presentations of new virtual buildings within a given environment, we have observed that apparent stability within the scene is much more important than the precise calibration of individual images by themselves. Thus, it is very important to make suitable stabilizing assumptions. In particular, assuming that the internal camera parameters (focal length) remain constant throughout the demonstration provides significant overall improvements -- even though such assumption might be violated. Using schemes that avoid computing all six external parameters together have further stabilized the augmentations. To this end, we currently determine camera rotation and translation in two discrete steps, optimizing the non-linear equations for only three unknowns at a time.

3.6 Sensor fusion: GPS and optical tracking

Many different technologies can be used to determine the current camera location, such as optical, magnetic, mechanical and inertial trackers. Each technology has advantages and drawbacks when compared to the others. Inertial or magnetic sensors, for example, are fast and can track abrupt motions. Yet, they are not precise enough to allow for an accurate alignment of the graphical object in the pictures. Mechanical trackers are more precise, yet they -- as well as magnetic trackers -- severely limit camera motions, essentially requiring controlled, indoor application scenarios, such as virtual studios. Optical tracking can deliver precise calibrations both indoors and outdoors. Yet, it has trouble coping with fast camera motion and with optically too complex scenes.

GPS [37] is another promising solution -- especially for outdoor applications. It is based on a set of 24 satellites orbiting earth at about 20 km height. Each satellite has a high accuracy atomic clock and transmits its time signal in a regular interval. A receiver on earth receives the time signal from at least 4 of these satellites and can calculate its position from the known orbits of the satellites via triangulation. The typical accuracy for standard GPS due to different kinds of error is 98 m in 3D. Using differential GPS (which needs information that comes from a private or public base station or which might be sent from official sources in the near future) brings the accuracy down to the meter level, with expensive equipment to the centimeter level. The problem with GPS is that it only works when the receiver can see the satellites. The signal is too weak to penetrate buildings or other cover. Thus within the canyons of a city's highrises or just standing close to a wall getting enough satellites can be a problem. Another problem with using GPS for real-time AR tasks is its perfectionism. Rather than giving false data it will give no data. Thus, several seconds can pass between successful measurements. The very high

end GPS systems are even slower, requiring a minute or more to initialize under imperfect circumstances. After initialization they are faster, yet when they lose track of some satellites, they require another initialization. In general most GPS receivers are not built for real-time measurements, a typical update rate is once per second, more expensive systems go up to five per second, thus GPS alone is not enough for real-time tracking, even besides the fact that GPS only gives positional information, not orientation.

A robustly operating system can be expected to benefit greatly from a well-designed fusion of sensor information provided by several different devices. For example, Jancene et al. combine optical camera calibration with mechanical tracking technology [26]. State et al. combine optical and magnetic trackers to augment table-top scenarios in real time. While several multi-colored circular targets are visible, the system relies on optical tracking results. When the camera moves off-target, the magnetic trackers ensure that the system maintains an overall sense of orientation and position, reinitializing the optical tracker when the targets come back into sight [60].

In our system we are exploring approaches towards injecting GPS data into optical trackers, since GPS is becoming increasingly available on today's construction sites, and since it is much better suited to outdoor scenarios than magnetic trackers. Attaching a GPS sensor to the camera, we can complement the optical tracking system with camera position data provided by GPS. Since the GPS signal is produced at an unpredictable, asynchronous rate, optical tracking using Tsai's calibration still constitutes the core of the system, integrating GPS information whenever provided and relying on purely optical techniques in the mean time.

4 Augmenting reality

Once appropriate reality models and camera calibrations have been obtained, they form the basis for mixing real and virtual worlds. The subsequent sections describe the different steps that need to be taken to achieve realistic and fast inclusion of virtual information into a real world.

4.1 Geometric data

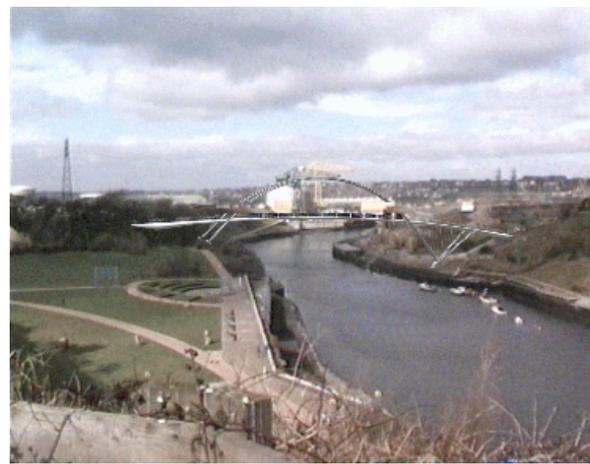
Since exterior construction is a very physical, three-dimensional business, much synthetic data relates directly to the 3D objects being designed and built. Such information is typically represented in 2D or 3D geometric primitives.

With AR, such virtual geometric objects can be integrated into the real environment during all phases of the life cycle of a building. Before the construction project is started, AR can support marketing and design activities to help the customer visualize the new object in the environment (Figure 13). During construction, AR can help evaluate whether the building is constructed according to its design (Figure 14). After the construction is completed, maintenance and repair tasks benefit from seeing hidden structures in or behind walls (Figure 15).

AR thrives on fast, real-time augmentations of the real world. All virtual information thus has to be rendered very quickly. To this end, we carefully tune and prune geometric models to achieve maximal rendering performance while maintaining an acceptable level of realism.



(a) Original scene.

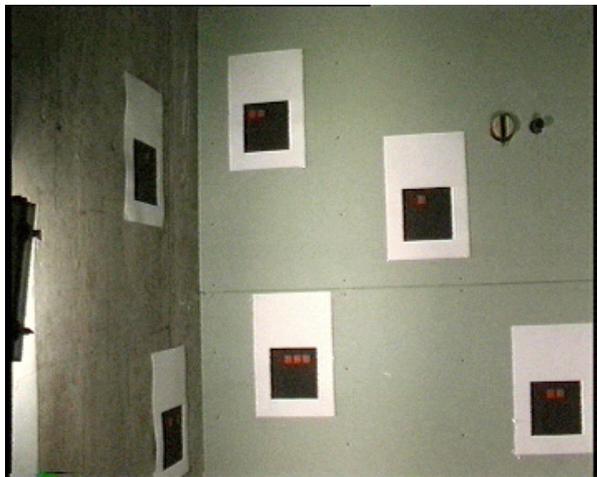


(b) Augmented with planned footbridge.

Figure 13: Side view of a new footbridge, planned to be built across the river Wear in Sunderland, UK.



Figure 14: A virtual wall at a real construction site.



(a) Original image.



(b) X-Ray view into the wall.

Figure 15: Seeing the piping in the wall.

4.1.1 3D models

3D models of medium-sized and large building projects are usually very complex. This is due to the inherent complexity of buildings, as a complete building has thousands of parts. Even if only the outside of the building is of interest typical models are comprised of several hundred thousand polygons. For off-line augmentation (e.g. for video sequences) this is not a big problem, rendering just takes longer. For online interactive or head-mounted augmentations these models are not useful, as even high-powered graphics supercomputers cannot render them at an acceptable frame rate (i.e., more than 10 Hz). Standard geometry optimizations for rendering

like conversion of individual polygons into triangle strips increases performance, but not enough in most cases. They thus have to be simplified. We employ an interactive in-house tool [57] building on standard algorithms [56,58]. Except for closeups, the resulting models are virtually indistinguishable from the originals, albeit at a fraction of the cost (Figure 16a).

Another problem stems from the fact that architectural models are usually not created for presentation but rather for building purposes. They are typically generated using standard CAD tools, which work in wireframe mode and do not pay attention to consistent orientation of polygonal faces. When such orientations are used to compute the surface normal for lighting purposes, many of them point the wrong way and thus are not shaded properly. The resulting models then typically have a checkered look (Figure 16b). There are some tools to help integrating the faces consistently in the model decimation system, but in general there is no automatic solution and some manual work is required until the modeling tools used and the resulting models get better.

A related problem is the optical appearance of the models. Material parameters like reflectivity and colors are not included in the standard modeling tools, thus this information has to be generated. The position and strength of light sources and other global lighting parameters have to be added to the model [47], unless they are extracted from the real images [18] or provided by other global information such as date, time, and place [46].

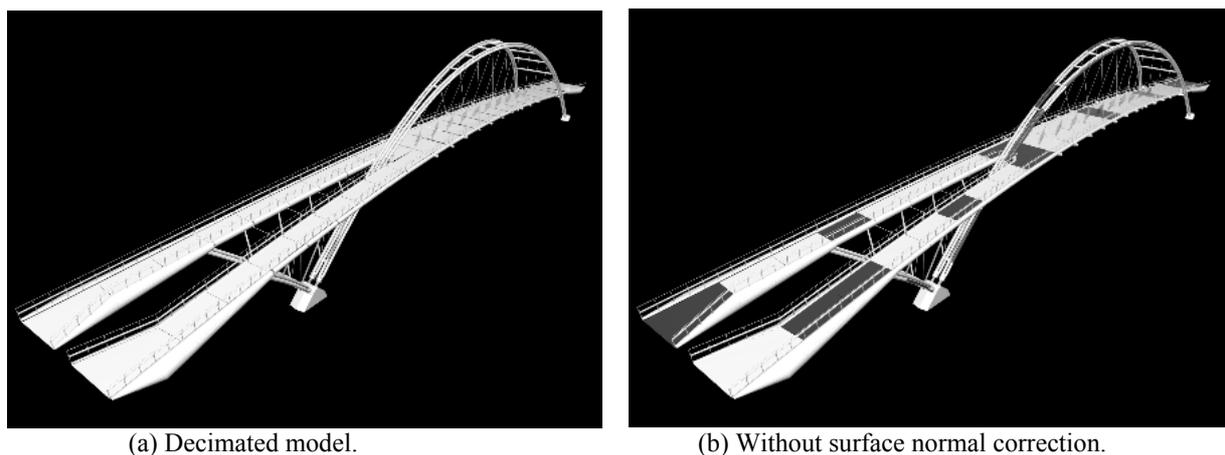


Figure 16: Model of Sunderland footbridge (Decimated from 45000 down to 25000 triangles).

Even if this information were available in the modeling system there is no a standardized way yet to extract it from the modeling system. The most common export format for most CAD systems is DXF, which was designed for the exchange of 2D drawings. It has been extended over the years to handle 3D data as well, but is nowhere

near being an adequate exchange format for high-quality models for rendering. For these purposes the SGI Inventor format [62] and the VRML1 or VRML2 formats are becoming popular, but they are still not universally supported. Public-domain [24] and commercial [49] converters are available, but most of them do not handle all possible variants of DXF reliably, so that sometimes specialized converters need to be written.

4.1.2 2D models

Much information is currently represented as two-dimensional plots rather than three-dimensional models. It currently seems more suitable for individual contractors to maintain their own 2D systems of information relevant to them than for everybody to access an all-inclusive, huge, 3D project model -- even at the risk and cost of having to ensure that changes are quickly propagated between the systems of all relevant subcontractors.

For AR applications however, data needs to be coordinated within a three-dimensional coordinate system. As one simple approach towards integrating 2D plots into a real-world framework, we have arranged the plots as a system of appropriately stacked panes (see Figure 17).

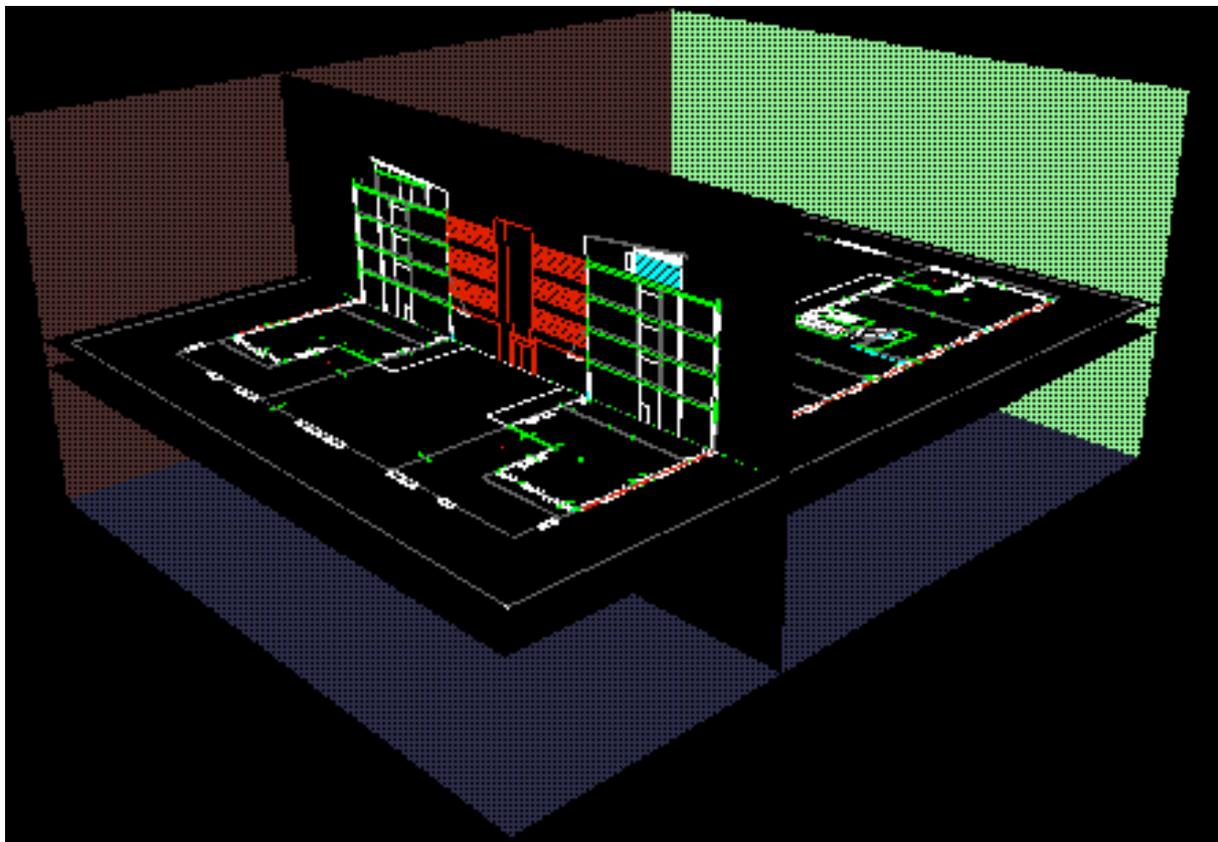


Figure 17: Arrangement of 2D plots in a 3D model.

4.2 Fast rendering and tracking

Besides fast and robust camera tracking, high-quality, real-time rendering [52,54] is the essential ingredient to an AR system. To both track and render at high quality and with high speed, a distributed solution is appropriate. This can either mean distributing across several machines or across tasks.

Distribution across several machines allows the use of cheaper machines and more importantly allows the use of vastly different machines. For example the tracking from simple sensors could be done by a small wearable machine while the high-quality rendering of a complex model is done on a stationary graphics supercomputer [59]. Or just the other way around, in which an expensive optical tracking algorithm using very little information about the scene runs on a supercomputer while simple models or textual information are rendered by the wearable machine. The problem of this multi-machine distribution is the necessary communication. The amount of data that has to be communicated can be quite high, e.g. video images for tracking or from rendering. But in any case this distribution adds lag to the entire system, which is quite detrimental to the effect of immersion.

The alternative is to use multiple threads on a single machine. For this to be useful the machine has to have multiple processors. A short time ago this meant an expensive workstation, but multi-processor Pentium boards are becoming commonplace. Running both tasks on one machine shortens the communication pathways dramatically, as in the worst case a memory to memory copy has to be done and in the best case of a shared memory system just pointers have to be moved. This is the model that we use for our high-quality real-time applications, running the optical tracking on one processor while the other is feeding the graphics pipeline.

Thus the natural separation of an augmented reality system into the two tasks of tracking and rendering allows parallel processing for high throughput and also asymmetric distributed processing to comply with constraints of specialized machinery like wearable computers.

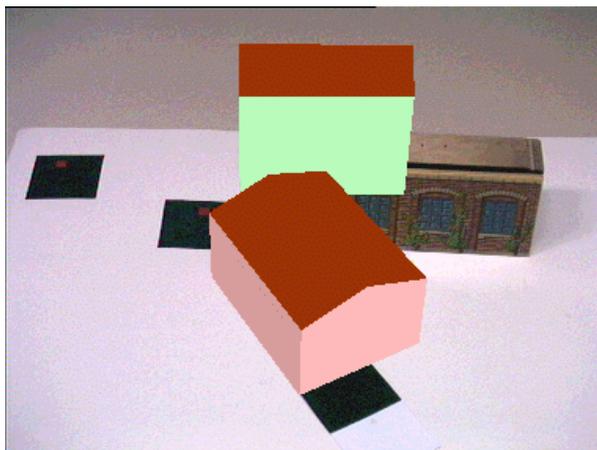
4.3 Interactions between virtual and real objects

Realistic immersion of virtual objects into a real scene requires that the virtual objects behave in physically plausible manners, i.e.: they occlude or are occluded by real objects, they are not able to move through other

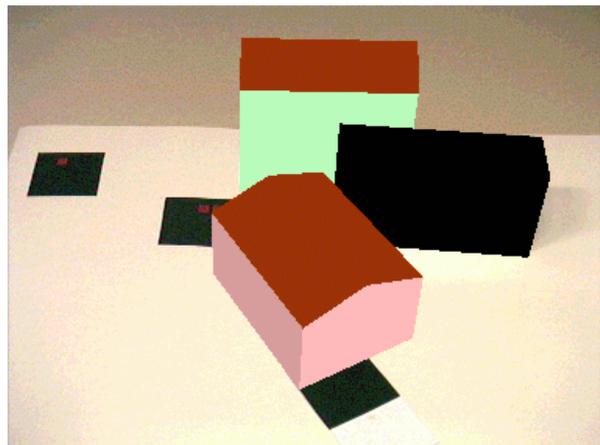
objects, and they are shadowed or indirectly illuminated by other objects while also casting shadows and mirror images themselves.

4.3.1 Occlusion

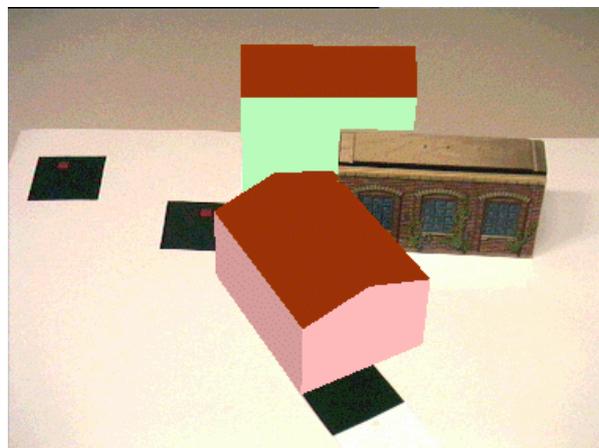
Occlusions between real and virtual objects can be computed quite efficiently by the geometric rendering hardware of high-quality graphics workstations, when provided with a list of the geometric descriptions of all real and virtual models. By drawing real objects in black, the luminance keying feature of video mixing devices can be activated to substitute the respective image area with live video data. As a result, the user sees a picture on the monitor that blends virtual objects with live video, while respecting 3D occlusion relationships between real and virtual objects (Figure 18).



(a) Without occlusion handling.



(b) Reality model shown in black.



(c) With occlusion handling.

Figure 18: A toy house occluding parts of a virtual pink house while being partially occluded by a virtual green house.

Other mixing approaches use depth maps of the real world obtained with a laser scanner [53] or vision-based scene recognition approaches [11,12,28]. The depth maps can be used to initialize the Z-buffer of the graphics hardware [29,73]. Occlusion of virtual objects is then performed automatically. When the virtual object is rendered, pixels that are further away from the camera than the Z values in the depth map are not drawn. By setting the background color to black, the real objects present in the original video are displayed in these unmodified pixels.

Both approaches have advantages and disadvantages, depending on the application. Full 3D geometric models are best for real-time movement of cameras. Polygonal approximations to depth maps can be used over a certain range of camera positions since the synthesized scene model is re-rendered when the camera moves. Copying the depth maps directly into the Z-buffer is the hardest approach: the map needs to be recomputed after each camera motion because the new projective transformation "shifts" all depth values in the depth map. Thus, this approach only works with stationary cameras or with shape extraction algorithms that perform at interactive speeds.

On the other hand, the geometric modeling approach suffers from an inherent dependence on scene complexity. If the scene needs to be represented by a very large polygonal model, the rendering technology may not be able to process it in real time. In contrast, the size of a depth map does not depend on scene complexity. Which approach to use in an application depends on the overall requirements and the system design.

4.3.2 Shadows and reflections

Objects in the real world not only determine their own shading, they also have an influence on the appearance of other, distant objects by means of shadows and reflections [18,19].

With the availability of reality models, standard computer graphics algorithms [17] can be used to compute the geometry of shadows cast by virtual objects onto real ones (see Figures 8 and 9 in [60]). Given the right hardware, this can even be done in real time. Depending on the amount of ambient light in the scene shadows should not completely replace the object they are falling on, but should rather be blended with the image of the underlying object [26].

Reflections are a more difficult topic that can be solved for many useful special cases, but not in general. Reflections of virtual objects in planar real mirrors can be resolved using standard computer graphics techniques [17]. Similar to shadows, perfect reflections are rare in the real world; the reflected object should be blended with the mirror image rather than replacing it. This also allows the simulation of essentially planar reflective surfaces like water (see Figure 19).



Figure 19: Virtual London bridge reflecting in the real water.

Difficult to handle are reflective virtual objects, as in general they would have to reflect things from the surrounding environment that are not visible in the image. For special cases this can be circumvented by placing real reflecting objects, such as a silver sphere, in carefully chosen locations in the scene and using their reflections as an environment map to determine the light reflections for the virtual object [61].

An alternative approach would be to use a high-quality reality model to render the reflection onto the virtual object, but that would defeat the idea of augmented reality not having to build such a complex, photometrically precise model.

4.3.3 Physical constraints: solid virtual objects, gravity

For an augmented world to be realistic the virtual objects not only have to interact optically with the real world, but also physically. This applies to virtual objects when animated or manipulated by the user. For example, a virtual chair shouldn't go through walls when it is moved, and it should exhibit gravitational forces [7].

According to the law of non-penetration, two solid objects cannot be in the same place at the same time. Thus, virtual objects should prevent themselves from moving into or through other objects. Given a reality model, this behavior can be achieved using the same collision detection and avoidance systems that are used for virtual reality systems [75].

Another important physical law concerns gravity: when not supported by anything virtual objects should move downwards while obeying the law of non-penetration until they reach the lowest possible position.

These two laws make up the most important physical constraints. A full physical simulation including more aspects of the interaction between real and virtual objects, such as elastic behavior and friction, would be desirable. For off-line applications this is possible if enough information about the virtual objects and a complete enough reality model is available. For real-time applications most simulation systems are not fast enough. Yet, even simple implementations of the above rules will make the system much more realistic.

4.4 Leaving reality behind

Augmented reality and virtual reality are not two discrete alternatives but rather part of a spectrum of mixed realities [44] with full virtual reality on one end and full physical reality on the other. Augmented Reality is in the middle, combining the best of both worlds. But sometimes it might be desirable to lean more in one direction or the other.

Because registered augmented reality by concept needs real images, its freedom of movement is limited to the places where an image recording device (possibly a human eye) can go. Unless employing rather exotic hardware like camera-carrying blimps [50] this limits the possible positions of the viewer. Virtual reality on the

other hand allows complete freedom of movement, as computer generated images can be generated for every possible viewpoint. Thus it is sometimes desirable to leave the augmented reality behind and switch into the virtual reality to take a look from a point where it's physically impossible to get, e.g. from above.

The disadvantage of leaving the augmented reality behind is that the the view now has to be constructed entirely from synthetic information, i.e., the virtual objects and the reality model, and from previous image data. A very promising area of current computer graphics research to circumvent this shortcoming is image based rendering [9,20,36,40], which strives towards generating images from new viewpoints given some images from other viewpoints. A future system might employ a camera to take images while viewing the augmented scene and later using these images to give the freedom of movement to the user while incorporating images taken on site just a short time ago, thus being as current as possible.

4.5 Non-geometric data

Virtual information doesn't have to be exclusively three-dimensional and geometric (polygonal) [59]. In large construction projects, many kinds of information are gathered, stored, maintained and shared digitally in many formats [34]. When suitable information visualization schemes are used, AR can bring any such information to users roaming the real world. Examples of such sources of information include:

- **Business data**, such as project schedules and time lines, building codes and tolerances, customer preferences, as well as texts describing sub-contractors (e.g.: their WWW entries). When available "on location", such information can provide the basis for on-site evaluations analyzing whether the construction is progressing according to schedule, which firm is to come in next, etc. Furthermore, contractors can be contacted immediately to discuss discrepancies between the specification and what was really built [10]. Business information can be made available as 2D windows on the world [15] or on virtual sheets of paper or panels attached to real walls or floating next to objects under discussion (see the virtual panel showing a heating schedule in **Error! Reference source not found.a** and the fire hose instruction sheet in **Error! Reference source not found.b**).
- **Image/video data**. Using on-line brochures of contractors and catalogs of material suppliers, customers can choose between various options, e.g. different wood grains for doors, within the real life context. Like

business data, brochures and catalogs can be presented (texture-mapped) on virtual panels. Even entire videos can be presented, advertising how novel window or door designs will much improve their ease of use, how they will reduce the evaporation of heat, or other things. Multi-media information will be at the architect's and customer's finger tips.

Images and videos can also show equipment like pipes or electric wiring inside walls from photos taken before the equipment was covered with plaster. When such images are shown well aligned with the walls they provide the illusion of X-ray vision skills (see Figure 15b).

- **Process data**, indicating the operating conditions of machinery in use. Such information is essential to building maintenance and renovation tasks. It can help find leaks in pipes etc. AR can show the information right on the device that is being inspected. In Figure 20a, we use a red-to-blue coloring scheme to symbolically represent warm-to-cold temperature variations within heating pipes

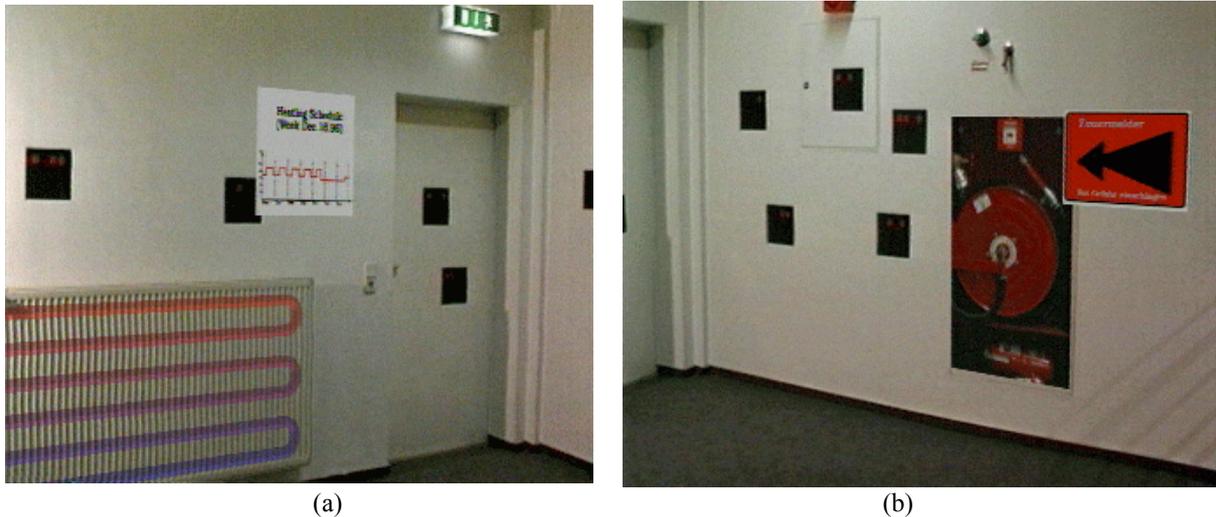


Figure 20: a) Heating schedule and visualized temperatures within heating pipes. b) Picture of a fire hose inside a cabinet and an instruction sheet, superimposed on (next to) the cabinet door.

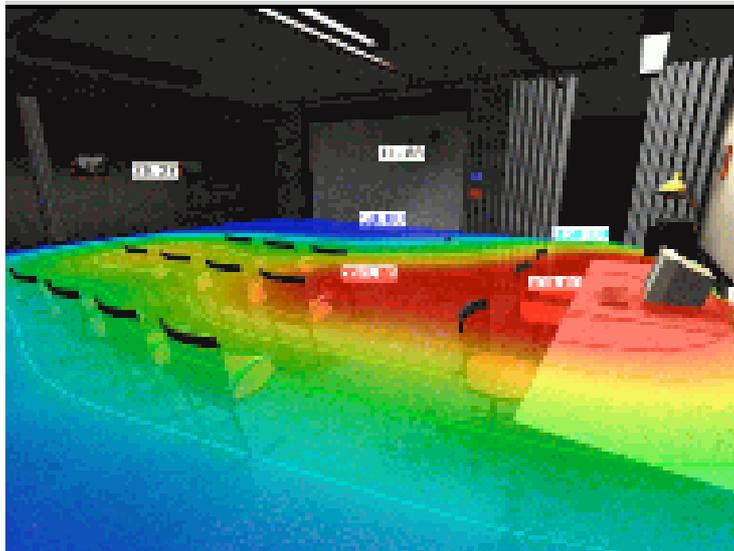
- **Instructions** what to build next. AR can remind people of the correct scheduling of tasks, showing them one step at a time what to do next (see Figures 20b, 25). Much time and material is wasted when a wall is erected too early and has to be removed again so that large equipment (e.g.: elevator equipment or a water tank) can be put into its correct place.

Further instructions can serve as navigation aides. Large constructions sites like a new airport or shopping mall are ever-changing mazes of roads. AR can help people navigate within the area, e.g., to find the currently shortest or safest path from one place to another (see Figure 21).



Figure 21: Navigation aide.

- **Simulated data**, such as the expected circulation of air within a building and lighting simulations to verify and optimize the placement of windows and artificial light sources inside buildings (Figure 22).



file: IGD-lighting.ps,height=5cm

Figure 22 : Simulated illuminance values on a working plane.

Various exterior construction applications benefit from granting workers and engineers on site access to all of these sources of information, augmenting reality in suitable, non-disturbing ways.

5 Diminishing reality

Many construction projects require that existing structures be removed before new ones are built. Thus, just as important as augmenting reality is technology to diminish it.

Figure 23a shows one of several pictures of TV-towers on Monte Pedroso near Santiago de Compostela, Spain. The project to build a new communications tower on Monte Pedroso for completion by 1999 was conceived by the Concello de Santiago as part of an overall plan for the city and for the mountain. The new tower, designed by the architects Sir Norman Foster and Partners and consulting engineers at Ove Arup and Partners, UK, is primarily intended for telecommunications users, but because of the drawing power of unusual buildings and its location, the tower is also intended to appeal to public visitors.



(a) Original image.



(b) Diminished reality.



(c) Augmented diminished reality.

Figure 23: Monte Pedroso near Santiago de Compostela, Spain.

Prior to augmenting the image with a model of the new tower, the existing towers need to be removed (Figure 23b). To this end, a part of the sky has to be extrapolated into the area showing the TV-towers and the barracks. Then the new tower can be put into place (Figure 23c).

In principle, the problem of diminishing reality consists of two phases. First expiring buildings have to be identified in an image. When such structures are well represented in a reality model, they can be located by projecting the model into the image according to the current camera calibration.

Next, the outdated image pixels need to be replaced with new pixels. There is no general solution to this problem since we cannot know what a dynamically changing world looks like behind an object at any specific instant in time -- unless another camera can see the occluded area and provides us with the information. Yet, some heuristics can be used to solve the problem for various realistic scenarios:

- The most simple approach might just de-emphasize outdated areas; greying them out or smoothing across them with large convolution windows.
- More sophisticated morphological approaches might extrapolate properties of surrounding "intact" areas (e.g: a cloudy sky) across outdated areas.
- When a building is to be removed from a densely populated area in a city, particular static snapshots of the buildings behind it can be taken and integrated into the reality model. Computer graphics technology can then map those textures into the appropriate spaces of the current image.

First results of such "X-ray vision" capabilities are shown in Figure 15b. In the case of diminished reality, the supplemented video image (piping) must be displayed at full alpha-level, thus completely hiding the current video (wall).

- For video loops of a dynamically changing world, computer vision techniques can be used to suitably merge older image data with the new image. Faugeras et al. have shown that soccer players can be erased from video footage when they occlude advertisement banners: For a static camera, changes of individual pixels can be analyzed over time, determining their statistical dependence on camera noise. When significant changes (due to a mobile person occluding the static background) are detected, "historic" pixel data can replace the current values [76].

In more general schemes using mobile cameras, such techniques can lead towards incremental techniques to diminish reality. While moving about in the scene, users and cameras see parts of the background

objects. When properly remembered and integrated into a three-dimensional model of the scene, such "old" image data can be reused to diminish newer images, thus increasingly effacing outdated objects from the scene as the user moves about.

We currently use interactive 2D tools to erase old structures from images. This approach can only be used for static, individual photos, but not for video sequences from a live, dynamically moving camera.

6 User interaction in a three-dimensional augmented world

Augmented reality is a technology by which a user's view of the real world is augmented with additional information from a computer model. Users can work with and examine real 3D objects while receiving additional information about those objects or the task at hand. Exploiting people's visual and spatial skills, AR thus brings information into the user's real world rather than pulling the user into the computer's virtual world.

6.1 A real-world interface to virtual worlds

The power of AR as a real-world interface to virtual worlds becomes evident when we present a virtual building on a real table top (Figure 24a). Rather than using complex 2D or 3D interaction metaphors, users can walk around the table to inspect the building from all sides, rotating a reference pattern on the table to turn the building. Using several reference patterns, several virtual objects can be moved independently of each other (Figure 24b). At the same time, users can reference other material on the table, such as maps, and discuss the model with colleagues.

In these examples, our approach incorporates the concepts of the DigitalDesk [69,70], the metaDESK [25] and the Responsive Workbench [32], extending them towards augmenting more complex realities than planar desktops. It leads to hybrid digital/real mock-ups, e.g., of a complex construction site for which a scaled-down physical model of the environment might already exist. AR can augment it with digital prototypes of new

buildings or objects until their design and layout matures. Similar concepts apply when extensions or renovations of large manufacturing facilities are planned.

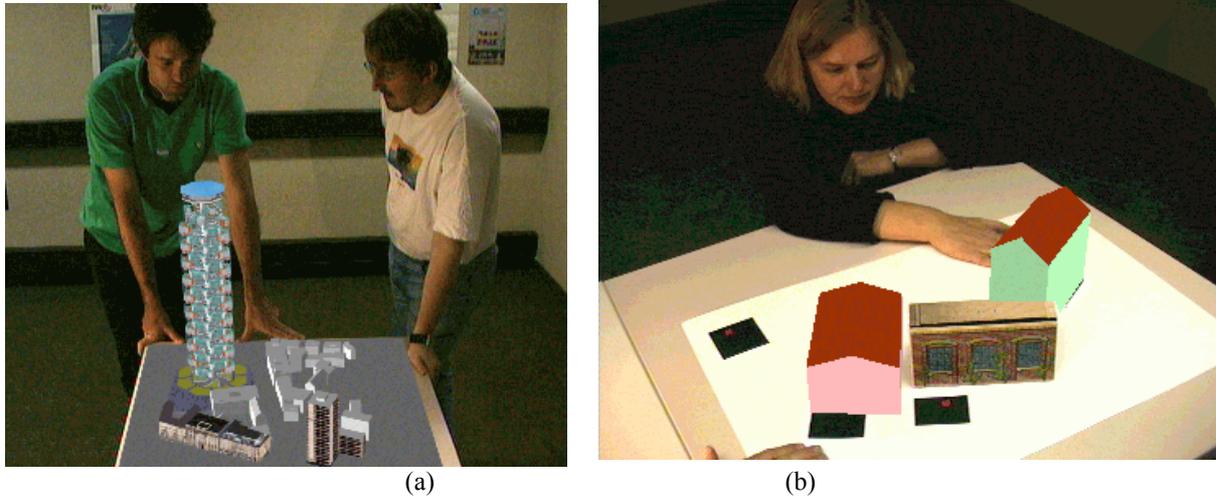


Figure 24: a) Virtual buildings viewed on a real table. b) Interactive layout of a city scape.

6.2 Computer-provided guidance to real-world tasks

Computer augmentations of the real world can provide the user with dynamic, up-to-date instructions how to perform a task [8]. We explore and demonstrate the potential of this new paradigm in our laboratory at the example of a Tangram game (Figure 25).

In contrast to 2D games played on a monitor screen, the Tangram game takes place on a real table using real Tangram pieces. Figure 25a shows our setup. The user sits in front of a small cubicle in which the game takes place. A camera behind his shoulder records the scene. Our live AR-system (see 1.2 , 3.1.1 , 3.4) running on the Indy in the corner is capable of tracking camera or cubicle motion, while maintaining the three-dimensional augmentations of the scene.

The computer three-dimensionally augments the real world view with instructions how to assemble a complex Tangram shape. A virtual sheet shows the entire shape (Figure 25b). Both on the sheet and superimposed on the table the next piece to align is highlighted. Either on the monitor or in a "feed-through" head-mounted display,

the user sees the augmentations while working in the scene. The user can now proceed assembling the entire Tangram shape.

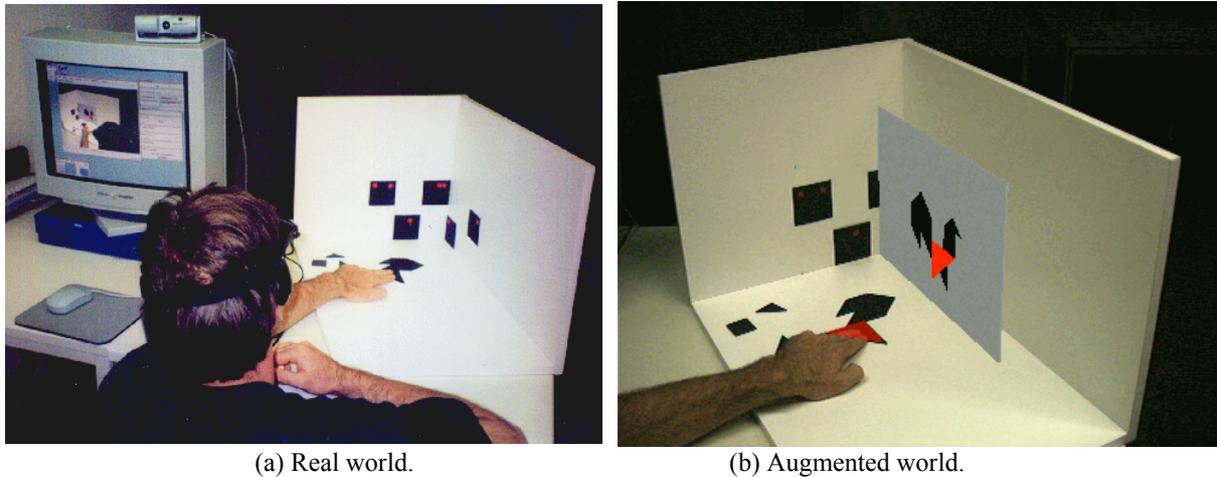


Figure 25: Computer-guided assembly of a Tangram shape.

Similar concepts also apply to exterior construction tasks, such as the computer-guided installation of an elevator in its shaft where extreme care has to be taken to plan the path along which to insert the elevator. Other applications include the assembly or repair of machines [16], and the installation of alluminum struts in a diamond shaped spaceframe [68].

6.3 Two-way human-computer interaction in the real world: reality tracking

To fully exploit the AR paradigm, the computer must not only augment the real world, but also receive input (feed back) from it. In truly three-dimensional human-computer interaction, actions or instructions issued by the computer cause the user to change the real world -- which, in turn, prompts the computer to perform further actions, as demonstrated in the ALIVE project [39,74] in which a person can interact with a virtual dog, gesturing it to sit down etc.

The use of magnetically tracked devices, such as data gloves and body suits, and GPS-tracked laser pointers provide three-dimensional, interactive schemes to communicate with the computer. Speech or sound input, as well as gesture recognition provide further interaction means. In their spaceframe construction demonstration, Webster et al. equip users with barcode readers to let them inform the computer of newly selected struts [68].



Figure 26: Real-world-based tic tac toe.

Yet, not all changes in the real world can be tracked by a discrete set of (more or less bulky) physical tracking devices, and the user also cannot describe all changes verbally or with a barcode reader. To this end, AR-systems must be capable of automatically detecting and tracking changes in the real world. We explore *optical reality tracking* approaches at the example of a Tic Tac Toe game (Figure 26). The physical setup of the game is similar to the one used in the Tangram scenario (Figure 25a). In this case, a real Tic Tac Toe board is sketched out on the surface of the cubicle. After the user has placed a stone on the board and hit the virtual "GO"-button by moving his hand across it, the computer detects the stone in the image and plans a counter move, indicating its decision by a virtual cross on the board and instructing the user on the virtual panel to continue. In this example, the user does not need to touch a keyboard or mouse once the game has started. All interactions occur directly on the game board, embedded in the real world.

These demonstrations are a first, essential step towards enabling truly real-world-based interactive AR applications. They form the basis of real construction, maintenance and repair tasks.

Sources of graphical material

- Figure 1: Private communication with David Leever, BICC (CICC Project).
- Sunderland Newcastle, UK:
Work conducted in collaboration with Ove Arup and Partners (CICC Project).
 - Figure 3a, Figure 13a,b: Picture of the river Wear, Sunderland Newcastle, UK. The bridge model was provided by Sir Norman Foster and Partners.
- Thames river, London, UK:
Work conducted in collaboration with Ove Arup and Partners (CICC Project)
 - Figure 9 - Figure 12, Figure: Pictures of the river Thames, London near St. Paul's Cathedral. The 3D model of a designed milleneum footbridge was provided Sir Norman Foster and Partners.
 - Figure 5: Three-dimensional model of selected areas (4 tiles) of the city of London. Acquired for the CICC project by Ove Arup and Partners.
- Bluewater Kent, UK:
Work conducted in collaboration with Bovis and Trimble Navigation Limited (CICC Project).
 - Figure 14: Picture from a video sequence.
- Santiago de Compostela, Spain:
Work conducted in collaboration with Ove Arup and Partners (CICC Project).
 - Figure 23a,b,c: Picture of Monte Pedroso near Santiago de Compostela, Spain. The model of the TV-tower was provided by Sir Norman Foster and Partners.
- Gmunder Straße, Munich, Germany:
Work conducted in collaboration with Philipp Holzmann AG, Germany.
 - Figure 3b, 15a,b: Indoor pictures of a bathroom under construction.
 - Figure 17: View of the CAD model.
- Valbonne, France:
Work conducted in collaboration with INRIA Sophia-Antipolis (Realise and Cumuli Projects).
 - Figure 6a,b,c: Picture and reconstructed model of the Arcades in Valbonne, France.
- Royal Institute of Chartered Surveyors, London, UK:
Work conducted by U. Leeds, JRC and BICC (RESOLV Project).
 - Figure 7: Pictures of the RESOLV trolley and the reconstructed model of the Royal Institue of Chartered Surveyors. Courtesy of the RESOLV project.

Acknowledgments

The research was conducted 1995-1998 and was financially supported by the CICC project (ACTS-017) in the framework of the European ACTS programme and by the Cumuli project (LTR-21914) in the ESPRIT programme. The laboratory space and the equipment was provided by the European Computer-industry Research Center (ECRC).

We are grateful to the current and former colleagues at Fraunhofer IGD, ZGDV and ECRC for many useful comments and insights which helped us develop and refine the work. Particular thanks go to Dieter Koller and Eric Rose. Both the CICC and the Cumuli consortium have deeply influenced our approach.

References

1. A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 17(6):562-575, June 1995.
2. R. Azuma and G. Bishop. Improving Static and Dynamic Registration in an Optical See-through HMD. *Proc. Siggraph'94*, Orlando, FL, July 1994, pp. 194-204.
3. R.T. Azuma. A survey of augmented reality. *Presence, Special Issue on Augmented Reality*, 6(4):355-385, August 1997.
4. M. Bajura and U. Neumann. Dynamic Registration Correction in Video-Based Augmented Reality Systems. *IEEE Computer Graphics and Applications*, 15(5):52-60, 1995
5. Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, New York, 1988.
6. J. Bowskill and J. Downie. Extending the Capabilities of the Human Visual System: An introduction to Enhanced Reality. *Computer Graphics*, 29(2):61-65, 1995.
7. D.E. Breen, E. Rose, and R.T. Whitaker. Interactive Occlusion and Collision of Real and Virtual Objects in Augmented Reality. Technical Report ECRC-95-02, ECRC, Arabellastr. 17, D-81925 Munich, 1995.

8. T.Caudell and D. Mizell. Augmented Reality: An Application of Heads-up Display Technology to Manual Manufacturing Processes. *Proc. Hawaiian International Conference on System Sciences (HICSS'92)*, pp. 659-669.
9. S.E. Chen and L. Williams. View Interpolation for Image Synthesis. *Computer Graphics (Proc. Siggraph'93)*, 27:279-288, August 1993.
10. CICC: Collaborative Integrated Communications for Construction. ACTS AC-0017, 1995-1998, <http://www.hhdc.bicc.com/cicc/>, 1995.
11. CUMULI: Computational Understanding of Multiple Images. Esprit LTR-21914, 1996-1999, <http://www.inrialpes.fr./CUMULI/>, 1996.
12. P.E. Debevec, C.J. Taylor, and J. Malik. Modelling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach. *Proc. Siggraph'96*, New Orleans, Aug. 4-9, 1996, pp. 11-20.
13. A. Gelb (ed.). *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.
14. O. Faugeras, S. Laveau, L. Robert, G. Csurka, and C. Zeller. 3D reconstruction of urban scenes from sequences of images. In A. Gruen, O. Kuebler, and P. Agouris (eds.), *Automatic Extraction of Man-Made Objects from Aerial and Space Images*. Birkhauser, 1995.
15. S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the World: 2D Windows for 3D Augmented Reality. *Proc. UIST'93*, Atlanta, GA, 1993, pp. 145-155.
16. S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-Based Augmented Reality. *Communications of the ACM (CACM)*, 36(7):53-62, July 1993.
17. J.D. Foley, A. Van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics, Principles and Practice*, 2. Edition. Addison Wesley, 1989.
18. A. Fournier. Illumination Problems in Computer Augmented Reality. *Journee INRIA, Analyse/Synthese d'Images (JASI)*, January 1994, pp. 1-21.
19. A. Fournier. Computer Augmented Reality and Illumination. *Proc. International Workshop MVD'95: Modeling - Virtual Worlds - Distributed Graphics*, St. Augustin, Germany, Nov. 1995.

20. S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The Lumigraph. *Proc. Siggraph'96*, New Orleans, Louisiana, Aug. 4-9, 1996, pp. 43-54.
21. D.S. Greer and M. Tuceryan. Computing the Hessian of Object Shape from Shading. Technical Report ECRC-95-30, <http://www.ecrc.de>, 1995.
22. A. Hildebrand, S. Müller, and R. Ziegler. REALISE _ Computer Vision basierte Modellierung für Virtual Reality. *Proc. International Workshop MVD'95: Modeling - Virtual Worlds _ Distributed Graphics*, Sankt Augustin, Germany, Nov. 1995, pp. 159-168.
23. B.K.P. Horn, and M.J. Brooks. *Shape from Shading*. MIT Press, Cambridge, MA, 1989.
24. Silicon Graphics Inc. and Abaco Systems Inc. *DxfToIV*. Tool Converting Autodesk DXF R12 Format into Open Inventor 2.0 Files, 1995.
25. H. Ishii and B. Ullmer. Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. *In Proc. CHI 97*, Atalanta, GA, March 1997.
26. P. Jancene, F. Neuret, X. Provot, J.-P. Tarel, J.-M. Vezien, C. Meilhac and A. Verroust. RES: Computing the Interactions Between Real and Virtual Objects in Video Sequences. *Proc. 2nd IEEE Workshop on Networked Realities*, Boston, MA, Oct. 1995, pp. 27-40.
27. A. Janin, D. Mizell, and T. Caudell. Calibration of head-mounted displays for augmented reality applications. *Proc. of the Virtual Reality Annual International Symposium (VRAIS'93)*, 1993, pp. 246-255.
28. T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A Stereo Machine for Video-Rate Dense Depth Mapping and Its New Applications. *Proc. 15th IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 1996.
29. G.J. Klinker, K.H. Ahlers, D.E. Breen, P._Y. Chevalier, C. Crampton, D.S. Greer, D. Koller, A. Kramer, E. Rose, M. Tuceryan, and R.T. Whitaker. Confluence of Computer Vision and Interactive Graphics for Augmented Reality. *Presence, Special Issue on Augmented Reality*, 6(4):433-451, August 1997.
30. D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan. Automated Camera Calibration and 3D Egomotion Estimation for Augmented Reality Applications. *7th Int'l Conference on Computer Analysis of Images and Patterns (CAIP'97)*, Kiel, 1997.

31. D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan. Real-time Vision-Based Camera Tracking for Augmented Reality Applications. *Proc. ACM Symposium on Virtual Reality Software and Technology (VRST'97)*, Lausanne, Switzerland, Sept. 15-17, 1997.
32. W. Krueger, C.-A. Bohn, B. Froehlich, H. Schueth, W. Strauss, and G. Wesche. The Responsive Workbench: A Virtual Work Environment. *IEEE Computer*, pp. 42-48, 1995.
33. K.N. Kutulakos, J. Vallino, Affine Object Representations for Calibration-Free Augmented Reality. *Proc. Virtual Reality Ann. International Symposium (VRAIS '96)*, 1996, pp. 25-36.
34. D. LeEVERS. Inner Space, the Final Frontier. *Proc. Int. Conference "From Desktop to Web-Top: Virtual Environments on the Internet, WWW and Networks"*. Picturevill, Bradford UK, April 1997.
35. D. LeEVERS. Private Communication. May 1997.
36. M. Levoy and P. Hanrahan. Light Field Rendering. *Proc. Siggraph'96*, New Orleans Louisiana, Aug. 4-9, 1996, pp. 31-42.
37. Trimble Navigation Ltd. *All about GPS*. <http://www.trimble.com/gps/index.htm>. 1997.
38. D. G. Lowe. Robust Model-Based Motion Tracking Through the Integration of Search and Estimation. *International Journal of Computer Vision (IJCV)*, 8(2):113-122, 1992.
39. P. Maes, T. Darrell, B. Blumberg, and A. Pentland. The ALIVE system: Full-body interaction with autonomous agents. *Proc. Computer Animation '95*, 1995.
40. L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Proc. Siggraph'95*, Los Angeles, CA, Aug. 6-11, 1995, pp. 39-46.
41. J.P. Mellor. *Enhanced reality visualization in a surgical environment*. Master's Thesis. Technical report 1544, MIT AI-Lab, 1995.
42. J.P. Mellor. Realtime Camera Calibration for Enhanced Reality Visualization. *Proc. IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed '95)*, 1995, pp. 471-475.
43. MICC: **M**obile **I**ntegrated **C**ommunications for **C**onstruction. ACTS AC-0088, 1995-1998, <http://www.uk.infowin.org/ACTS/RUS/PROJECTS/ac088.htm>, 1995.

44. P. Milgram and F. Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, E77_D(12), December 1994.
45. P. Milgram, S. Zhai, D. Drascic, D. and J.J. Grodski. Applications of Augmented Reality for Human-Robot Communication. *Proc. International Conference on Intelligent Robots and Systems (IROS'93)*, 1994, pp. 1467-1472.
46. S. Müller, W. Kresse, N. Gatenby and F. Schöffel. Approach for the Simulation of Daylight. *Proc. 6th Eurographics Workshop on Rendering*, Springer Verlag, 1995, pp. 137-146.
47. S. Müller, M. Unbescheiden and M. Göbel. GENESIS – Eine interaktive Forschungsumgebung zur Entwicklung parallelisierter Algorithmen für VR-Anwendungen. In *Virtual Reality – Anwendungen und Trends* (H.J. Warnecke and J.-J. Bullinger, eds.), Springer Verlag, Reihe: Forschung und Praxis, Bd. T35. pp. 321-341, 1993.
48. U. Neumann and Y. Cho. A Self-Tracking Augmented Reality System (USC), *IEEE Virtual Reality Annual International Symposium (VRAIS'96)*, Hong Kong, 1996.
49. Okino. Model Translation Software. <http://www.okino.com/>, 1997.
50. E. Paulos and J. Canny. Ubiquitous Tele-Embodiment: Applications and Implications. *International Journal of Human-Computer Studies, Special Issue on Innovative Applications of the World Wide Web*, 1997.
51. S. Ravela, B. Draper, J. Lim and R. Weiss. Adaptive Tracking and Model Registration Across Distinct Aspects. *Proc. International IEEE Conference on Intelligent Robots and Systems*, Pittsburgh, PA, 1995.
52. D. Reiners. *High-Quality Realtime Rendering for Virtual Environments*. Diplomarbeit, TU Darmstadt, 1994.
53. RESOLV: **R**econstruction using **S**canned **O**ptical **L**aser **V**ideo. ACTS AC-0021, 1995-1998, <http://www.hhdc.bicc.com/resolv/>, 1995.
54. J. Rohlf and J. Helman. IRIS performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. *Proc. ACM Siggraph'94*, Orlando, FL, July 1994, pp. 381-395.
55. E. Rose, D. Breen, K.H. Ahlers, C. Crampton, M. Tuceryan, R. Whitaker, and D. Greer. Annotating Real-World Objects Using Augmented Reality. *Computer Graphics: Developments in Virtual Environments*. Academic Press, 1995.

56. J. Rossignac and P. Borrel. Multi-resolution 3D approximation for rendering complex scenes. *Proc. 2nd Conference on Geometric Modelling in Computer Graphics*, Genova, Italy, June 1993, pp. 453-465.
57. J. Schiefele. *Methoden der automatischen Komplexitätsreduktion zur effizienten Darstellung von CAD-Modellen*. Diplomarbeit, TU Darmstadt, 1996.
58. W.J. Schröder, J.A. Zarge and W.E. Lorensen. Decimation of Triangle Meshes, *Computer Graphics (Proc. Siggraph'92)*, 26:65-70, July 1992.
59. T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R.W. Picard and A. Pentland. Augmented reality through wearable computing. *Presence, Special Issue on Augmented Reality*, 6(4): 386-398, August 1997.
60. A. State, G. Hirota, D.T. Cheng, W.F. Garrett and M.A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Proc. Siggraph'96*, New Orleans, Aug. 4-9. 1996, pp. 429-438.
61. A. State, G. Hirota, D.T. Cheng, W.F. Garrett and M.A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. First picture on <http://www.cs.unc.edu/us/hybrid.html>; unfortunately, this figure is not included in the Siggraph paper, 1996.
62. P.S. Strauss and R. Carey. An Object-Oriented 3D Graphics Toolkit. *Computer Graphics (Proc. Siggraph'92)*, 26:341-349, 1992.
63. R. Szeliski and S.B. Kang. *Recovering 3D Shape and Motion from Image Streams Using Non_linear Least Squares*. Technical Report CRL 93/3, Cambridge Research Lab, Digital Equipment Corporation, One Kendall Square, Bldg. 700, March 1993.
64. R.Y. Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. *Proc. CVPR*, pp. 364-374, 1986. See also <http://www.cs.cmu.edu/rgw/TsaiCode.html>.
65. M. Tuceryan, D. Greer, R. Whitaker, D. Breen, C. Crampton, E. Rose and K. Ahlers. Calibration requirements and procedures for a monitor-based augmented reality system. *IEEE Transactions on Visualization and Computer Graphics*, 1:255-273, Sep. 1995.
66. M. Uenohara and T. Kanade. Vision-Based Object Registration for Real-Time Image Overlay. *Proc. IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMED '95)*, 1995, pp. 13-22.

67. Vanguard: Visualization Across Networks using Graphics and Uncalibrated Acquisition of Real Data. ACTS AC-0074, 1995-1998, <http://www.esat.kuleuven.ac.be/konijn/vanguard.html>, 1995.
68. A. Webster, S. Feiner, B. MacIntyre, W. Massie and T. Krueger. Augmented reality in architectural construction, inspection, and renovation. *Proc. ASCE Third Congress on Computing in Civil Engineering*, Anaheim, CA, June 17-19, 1996, pp. 913-919.
69. P. Wellner, P. Interacting with paper on the digital desk. *Communications of the ACM (CACM)*, 36(7):87-96, July 1993.
70. P. Wellner, W. Mackay and R. Gold. Computer Augmented Environments: Back to the Real World. *Communications of the ACM (CACM)*, 36(7):87-96, July 1993.
71. J. Weng, T.S. Huang and N. Ahuja. Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451-476, 1989.
72. R.G. Willson. Modeling and Calibration of Automated Zoom Lenses. Ph. D. thesis, Robotics Institute, Carnegie Mellon, Pittsburgh, PA. Jan. 1994. (<http://www-cgi.cs.cmu.edu/afs/cs/usr/rgw/www/thesis.html>.)
73. M. Wloka and B. Anderson. Resolving Occlusion in Augmented Reality. *Proc. of the ACM Symposium on Interactive 3D Graphics*, 1995, pp. 5-12.
74. C. Wren, A. Azarbayejani, T. Darrell and A. Pentland. Pfunder: Real-Time Tracking of the Human Body. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):780-785, July 1997.
75. G. Zachmann. Real-time and Exact Collision Detection for Interactive Virtual Prototyping. *Proc. ASME Design Engineering Technical Conference*, CIE-4306, 1997.
76. I. Zoghiami, O. Faugeras and R. Deriche. Traitement des occlusions pour la modification d'objet plan dans une sequence d'image. <http://www.inria.fr/robotvis/personnel/zimad/Orasis6/Orasis6/html>, 1996.