Graz University of Technology

Institute for Computer Graphics and Vision

Dissertation

Visualization in Outdoor Augmented Reality

Stefanie Zollmann

Graz, Austria, June 2013

Thesis supervisors

Prof. Gerhard Reitmayr

Institute for Computer Graphics and Vision

Prof. Tobias Höllerer

University of California, Santa Barbara

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ..

(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

date ..

(signature)

To Tobi

"The goal of visualization is insight, not pictures."

Ben Shneiderman

Abstract

Industrial outdoor applications in the context of architecture, construction and engineering often require fast access to relevant data directly on-site. Augmented Reality (AR) provides this data access, as well as the spatial relationship between virtual information and the physical environment, by accurately overlaying the data directly onto the user's view. Instead of tediously mapping relevant information from a paper sheet or a map into the actual working environment, users are able to focus on the important tasks, while getting the information embedded into their view. Relevant tasks, such as on-site planning of new infrastructures, querying information about subsurface or overground infrastructure elements, surveying of infrastructure, and monitoring of construction sites can benefit from such a visual overlay.

In addition to accurate registration methods and the integration of existing data sources, it is highly important that data is presented in a comprehensible way. In particular, when combining virtual information with information from the physical environment, severe perceptual problems can occur. For instance, missing pictorial cues can lead to a wrong perception of the spatial layout of a scene. Furthermore, even if natural pictorial cues are integrated into the combination of virtual and physical information, often these natural cues are insufficient for estimating absolute or relative depth. Moreover, the visualization of complex information in an AR overlay often leads to information clutter, since the complexity of the data adds to the natural complexity of dynamic outdoor environments.

In this thesis, we address these problems by proposing adequate visualization techniques that take these perceptual difficulties into account. In particular, we introduce methods that 1) provide visual coherence through a seamless integration of virtual content into the physical environment, 2) integrate additional graphical hints for improving the depth estimation and 3) that use information filtering and abstraction that help to avoid information clutter.

Kurzfassung

Im industriellen Anwendungsbereich der Planung, Vermessung und Erstellung von Gebäude- oder Infrastrukturelementen wird oftmals ein schneller Zugriff auf komplexe Daten direkt vor Ort benötigt. Augmented Reality erlaubt diesen Zugriff und bietet zusätzlich den räumlichen Zusammenhang zwischen virtuellen Daten und realer Welt durch eine präzise visuelle Überlagerung. So können Entscheidungen und Arbeiten speziell im Bereich des Architektur- und Bauingenieurwesens unterstützt werden. Beispielsweise seien hier das Vor-Ort Planen von neuen Strukturen, der Zugriff auf Informationen über oberirdische, sowie unterirdische Infrastruktur, das Vermessen von Objekten und das Überwachen von Baustellentätigkeiten genannt.

Neben einer akkuraten Registrierung und der Integration von existierenden Daten und Arbeitsabläufen sind dabei im Besonderen geeignete Visualisierungstechniken notwendig, welche auf die Wahrnehmung des Nutzers angepasst sind. Das Ziel ist es den Mehrwert der Präsentation der Daten in räumlicher Abhängigkeit zu wahren und Verständnisprobleme, welche durch die Kombination von virtuellen und realen Informationen entstehen, zu kompensieren. Im Rahmen dieser Arbeit werden wir die Herausforderungen für Vor-Ort Visualisierungen von industriellen Daten mittels Augmented Reality diskutieren. Hierbei werden wir insbesondere auf die Themen visuelle Kohärenz, Tiefenwirkung und Informationsüberfluss eingehen. Im weiteren Verlauf werden wir Ansätze und Techniken vorstellen, welche es erlauben diese Herausforderungen zu bewältigen. Visuelle Kohärenz kann durch die Extraktion von wichtigen Strukturen aus einem Kamerabild erreicht werden. Eine andere Technik basiert auf der Kombination von Kamerainformation und GIS Daten. Um die Tiefenschätzung zu verbessern, werden wir eine Reihe von zusätzlichen graphischen Hinweisen in die AR Szene integrieren. Darüber hinaus werden wir Filter- und Abstraktionstechniken vorstellen, die es erlauben den Informationsüberfluss zu adressieren.

Acknowledgments

This thesis would not have been possible without the support from all my former colleagues, my current colleagues from the ICG, my friends, and my family. I would like to thank all those people that were involved in my research, but also those who supported me with their encouragement and understanding.

First of all, I thank my advisor Prof. Gerhard Reitmayr for all the advises, ideas, the insights into tracking, scenegraphs, programming of AR prototypes and research in general. I really appreciated the numerous valuable discussions, as well as all the help ranging from mathematical background over to support for writing papers.

I thank Prof. Tobias Höllerer for being my second supervisor and spending time on reading and discussing the contents of the thesis. I also thank Prof. Dieter Schmalstieg for giving me the opportunity to work at the ICG and for valuable discussions. I would not have started with researching without my former advisor Prof. Oliver Bimber. Thanks for introducing me to the research field and arousing my interest for research.

I would like thank the colleagues, I worked with, for great inspiration and discussions. In particular, I want to thank Raphaël Grasset for discussing ideas for Chapter 2, visualization ideas and for sharing his knowledge about user studies, Tobias Langlotz for discussing research ideas and giving me valuable feedback, Denis Kalkofen and Eduardo Veas for all the discussions related to visualization, Christof Hoppe for supplying me with data and navigating the quadcopter for several outdoor tests, Gerhard Schall for supporting several live demos within the Smartvidente project, Tom Pock for support concerning the segmentation methods and Lukas Gruber for all the discussion within the CityofSight project.

My thanks goes also to the administrative staff of our Institute, Christina, Renate, Karin, Andy, Albert and Marc for all the help with red tape and hardware issues.

Also all the hard-working proof-readers, Jonathan Ventura, Raphaël Grasset, Rebecca Voss, Erich Bruns, Tobias Langlotz and Denis Kalkofen, deserve a big thank-you for

spending their time on reading single or multiple chapters of this thesis. I really appreciate your work.

Furthermore, I thank all my flatmates for giving me a new home in Graz, supporting me with delicious cakes and made our flat a great place for relaxing from work and sometimes also a nice place to work. I also would like to thank my best friends that supported me by being friends even though living far away. Thank you; Katrin, Claudi, Steffi, Susi, Sepp, Robert and Naps.

I am very grateful to my complete family for all the support during my studies (and my entire life). Without them, I would not be here writing acknowledgments for my PhD thesis.

Finally, I dedicate this work to Tobi to show my gratefulness for supporting me even in stressful times. Thanks for being such a good friend, partner and colleague in one person. I really appreciate sharing my life and work with you.

Contents

List of Figures

Chapter 1

Introduction

Contents

1.1 Augmented Reality

Augmented Reality (AR) is a human computer interface that integrates virtual information into the user's perception of the physical world by combining virtual and physical information into one view. Such a combination allows for the display of additional information that is not physical present. For instance, virtual information can represent non-existing objects, meta information or hidden information. The visualization of this kind of information has several fields of application.

For instance, the visualization of *non-existing objects* supports all kinds of applications that require fictional content to be embedded into the physical world. Applications range from entertainment applications, where fictional characters can be displayed, to professional planning applications, where proposed objects are superimposed on the physical scene. In Figure 1.1 (Left), a planning application in the context of geospatial data is depicted. As-planned buildings are superimposed on video images of the physical world in order to enable the user to experience their appearance in the physical world.

Meta information supports the user by providing information about physical world objects. Mostly, it is presented in the form of labels (Figure 1.1 Middle), but it includes any abstract information about the real world as well.

Figure 1.1: Visualization of non-existing objects [3], meta information [43] and hidden information in AR.

Furthermore, AR allows visualizing information that would be *invisible* in the physical world, because it is located behind or inside a physical object. X-Ray visualizations provide the user with information that is occluded by physical world objects, such as the interior of a house seen from the outside or subsurface pipes (Figure 1.1, Right).

In particular, outdoor applications in the industrial field of *Architecture, Construction and Engineering (ACE)* can benefit from an AR interface presenting the aforementioned information on-site. Usually, on-site operations are cost-intensive and companies aim to reduce outdoor operation times. AR can help to support the reduction by providing the spatial relationship between the data and the user's environment.

In 1994 Milgram et al. visualized the relationship between virtual and physical information in the Reality-Virtuality (RV) continuum [91]. Depending on the amount of physical (real) information and virtual information, they differentiate between Virtual Reality (VR), Augmented Virtuality (AV), Augmented Reality (AR) and pure Reality.

In addition to the relationship between real and virtual information in the AR-VR continuum, in 1997 Azuma defined

> "AR as systems that have the following three characteristics:
>
> 1. Combines real and virtual
> 2. Interactive in real time
> 3. Registered in 3-D" [8].

Back at the time when Azuma formulated these requirements, it was still computationally expensive to achieve them. Only the application of expensive and powerful computing devices allowed addressing all characteristics in one system. This limitation prevented the ubiquitous deployment of Augmented Reality for conventional users due to high cost, bulky equipment and limited availability. In recent years, with the increasing computational power of even small devices, omnipresent hardware, such as mobile phones and tablet computers, have grown powerful enough to fulfill Azuma's requirements. These further developments have worked towards a ubiquitous experience of the mixture of physical and virtual information and opened new fields of application, such as entertainment or advertisement, but also various professional applications. One central challenge that all these applications have in common is to find an adequate way of visualizing the available information. In order to address this challenge, it is important to understand the process of visualization in general.

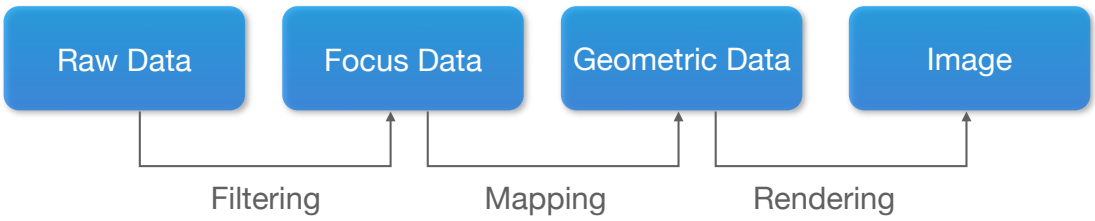


Figure 1.2: Information visualization pipeline.

1.2 Visualization

There are various definitions describing the term *visualization*. A traditional definition is given by the Oxford Dictionary[1]:

1. "form a mental image of; imagine
2. make (something) visible to the eye".

Other researchers defined the term more in relation to computer-supported visualization. For instance, Haber and Naab defined visualization as:

> "The use of computer imaging technology as a tool for comprehending data obtained by simulation or physical measurement by integration of older technologies, including computer graphics, image processing, computer vision, computer-aided design, geometric modeling, approximation theory, perceptual psychology, and user interface studies." [47]

In general, visualization can be described as the process of converting abstract data into a visual representation that is comprehensible by a human observer. The visualization process itself is often described step-by-step in one of the various versions of the *visualization pipeline* [47].

One version defines the pipeline consisting of three main steps: filtering, mapping, and rendering as shown in Figure 1.2. In this concept, the first step, *Filtering*, is a data-to-data mapping converting *raw data* into *focus data*, for instance by producing a smaller subset of the raw data set. *Mapping* is the second step and uses the focus data to generate geometric information. An example are data points that are mapped to 2D points or a line with a specific color. The last step is the *Rendering* of this geometric data to produce a 2D image that can be display on an output device.

1.3 Visualization in Augmented Reality

In contrast, visualization in AR is usually defined in a different way. Available definitions focus more on the fact that not only virtual data is mapped to a visual representation, but also spatial relationships between the physical world and raw (virtual) data, and a composition of them is required to generate the final 2D image. In general, visualization

[1] `http://oxforddictionaries.com`

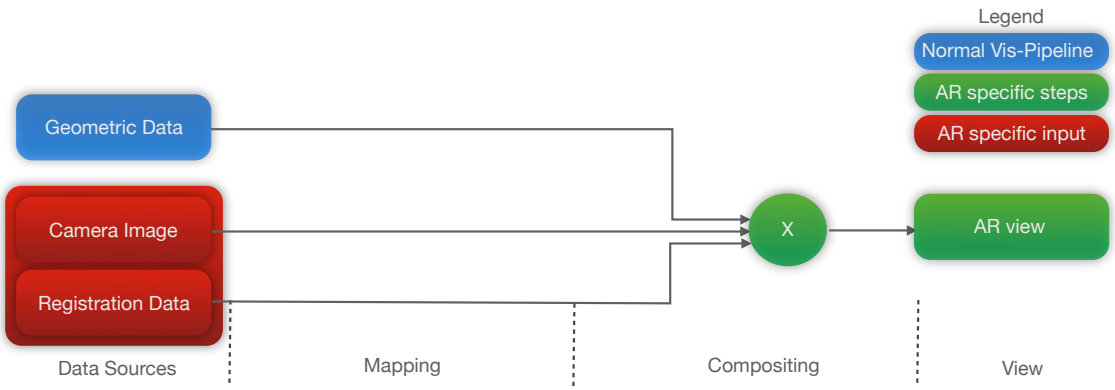


Figure 1.3: AR Visualization Pipeline illustrating a naïve overlay using predefined geometric data, a camera image and registration data for creating the composition.

in AR refers to the first item of characteristics from Azuma's list; the combination of real and virtual information.

In order to apply the general visualization pipeline for AR visualizations, it has to be adapted. By adding registration information, a camera image of the physical world (in video-based systems) and a composition step to the original pipeline, we can adapt it to reflect the characteristics of AR visualization in the AR visualization pipeline (Figure 1.3).

At first glance, the implementation of these additional steps seems to be straightforward if the registration between virtual content and the physical world representation is known (for instance in terms of a camera transformation matrix). In this case, data can be combined by simply overlaying the registered virtual content to the user's view. However, in a lot of situations a composition implemented with such a naïve overlay can lead to serious perceptual problems that may prevent the user from comprehending the visualization.

For instance, one of the problems that often arises in AR when visualizing non-existing objects using a naïve overlay, is incomplete scene integration. Incomplete scene integration refers to situations where important perceptual cues are missing. This is a problem, since humans need these cues to understand spatial relationships. If, for example, the composition method does not take the occlusions between virtual and physical objects into account, this will result in a wrong perception of the depth ordering. The virtual objects will always be seen as being in front of the physical world objects. Figure 1.4 (Left) demonstrates this problem within a planning application. As-planned lamps are superimposed on video images in order to enable the user to experience their appearance in the physical world. The incomplete scene integration of virtual and physical objects leads to the wrong perception of floating virtual lamps.

A similar problem occurs when using a simple overlay to visualize information that is naturally invisible. In Figure 1.4 (Middle), we show several subsurface pipes visualized in an X-Ray view. Since the pipes are just naïvely superimposed on the video image, important natural depth cues are missing. Instead of being perceived as subsurface, the pipes seem to float over the ground and it would be difficult to judge how deep they were buried.

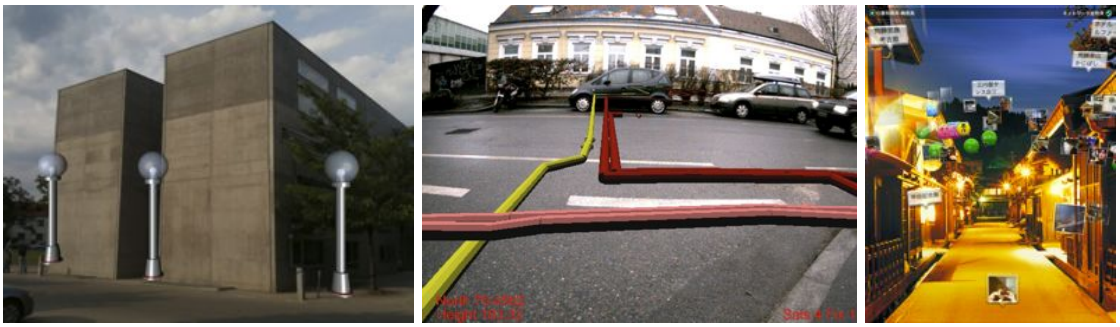


Figure 1.4: Visualization problems caused by naïve composition functions. Left) Virtual planned lamps are perceived to be located in front of the house not next to it. Middle) Virtual pipes seem to float over the ground. Right) In this touristic scenario the naive overlay of sight labels produces information clutter.

Another problem that can be found in AR visualization is information clutter. This problem often appears when visualizing meta information with a naïve overlay method. In Figure 1.4 (Right) we show an example of the visualization of labels indicating sights in a city. In addition to the already complex physical world, the large amount of meta-information produces a completely cluttered composition. The user has difficulty in reading all the labels and hardly understands the relationship between the meta information and the physical world.

These examples emphasize the need for sophisticated composition methods to facilitate comprehension. This work will focuses on how to adapt composition methods for AR to these needs. To adapt the composition methods, it is important to understand the way how information is combined technically. This differs mainly depending on the display device. In *Optical-See-Through (OST)* devices, the virtual content is superimposed on the user's view of the physical environment by using a semi-transparent mirror. In contrast, *Video-See-Through (VST)* devices combine the virtual content with a video image of the physical environment before it is displayed to the user. Along with this technical difference between OST and VST devices, there is a difference in the amount of influence on the implementation of the composition function. In OST systems the combination of virtual and physical content is technically complicated to modify [73]. In contrast, the explicit processing step for combining real and virtual information in VST systems provides more flexibility. This flexibility allows one to influence the combination of virtual and physical content, a precondition for addressing perceptual problems. This advantage of VST was also mentioned by Azuma:

> "Video see-through may ultimately produce more compelling environments than optical see-through approaches."[8]

From Azuma's statement, two main questions arise:

- What characterizes a compelling visualization in AR?
- How can we achieve a compelling visualization?

Several research groups have addressed the problem of achieving compelling visualizations in AR. Some of them investigated how to integrate the *cue approach to depth perception* [38] into AR visualizations. Particularly, there is a lot of work that focuses on the description, extraction and integration of pictorial cues in the final AR composition. As described by Goldstein, pictorial cues are monocular cues that are a source of depth information given by a picture [38]. Pictorial cues that have been successfully applied in previous AR visualizations comprise shadows [49], occlusion cues [15], or atmospheric cues [84]. Other researchers integrated additional pictorial cues such as virtual shadow planes, color encoding or top-down views [131], or a virtual tunnel tool [10].

Another way how researchers approached compelling visualizations in AR is by reducing information clutter. For instance, Livingston et al. applied information filtering to reduce the displayed content in the AR visualization [84], and Kalkofen et al. investigated Focus&Context techniques in AR in order to display virtual information only in a dedicated area of the field of view [63].

Other research groups focused on photo-realistic Augmented Reality with the main goal to render virtual objects with the same visual appearance as the physical world objects [77] and simulating image processing artifacts [75]. Completely in contrast are non-photorealistic renderings that display the physical environment with the same non-realistic visual properties as the virtual objects [33].

All these research directions have the same goal of increasing the convincibility of the AR visualization by influencing the composition method. However, they diverge in the way how they accomplish this. Apparently, there are two main directions:

- Presenting **comprehensible** compositions by:
  - Achieving a convincing scene integration
  - Including additional graphical cues
  - Information filtering
- Minimizing the **visual difference** between virtual and physical content with:
  - Photo-realistic AR
  - Non-photorealistic AR.

The list shows that either comprehensible compositions or the indistinguishability between virtual and physical content was important for visualization research in AR. By implementing methods that address these characteristics, researchers were able to achieve compelling visualizations.

Indistinguishability is important for applications that focus on appearance, such as design decisions. In contrast, outdoor applications in the industrial sector have a higher need of a comprehensible presentation of content. Since a lot of industrial applications can benefit from an AR interface and most of our research took place in the context of industrial application in outdoor environments, we focus in the following on providing *comprehensible visualizations in AR*. Furthermore, comprehensible visualization are the foundation for understanding the presented content and for more advanced techniques such as achieving photo-realistic AR renderings.

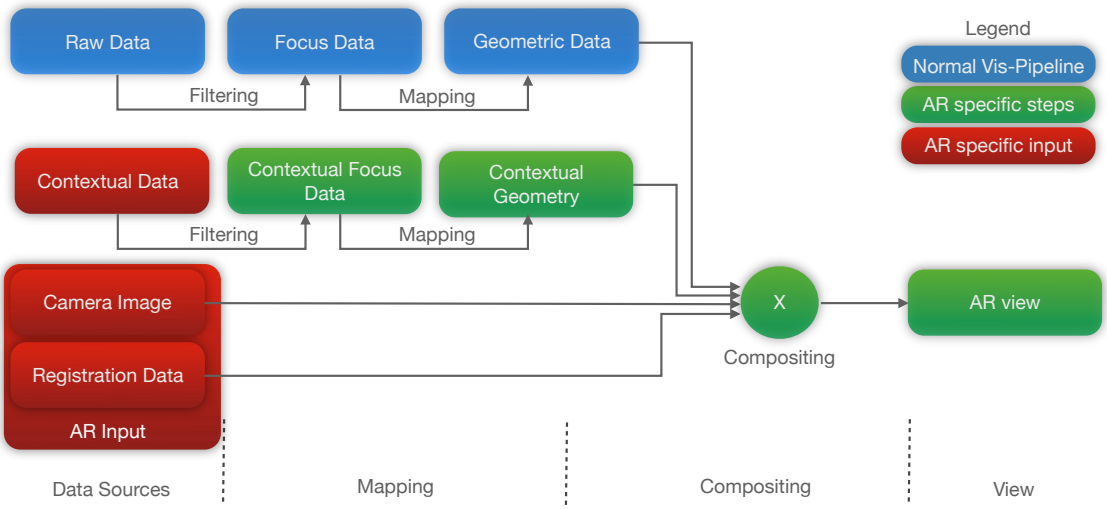


Figure 1.5: Comprehensible AR Visualization Pipeline.

By having a closer look at the techniques that address comprehensibility in AR visualization, we see that the simple AR visualization pipeline (Figure 1.3) has to be adapted to address the problems of comprehension. Techniques that aim to increase the comprehension convert the virtual data into more comprehensible data by either enriching the presentation with information about the physical world or by reducing the presented information in an adequate way (Figure 1.5).

1.4 Problem Statement

Previous research in the area of comprehensible AR visualizations showed that there are three main problems that keep AR visualizations from being comprehensible.

- Missing natural pictorial cues from the physical world: Due to missing physical pictorial cues, a composed scene may be easily misinterpreted.
- Insufficient natural pictorial cues: The available natural pictorial cues may be not sufficient for interpreting the spatial relationship (layout), especially if the composed content appears unfamiliar to the observer (e.g X-Ray vision).
- Information clutter: Complex data can be subject to information clutter and self-occlusions.

These problems arise directly from the mixture of virtual and physical information. Other interpretation problems may occur due to registration or data inaccuracies, but they are not addressed in this work. In the following, we will explain each of these three problems more in detail.

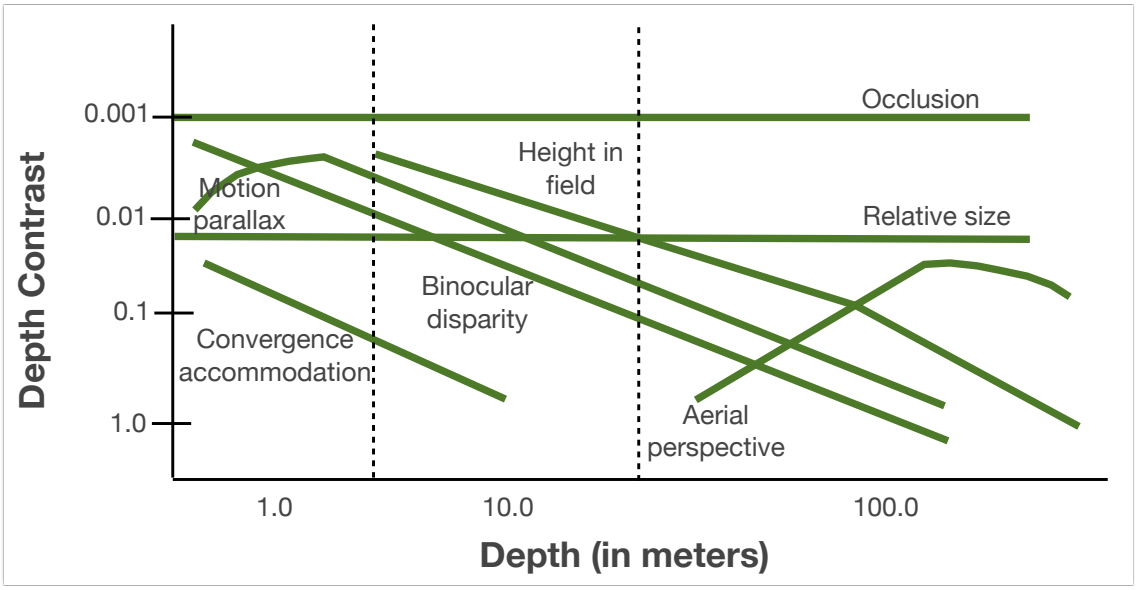


Figure 1.6: Pictorial depth cues. Cutting mapped different cues to their effectiveness to transfer depth in different distances. The depth contrast describes how effective they are. The depth on the x-axis describes in which distances they are working [23].

1.4.1 Missing Physical Pictorial Cues

Missing important physical pictorial cues lead to an incomplete scene integration between virtual content and physical world. In Figure 1.4 (Left), we show objects being rendered on top of the video image without taking care of a comprehensible composition. This provokes scene misinterpretation: Because of the missing occlusion cues, both lamps in the back are perceived as floating over the ground.

Cutting defined a set of important pictorial cues that help humans to interpret scenes and their spatial layout [23]. These cues include relative size, height in visual field, aerial perspective, motion parallax, accommodation and occlusion (Figure 1.6). Although several of these pictorial cues are already given by the rendering (such as relative size), in this composition, occlusion cues are missing. In order to avoid the problem of scene misinterpretation, we suggest that a convincing AR visualization should try to provided as many pictorial cues as possible in the composition. Since occlusion has been identified to be the most important pictorial cue [23], we primarily focused our research on addressing the problem of missing pictorial cues on a convincing occlusion management.

Generally there are two main goals that are part of occlusion management. Physical correct visualizations with front objects making back objects invisible; and X-Ray visualizations with partially translucent front objects that allow one to look inside or behind front objects. With these goals of occlusion management in mind, there are two main issues that have to be addressed:

- Determining the depth order of virtual and physical objects.
- Determining which information from the occluding object or the hidden object

should be preserved to make the scene visually plausible but also make hidden objects of interest visible.

Apparently, comprehensibility is in direct conflict with the visibility of hidden objects, since this kind of X-Ray view is not physically plausible. To address this problem, convincing occlusion management has to find the best compromise between correct occlusion cues and object visibility.

Several research groups introduced methods to determine the **depth order** of virtual and physical objects. In the early work of Breen et al., the researchers assumed to have a model of the physical world [15]. Fischer et al. applied a Time-of-Flight (TOF) range sensor to determine the depth order of virtual and physical objects [34].

The second issue, the decision of which information will be preserved in the final rendering, was also already tackled by some research groups. For instance, in the work of Avery et al. [6] and Kalkofen et al. [63], the researchers aimed at providing missing visual cues from the physical world by extracting these cues from the video image of the physical environment. Nevertheless, so far only a few of the available visual cues, such as edges, were extracted and preserved. There are still situations where these approaches will fail, since too few natural cues are present.

One of the main goals of this thesis is to achieve a seamless integration of virtual content in the physical scene by addressing the problem of missing physical cues. For this purpose, we will analyze available data sources such as video images and databases to extract as many physical cues as possible to achieve a convincing composition.

1.4.2 Insufficient Physical Pictorial Cues

Even if we can achieve a convincing scene integration by adding missing physical cues, these natural pictorial cues can still be insufficient for a complete interpretation of the AR visualization. This issue emerges in particular in outdoor scenes with a large working area as several depth cues are only powerful in the near field (Figure 1.6). Another issues with scenes containing virtual objects is that they do not always follow natural laws. These unrealistic scenes comprise X-Ray visualizations, such as the visualization of subsurface objects, or the visualization of floating objects. In these cases the requirements for using the pictorial cue of "height in visual field" are not fulfilled. According to Cutting [23] this visual cue requires

"Gravity, or the bases of objects are on the ground plane."

and

"Opacity of objects and of the ground plane"

which is by definition - objects that do not follow physical principles or float - not valid. But the lack of this specific visual cue is critical, since it is one of the few static pictorial cues that is assumed to provide more than just an ordinal depth measurement [23]. Without them it is nearly impossible to judge the depth of these kind of objects because the spatial relationship between physical world and virtual objects cannot be established by the user.

Figure 1.7: Adding virtual pictorial cues to an X-Ray visualization: Left) The virtual object is visualized in the X-Ray AR view without additional hints. Without additional depth cues it is nearly impossible to judge the depth of the virtual object in relation to the physical ground. Right) By rendering a virtual excavation along the virtual object, additional depth cues are provided.

Therefore, we suggest to create additional virtual cues that communicate the connection between physical world and virtual objects. These additional cues should help users in tasks, where they have to interpret spatial relationships in absolute or relative measurement between virtual information and the physical world. This is from high interest for industrial applications where the depth of objects plays a major role, such as the estimation of depth of an occluded pipe as shown in Figure 1.7 (Right).

The problem of insufficient physical pictorial cues for challenging scene interoperation was addressed by different research groups. For instance, Livingston et al. [84], Feiner et al. [30] and Coffin et al. [21] created additional visual cues that aim to support the user in interpreting AR visualizations. However, this information is usually created for selected test objects or based on user interaction. So far there exists no work on how this kind of information can be automatically derived from professional data sources as required for professional applications such as GIS. Another open research question is how this kind of visualization can be maintained in a bi-directional way, which is in particular interesting when interacting with big data sets or commercial data sets.

1.4.3 Information Clutter

Another issue that leads to a reduced comprehensibility of AR visualizations is *information clutter*. This issue often occurs in the visualization of meta data and especially in the visualization of big and complex data outdoors in AR as demonstrated in Figure 1.8 (Right). Several characteristics that are specific to AR visualizations contribute to this problem:

1. The view of the user is usually fixed to his own viewpoint, since the visualization is registered to the physical world. This makes it difficult to explore complex data from different viewpoints as possible in VR environments.

2. Without depth knowledge about the physical environment, meta information is shown even for objects that are not in the field of sight (Figure 1.8,Left).

Figure 1.8: Cluttered visualization. Left) Next to labels that describe the red building, labels that are not inside the field of sight are visualized. Right) Cluttered visualization of a 3D mesh.

3. The physical outdoor environment is already complex by nature compared to VR environments. This makes it even more complicated to integrate complex information.

4. Complex information may be subject to self-occlusion

5. When visualizing multiple data sets, it is complicated to compare them directly due to change blindness and data complexity.

As shown in Figure 1.8 (Right) an already complex outdoor scene is combined with complex virtual data. The image of a construction site is superimposed with 3D data from a previous point in time. The complexity of the augmentation prevents the user from understanding the presented information.

The problem of the visualization of complex data in AR was only addressed by a small amount of research groups. For instance, Schmalstieg et al. introduced an approach to manage complex augmented reality models [114]. Julier et al. proposed a system for information filtering in AR [61]. On the other hand Kalkofen et al. used Focus&Context techniques for information filtering in AR. Within their CityLens project, Nokia proposed an information filtering method called *Sightline*[2] that removes unnecessary information. But so far, no one has shown how to visualize complex information that contains more than three dimensions or how to compare multiple data sets such as required for professional applications in the context of construction site monitoring.

1.5 Hypotheses

In this thesis we investigate problems that arise in creating comprehensible visualizations in AR. From the issues discussed above, we derive the following hypotheses:

- H1: AR visualization requires a seamless integration of virtual and physical information that maintains existing visual cues of the physical environment to achieve a convincing composition.

[2] `http://betalabs.nokia.com/trials/nokia-city-lens-for-windows-phone`

- H2: These physical visual cues can be automatically derived from different data sources such as video images, GIS data bases or combinations of both.
- H3: Additional virtual pictorial cues that convey the spatial relationship between virtual content and the physical environment can be automatically created from georeferenced data.
- H4: Filtering methods reduce self occlusions and information clutter in scenes visualizing single 3D datasets.
- H5: Automatic methods for abstracting complex information in combination with interactive focus and context techniques reduce information clutter and self occlusion for multiple complex 3D datasets.

1.6 Contribution

The research that was conducted within this thesis contributes to the field of AR in general as well as to the field of visualization techniques for outdoor AR and perception in AR. Furthermore by implementing selected methods in the context of professional industrial applications, it provides contribution to the field of industrial AR as well.

We contribute a set of visualization techniques that work towards comprehensible visualization and demonstrate how these techniques can improve the in-situ visualization of professional information in civil engineering and architecture. The main contributions are:

- Techniques that automatically **preserve physical visual cues** for a seamless integration of virtual and real information from different available data sources.
- Demonstrating how these **physical cues improve the perception of depth order** in an industrial AR application.
- Methods that **create additional virtual pictorial cues** from georeferenced data automatically or interactively.
- Demonstrating how these **virtual cues are applied in industrial AR applications** such as maintaining subsurface infrastructure or aerial navigation for construction site monitoring.
- Techniques that **automatically abstract complex information** for a comprehensible AR visualization and furthermore Overview&Detail techniques that combine abstract and complex data.
- Demonstrating how these abstraction techniques can be used for **visualizing time-oriented data for construction site monitoring**.
- Additionally, we developed a set of **prototypes** that allow us to apply our visualization methods visualize different kind of industrial data, such as GIS data and 3D reconstructions from construction sites.

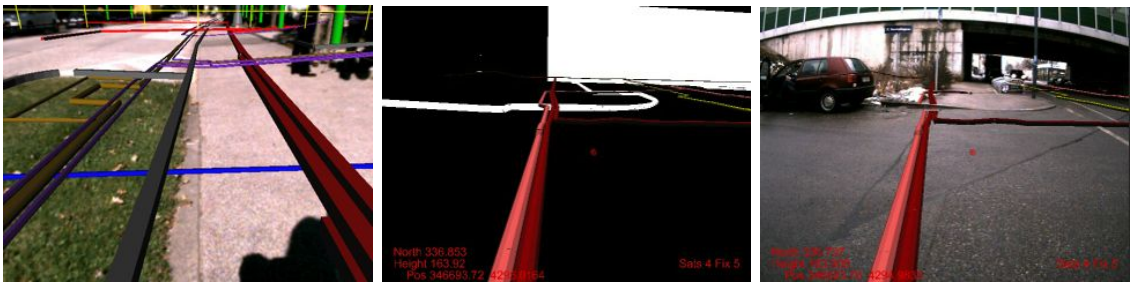


Figure 1.9: Sparse physical cues used for occlusion management. Left) Physical world objects such as curbstones represented in the GIS database are added to the visualization for providing occlusions cues. Middle) Sparse representations are used to create a occlusion mask (in white). Right) The same occlusion mask is used for occlusion culling to provide occlusion cues.

One of the main challenges in AR visualization is that, in contrast to other visualization environment (for instance virtual reality), the available knowledge about the content that is visualized varies. While we have an almost complete description of the virtual content, exact 3D information about the physical world is often only partly available. Visualization techniques differ depending on the available knowledge about the physical world scene:

- No knowledge about the physical world scene
- Sparse knowledge about the physical world scene
- Semi-dense representation of the physical world scene

If no further knowledge about the physical world scene is available, visual cues can be derived from the video image of the physical world. This information provides physical cues that support the seamless integration of virtual content in the image of the physical world.

In cases where sparse knowledge about the physical world is available, physical cues, such as occluding curbstones, can be derived by using filtering and transcoding techniques. However, such derived physical cues are often too imprecise to achieve a seamless integration as shown in Figure 1.9 (Left). In order to address this problem, sparse data can be combined with registered images of the physical world. The result of this combination are dense representations that are more precise in comparison to using purely sparse data. It is important to note, that this approach requires a perfectly registered AR system to achieve adequate results that can be used for visualization purpose.

In applications where a semi-dense representation of the physical world is available, for instance given by aerial 3D reconstruction, it is important to extract the important information and preprocess the data to achieve a meaningful visual representation that support the user's scene understanding.

1.7 Results

In this thesis, we show that it is possible to achieve a seamless integration of hidden virtual content into the user's perception of the physical world by maintaining important and

Figure 1.10: Seamless integration of virtual content in an X-Ray visualization of a virtual room inside a physical building. Left) Simple overlay. The room is perceived to be located in front of the building. Right) Preserving occlusion cues from the video image. The virtual room is perceived to be located inside the building.

selected cues extracted from the video image representations of the physical world. This combination should provide users with essential perceptual cues to understand the depth order between hidden information and the physical scene. To achieve this we group pixels into perceptually coherent image regions and compute a set of importance characteristics for each region. The importance measurement is then used to decide if an image pixel is either preserved or replaced by virtual information (Figure 1.10).

Afterwards, we show that there are applications where the AR visualization needs to provide additional cues about the relationship between virtual and physical objects in the scene. For instance, to judge the depth of an occluded object that is visualized in an X-Ray AR visualization, it is complicated for the user to understand the relationship between virtual and physical objects (Figure 1.7, Left). For these cases we show how to automatically derive additional depth cues from professional databases and visualize these cues in an adequate way (Figure 1.7, Right). Furthermore, we show that by maintaining data consistency between visualized additional information and information in the database, it is possible to allow interactive modifications of the abstract information in the databases while providing consistent additional visualization cues.

Finally, we discuss problems of visualizing complex data in AR. As an example we implemented an approach for visualizing time-oriented data of dynamic scenes in an on-site AR view. Such visualizations of complex data have special challenges in comparison to the visualization of arbitrary virtual objects. One problem is that the data may occlude a large part of the real scene. Additionally, the data sets from different points in time may occlude each other. Thus, it is important to design adequate visualization techniques that provide a comprehensible visualization.

1.8 Collaboration Statement and Selected Publications

This section gives an overview of the publications the author contributed to and describes the contribution of the author in detail.

For the following papers form the main contributions of this thesis.

- **Image-based Ghostings for Single Layer Occlusions in Augmented Reality** Zollmann Stefanie, Kalkofen Denis, Mendez Erick, Reitmayr Gerhard, In Proceedings of the International Symposium on Mixed and Augmented Reality 2010 [134]

  This publication contributes mainly to the seamless integration of virtual and physical content. It forms the main content of Chapter 4. The main idea of this publication is to analyze video images to extract physical pictorial cues. The extracted cues are then used in the composition method to preserve important selected cues. For this paper, the author was the main contributor to idea development, implementation and paper writing. The co-authors were contributing with valuable discussions, implementation suggestions and with paper writing.

- **Dense Depth Maps from Sparse Models and Image Coherence for Augmented Reality**, Zollmann Stefanie, Reitmayr Gerhard, In Proceedings of VRST2012 [135]

  This paper describes methods for extracting dense depth data from sparse depth information such as provided by GIS databases. The content of this publication forms the main content of Chapter 5. After presenting a method for combining sparse depth features with image data from an AR system, the paper also describes various applications where this extracted dense depth data can be used. These applications comprise: occlusion management, shadow visualization and interactions with the physical objects. The author of this thesis was the main contributor for this paper and implemented the most parts of this system as well as the applications. Gerhard Reitmayr was supervising and contributing to the implementation with valuable implementation suggestions and improvements and was also contributing to writing.

- **Comprehensible and Interactive Visualizations of GIS Data in Augmented Reality** Zollmann Stefanie, Schall Gerhard, Sebastian Junghanns, Reitmayr Gerhard, In Proceedings of International Symposium on Visual Computing, 2012 [136]

  This paper contributes to the creation of virtual pictorial cues from a professional GIS database that forms the main content of Chapter 6. By using filtering and transcoding functions, we derive these important pictorial cues automatically from a database. Additionally, by introducing an additional data layer, we are able to maintain data consistency between visualized additional information and information in the database. This allows interactive modifications of the original data even if the user is working with the additional data. This consistency is especially important for professional applications working with this kind of databases. The author was the main contributor for the development of visualization techniques, system design, implementation and the integration of data and interaction methods as well as for the paper writing. The co-authors were mainly contributing to design of ideas and by providing data and system components such as the tracking technology.

- **Interactive 4D Overview and Detail Visualization in Augmented Reality** Zollmann Stefanie, Kalkofen Denis, Hoppe Christof, Kluckner Stefan , Bischof Horst, Reitmayr Gerhard, In Proceedings of ISMAR'2012, IEEE, 2012 [133]

This paper contributes to the visualization problem of information clutter and forms the main content of Chapter 7. By implementing an automatic method to derive an abstract data representation and applying Overview&Detail methods to inspect this information, this paper shows how to visualize time-oriented information in an AR environment. For this paper the author was the main contributor for the design of visualization and interaction techniques and the implementation as well as for the paper writing. The co-authors were contributing to the design of ideas, the 3D data reconstruction and writing.

- **FlyAR: Augmented Reality Supported Unmanned Aerial Vehicle Navigation** Zollmann Stefanie, Hoppe Christof, Reitmayr Gerhard, Submitted to ISMAR'2012, IEEE, 2013

  In this paper, we propose a set of visualization techniques for supporting the navigation and flight management of aerial vehicles. A set of additional graphical hints support the depth estimation and help the user to avoid critical situations during a flight session. This paper contributes to Chapter 6. The author of this thesis was the main contributor for developing and testing the visualization techniques, the data processing over the network and writing the paper. The co-authors took part in active discussions for developing the idea and improving the paper. Furthermore, they supported the testing with the mobile robot and provided the functionality for data exchange on the aerial vehicle site.

The following list contains publications that describe the systems and tracking technology that were used to implement and test the visualization methods described in this thesis.

- **Bridging the Gap between Planning and Surveying with Augmented Reality User Interfaces** Schall Gerhard, Zollmann Stefanie, Reitmayr Gerhard In Mobile HCI 2011 Workshop *Mobile Work Efficiency: Enhancing Workflows with Mobile Devices*, 2011-August [112]

- **Smart Vidente: Advances in Mobile Augmented Reality for Interactive Visualization of Underground Infrastructure** Schall Gerhard, Zollmann Stefanie, Reitmayr Gerhard, Personal and Ubiquitous Computing, 1 – 17, 2012 [113]

  Both publications describe a system for outdoor subsurface infrastructure AR visualization. The system was used to test most of the X-Ray visualization techniques in this thesis. The author made contributions to the concrete idea of the system, and main contributions for system implementation, interaction techniques and visualizations. The initial ideas for the system came from Gerhard Schall and Gerhard Reitmayr, who also mainly contributed to the registration and tracking technology of the systems

- **Construction Site Monitoring from Highly-Overlapping MAV Images** Kluckner Stefan , J. Birchbauer, C.Windisch, Hoppe Christof, Irschara Arnold, Wendel Andreas, Zollmann Stefanie, Reitmayr Gerhard, Bischof Horst, In Proceedings of the IEEE International Conference on Advanced Video- and Signal-based Surveillance (AVSS), Industrial Session, 2011-August [76]

- **Photogrammetric Camera Network Design for Micro Aerial Vehicles** Hoppe Christof, Wendel Andreas, Zollmann Stefanie, Pirker Katrin, Irschara Arnold, Bischof Horst, Kluckner Stefan, In Proc. Computer Vison Winterworkshop, Mala Nedelja, Slovenia, 2012, [54]

- **Image-based As-Built Site Documentation and Analysis - Applications and Challenges** Stefan Kluckner, Juergen Hatzl, Manfred Klopschitz, Jean-Severin Morard Christof Hoppe, Stefanie Zollmann, Horst Bischof, Gerhard Reitmayr, In Proceedings DAGM, Workshop Computer Vision in Applications, 2012

  These three publications contain implementation details about a system for construction site monitoring. Whereas these publications describe how to efficiently capture images with an aerial vehicle that can be used for 3D reconstruction, the main contribution of the author was to integrate this kind of data into an AR visualization system to visualize this kind of data directly in place. This data was then used for testing the methods for visualizing complex data. The main contribution of the author was here to implement the AR system but also to allow a synthesis of virtual views of the construction site for optimizing the flight planning.

- **Incremental Superpixels for Real-Time Video Analysis** Steiner Jochen, Zollmann Stefanie, Reitmayr Gerhard, 16th Computer Vision Winter Workshop, Andreas Wendel Sabine Sternig Martin Godec, 2011-February [120]

  This publication describes an incremental approach for real-time oversegmentation of incremental video data such as panoramic images. This method was also was implemented to support the image analysis in Chapter 4. The author was developing the main ideas, supervising the first author in implementing the system, and was a main contributor to paper writing.

Chapter 2

Background

Contents

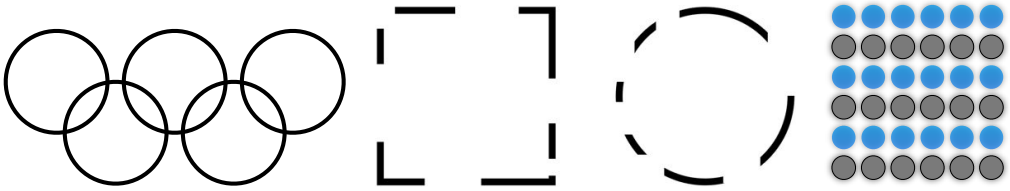
2.1 Perceptual Background

Supporting comprehension in AR visualization and achieving convincing compositions of virtual and physical information is only possible if we understand the way the Human Visual System (HVS) processes information. In general, humans build perceptional models from the environment during a learning process starting in early childhood [38]. These models contain a set of assumptions that support a fast understanding of objects and their relationships. Every time when humans want to understand a new situation and spatial relationships, they apply these assumptions. Researchers and psychologists have developed a set of models and theories that aim to describe laws and assumption important for human perception.

Only if we reflect these perceptual processes in AR visualization techniques, we are able to convince the user that the virtual data belongs into the physical world. For this purpose, we have to understand the perception process and have to include characteristics of these models into the visualization process. Since existing perceptual theories are the main foundation for this goal, we will start with describing the characteristics of important perceptual models and theories. Later on, we will discuss how we can include them to improve the visualization.

2.1.1 Gestalt laws

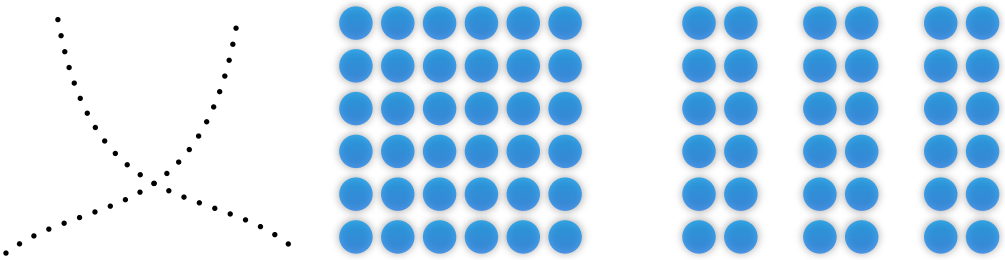
In the beginning of the 20th century, psychologists developed the *Gestalt theory*. One of the main ideas of this theory is that humans do not only use single entities to process their perception but use the complete structure [38]. These complete structures are called



(a) Law of Praegnanz.

(b) Law of Completion.

(c) Law of Similarity.



(d) Law of Good Continuation.

(e) Law of Proximity.

Figure 2.1: Gestalt laws.

Gestalt (German word for shape). The aim of the Gestalt theory was to find the rules that are used to process complex structures:

- The Law of Prägnanz or good shape. Humans tend to perceive simple shapes instead of complex ones. Complex structures are likely to be divided into simple shapes. For instance, we tend to perceive the object in Figure 2.1(a) as a set of circles instead of one complex shape.

- Law of Completion. Incomplete information, such as gaps in objects or partially occluded objects are likely to be automatically completed. Missing information is filled with information that is likely to be missing. This is depicted in Figure 2.1(b) where we still see a rectangle and a circle, even though some parts are missing.

- Law of Similarity. Objects that are similar in color or shape are likely to be grouped to one bigger entity. This is demonstrated in Figure 2.1(c) where the grey circles are perceived as one group and the blue circles are perceived as one group.

- Law of Good Continuation. This law describes the tendency to perceive lines and curves as following a defined direction, instead of assuming abrupt changes in directions. For instance, in Figure 2.1(d), we have the impression that there are two lines crossing in the middle, instead of two lines changing their direction.

Figure 2.2: Law of completion in Illustrative X-Ray visualization. Left) The front part of the leg is not rendered, but completed by our perception (Bruckner et al. [16]). Right) Comparing traditional volume rendering for Illustrative X-Ray visualization with screen-door transparency. Missing information created by the screen-door transparency is completed in such a way that we perceive the complete gecko (Viola et al. [126]).

- Law of Proximity or Nearness. Objects in close proximity are likely to be perceived as a bigger entity (Figure 2.1(e)).

An understanding of these laws helps us to take advantage of them. For instance, the law of completion allows a partial replacement of physical image information with virtual information, while the user is still able to understand the structure of the physical world. This phenomenon is used in technical illustration, in medical studies, and in illustrative renderings to provide an X-Ray view [16, 126]. Occluding structures are only preserved partially to provide depth cues, the law of completion helps the user to complete the incomplete information (Figure 2.2).

In addition, we can improve the visual coherence in AR visualization by considering these laws for the development of visualization techniques. For instance, the law of similarity should be considered when manipulating the appearance of virtual or physical objects. If different manipulations are applied to to similar objects, their grouping can get lost.

2.1.2 From 2D to 3D

Another important aspect of human perception is how humans process 2D information to build 3D representations from it. In 1982, Marr proposed a computational approach explaining object perception [88]. His approach is based on a series of processing steps within each step transfers information into different representations. Starting with the input image defined by the perceived intensities, the 2D image is transfer into a primal sketch consisting of low level features such as zero crossings, blobs, edges and other features. The primal sketch is used to create the 2.5D sketch that represents the orientation and depths of primitives as well as discontinuities in orientations and depths. Finally, a 3D model representation of the scene can be created from the 2.5D sketch. Thereby the 3D model consists of 3D primitives that are hierarchically organized with spatial relationships.

Table 2.1: Nine sources of information about spatial relationships. (adapted from [23]).

Source of Information	Assumptions	Implied Measurement Scale
All	Linearity of light rays, Luminance or textual contrast, Rigidity of objects	-
Occlusion	Opacity of objects, Helmholtz's rule, or good continuation of the occluding object's contour	Ordinal.
Height in the visual field	Opacity of objects and of the ground plane. Gravity, or the bases of objects are on the ground plane. The eye is above the surface of support. The surface of support is roughly planar. (In hilly terrain, use may be restricted to the surface directly beneath the line of sight to the horizon.)	Ordinal, perhaps occasionally better. In [24], Cutting and Vishton state an absolute depth measurement.
Relative size	Similarly shaped objects have similar physical size. Objects are not too close. Plurality of objects in sight. (Not familiarity with the objects, which denotes "familiar size").	Unanchored ratio possible, but probably ordinal.
Relative density	Similarly shaped objects or textures have uniform spatial distribution. Plurality of objects or textures in the field of view.	Probably ordinal at best.
Aerial perspective	The medium is not completely transparent. The density of the medium is roughly uniform.	Probably ordinal.
Binocular disparities	The distance between eyes. The current state of vergence. Unambiguous correspondences.	Absolute, but perhaps only ordinal
Accommodation	Complex spatial frequency distribution.	Ordinal at best.
Convergence	The distance between eyes. The current state.	Ordinal.
Motion perspective	A rigid environment. A spatial anchor of zero motion (horizon or fixed object).	Absolute, unanchored ratio. Perhaps only ordinal.

The knowledge about the steps applied by the *Human Visual System (HVS)* for creating 3D models from 2D information, provides details how information is processed and how the brain creates depth information from it. In order to create convincing AR visualizations, it is important that the user can derive the correct spatial relationships from the displayed content. Accordingly, visualization techniques should include information that is required for creating depth information.

2.1.3 Spatial Layout

It is important to note that Marr did not further defined the way how humans estimate depth in the processing step that creates a 2.5D representation. This question was investigated by several other researchers. For instance, in his work from 1997, Cutting defined nine sources of information used by humans to derive spatial relationships. He compared these cues according to their relative efficiency and described assumptions that have to be fulfilled to make them work (Table 2.1). He further introduced a measurement of depth contrast describing each cue's efficiency (Figure 1.6).

As described in his research, **Occlusion** is a cue that provides ordinal depth information. Occlusion as a cue can be used when objects are occluded or partially occluded by another one. In this case, we can derive that the occluded objects are located behind an occluding object. Occlusion is considered to be the most powerful depth cue. It works at

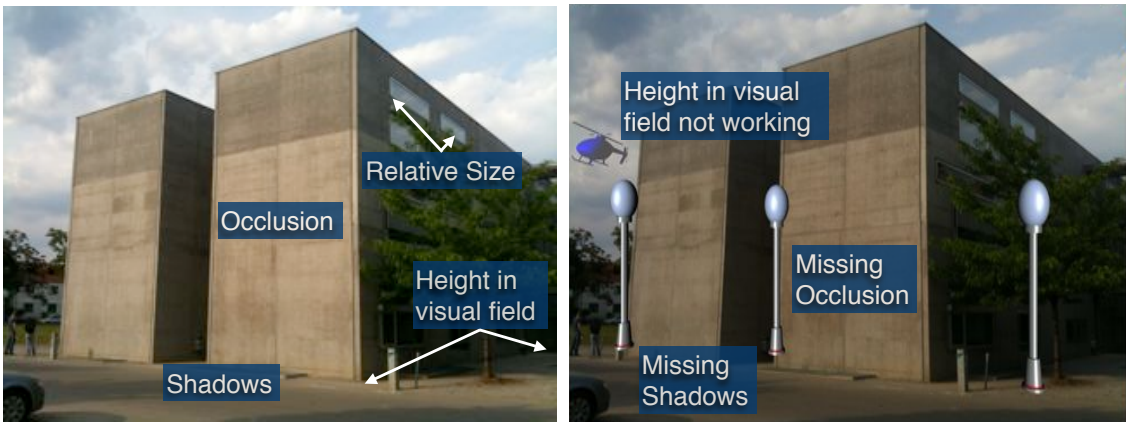


Figure 2.3: Pictorial depth cues in the physical world and in AR. Left) Pictorial depth cues in an urban scenario. Right) In naïve AR overlays important depth cues are missing and lead to misinterpretations.

all distances and has a higher depth contrast than other cues (Figure 1.6). It is important to consider that occlusion only works if objects in a scene are opaque. Apparently, for transparent objects there exists no occlusion. This emphasizes that occlusion cues are in particular a problem for X-Ray visualizations.

The depth cue **Height in visual field** describes that objects that are further away from the observer appear at a higher position in the visual field and are closer to the horizon. Cutting and Vishton assumed this cue to have the potential of yielding absolute distances, but only if a set of assumptions is fulfilled [24]. These assumptions include that the ground plane of objects in a scene is nearly planar and the objects have their base on this ground plane. This implies that absolute depth estimations for floating or subsurface objects is more complicated, since in this case the assumptions can not be fulfilled (Figure 2.3, Right).

Relative size and **Relative density** are two sources of depth information that are connected to each other. Objects that are further away appear smaller in the image. The same happens to textures, the density of the texture will appear to increase for objects that are further away. If the scene contains several similar shaped or textured objects this can be used as ordinal depth cue. Relative size is even considered to be able to provide relative measurements.

Aerial perspective refers to the fact that objects that are located further away loose their contrast and adjust more to the color of the atmosphere. Cutting consider this cue to be probably ordinal.

Other cues comprise *Binocular disparities*, *Accommodation*, *Convergence* and *Motion perspective*. Since these cues either only work in the near field, require stereo, or require the user to move during observing [45], they often do not apply for hand-held outdoor AR systems, such as used for industrial outdoor applications. Hence, we will not explain them in detail.

From photographs of the physical world we can use pictorial depth cues to derive depth information (Figure 2.3, Left). Unfortunately, some of these cues are not available

when using a naïve AR overlay (Figure 2.3, Right). Visualization techniques in AR should address these theoretical models and laws to avoid conflicting situations and improve the comprehension of the composition. In the next sections, we will discuss related work that include these laws and cues into their techniques.

2.2 AR Visualization Taxonomy

There is already a huge amount of related work that addresses perceptual issues in AR. In this section, we introduce a taxonomy to find a classification of visualization techniques in AR. This classification helps us to understand the similarities and difference between different existing techniques. In contrast to the work of Kruijff et al. that classified perceptual problems in AR [78], our classification focuses on the characteristics of the visualization techniques themselves similar to work of Elmquist et al. [28]. Elmquist et al. proposed a classification of 3D occlusion management techniques and used it to classify 50 different techniques. They used the taxonomy to identify areas that are not covered by existing techniques. Likewise, the introduction of a taxonomy for visualization techniques in AR can help us to identify gaps for outdoor visualization.

We define the space of AR visualization with a set of dimension. Within each dimension we can define the location of each visualization technique. This allows us to define a consistent language and classify the techniques. We found the following dimensions to be important:

- Addressed visualization problem
- Virtual Data Visibility
- Pictorial Depth Cues
- Filtering
- Abstraction
- Composition

Addressed Visualization Problem The common goal of all AR visualization techniques is to improve the comprehension of virtual content being integrated into the physical world. However, there are different aspects each visualization technique addresses to improve the comprehension. Some techniques focus more on achieving a convincing scene integration between virtual information and physical environment, others are focusing on supporting the depth estimation or reducing information clutter. These primary objectives are covered by this first dimension.

Domain: Scene integration, Supporting depth estimation, Reducing clutter
Example Techniques:
Achieving a convincing scene integration: Occlusion culling [15],
Supporting depth estimations: Virtual depth cues [84],
Reducing information clutter: Filtering [61].

Virtual Data Visibility Another aspect is the visibility of virtual data, since it has an influence which visualization problems have to be addressed. Virtual data can be occluded by physical objects or be directly visible. Especially the visualization of invisible data, so-called X-Ray visualization, is a popular visualization technique in AR and is

used for various applications, such as subsurface visualization or medical applications. As described in the last section, this kind of visualization has special challenges, since some of the natural depth cues are not working.

Domain: Occluded, Partially occluded, Visible

Example Techniques:

Occluded: Subsurface visualization in Vidente [110],

Partially occluded: Occlusion culling [15],

Visible: Tramlines [83].

Pictorial Depth Cues This dimension describes if depth cues are added to the visualization and their characteristics. We differentiate between physical and virtual pictorial depth cues. Regarding their characteristics, we define physical cues as cues that try to mimic or rebuild natural pictorial depth cues, such as occlusion or shadows. They can be computed from different information sources that contain contextual information about the physical and virtual world. For instance, edges can be extracted from a video image and used as natural occlusion cue. We define virtual cues to be graphical aids that are naturally not available in the physical world such as virtual scales, measurements or other graphical hints.

Domain: none, physical, virtual

Example Techniques:

None: Magic Book [42],

Physical: AR Ghostings [64],

Virtual: Cutaways [36].

Filtering The amount of filtering is another dimension that allows us to differentiate visualization techniques in AR. Simple AR visualization methods, usually apply no filtering and render a predefined geometric model in a naïve overlay. When it comes to the visualization of more complex data, an adequate filtering is required to avoid information clutter. In particular, information clutter is increased in outdoor environments due to the complexity of the physical environment. We can divide the dimension into techniques that use raw data and techniques that apply filtering.

Domain: Raw, Filtered

Characteristic Techniques:

Raw: Shadowplanes [10],

Filtered: Global filtering [61]

Abstraction Another way of reducing complexity is to move from a concrete representation to a more abstract representation. Abstractions allow reducing the amount of information of the data by finding a different representation that preserves only the relevant information. We differentiate between techniques that display the pure data and techniques that use an abstraction of the data.

Domain: Concrete , Abstract

Example Techniques: Most techniques display concrete information.

Compositing The way the virtual and the physical information is composed into the final AR view strongly depends on the objectives of the visualization. We differ between techniques that use a simple overlay, masking or information transformation. Simple overlay describes all techniques where the virtual content V is simply superimposed on the physical world representation P. The combination of video image P and virtual content V defines the final composition $C = V + P$. Masking techniques use a mask M to control which virtual and which physical information is visualized or omitted. The final composition is then given by $C = M * V + (1 - M)P$. The transformation of information is a composition technique that rearrange physical or virtual items in order to avoid cluttered views. These techniques create the final composition by $C = T_V(V) + T_P(P)$ and depend on the transformation of virtual content T_V and of physical content T_P.

Domain: Simple Overlay, Masking, Transforming Information

Example Techniques:

Simple Overlay: Shadow Planes [10],

Masking: Ghostings [6],

Transforming Information: Distorted Views [106].

Summary We use these dimension and their domains to classify the available visualization techniques from previous work. A visualization of the classification is depicted in Figure 2.4. In this overview, we show a graphical mapping between existing techniques and dimension of our taxonomy. For simplifying the classification, we removed the dimension abstraction, since we are not aware of techniques focusing on abstraction techniques in AR.

The graphical classification shows that there is a big amount of visualization techniques for X-Ray AR. We can further see that some dimensions have an equal distribution in their domains, while others seem to be clustered to one domain. For instance, the usage of physical, virtual and no cues is nearly equally distributed. In contrast, only a small amount of techniques apply a filtering techniques.

Another important aspect that becomes evident from our classification is the relationship between addressed visualization problem and data visibility, depth cues, filtering as well as compositing. Firstly, it seems that the most visualization techniques that support depth estimation use simple overlays of virtual depth cues and no filtering. Improving the depth estimation seems to be of interest for visible as well as for hidden information. Secondly, visualization techniques that aim to support seamless scene integration are used for hidden and partially visible virtual information. Apparently, there is no need to address scene integration for visible virtual information. In order to integrate hidden virtual information in the compositing, the most techniques apply a masking method.

Finally, our classification shows that filtering techniques and information transformation techniques are mostly used for reducing information clutter. Depth cues seem to be not used in techniques that focus on reducing information clutter.

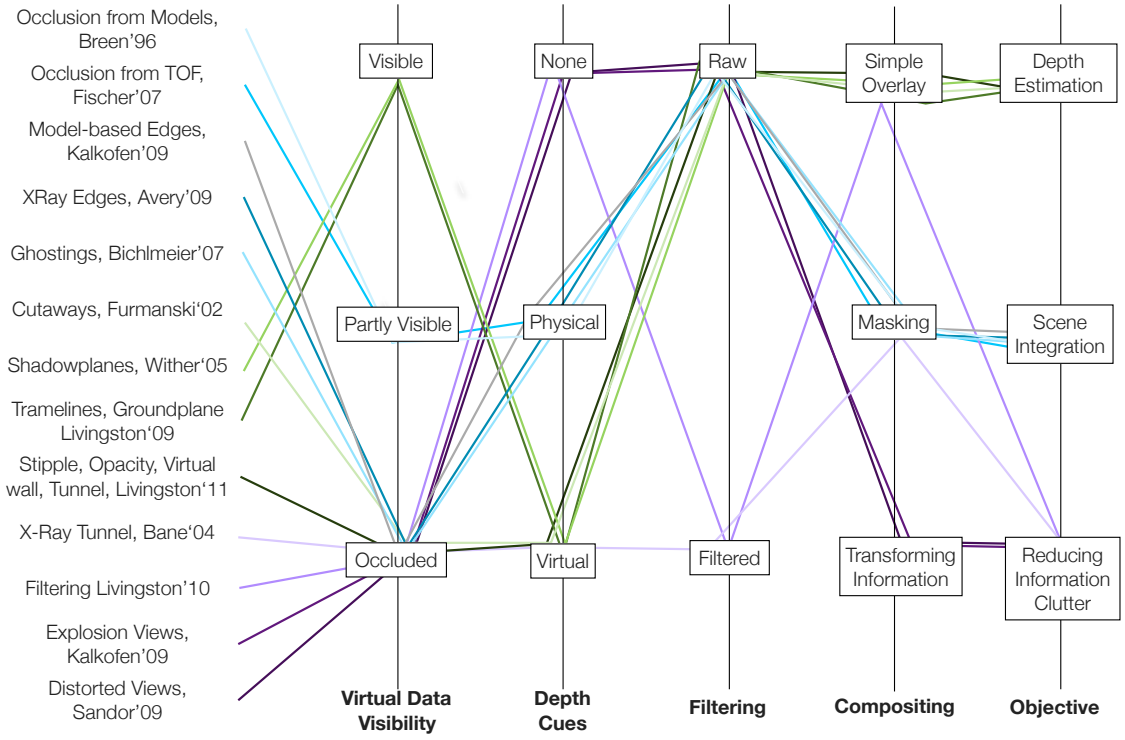


Figure 2.4: AR visualization techniques mapped to the taxonomy.

2.3 AR Visualization Pipelines

Based on the classification in the taxonomy, we can identify which visualization techniques are appropriate for which kind of visualization objective and which kind of visualized data. In this section, we use our dimensional space to redefine the classical visualization pipeline for AR. Our AR visualization pipeline provides a simplified representation of the visualization process. In order to be able to represent the complete classification of exiting techniques, we describe different implementations of the AR pipeline each addressing different visualization problems.

As mentioned in the introduction, for simple AR scenes that contain no occlusion and no complex data a simple compositing can be used that combines a defined registered virtual geometry with the video image based (Figure 1.3). Examples for this kind of visualization are the Magic Book where virtual content is overlaid over a book [42] or the see-through visualization within the Touring Machine [29].

The simple pipeline is not working for more complex situations with partially or completely hidden, or complex information. Therefore, several researcher groups developed methods that aim to increase the comprehensibility in these situations. From the last section we learned that there are three primary objectives for the visualization techniques:

- Achieving visual coherence
- Supporting depth estimation

- Reducing visual clutter

The taxonomy shows that these objectives are achieved by adding or removing different information. This difference requires that we adapt the visualization pipelines for the different needs (Figure 1.5).

2.3.1 Achieving Visual Coherence

Researchers in AR proposed various techniques that achieve visual coherence by extracting and using natural cues from the physical environment. Often occlusion or shadows were used to achieve a convincing scene integration between virtual content and physical world.

We are referring to these natural cues as physical cues, since they can also be found in the physical world. These cues try to mimic the natural behavior of physical objects for virtual objects. The main goal is to achieve visual coherence based on the assumption that our visual system knows how to process these cues.

The most common techniques used for AR are occlusion culling and ghostings. Occlusion culling basically makes sure that occluded virtual information is not visible. In contrast, ghostings try to make occluded information visible while preserving enough information from the occluding layer. Ghostings techniques in AR have their roots in illustration and illustrative renderings. They share the goal of uncovering hidden structures. Accordingly, a lot of ghosting techniques in AR are based on techniques from illustrative renderings and volume rendering.

We can subdivide the methods of embedding physical cues based on the data they are using. This choice depends on the availability of the contextual information about the physical world. Milgram was referring to the availability of the physical world representation as *Extent of World Knowledge* [91]. Ideally, the contextual model would be a complete model of the physical environment reflecting all its characteristics, such as geometry and appearance. The assumption of having such a complete model is valid for other visualization environments, such as Information Visualization or Volume Rendering, where all information that we see is described by a data representation.

Unfortunately, in AR such complete models from the environment, we are looking at, are not usually not available. Hence, it is important that AR visualization techniques adapt to the availability of the data. Data sources giving us information about the environment are the live video image and 3D models of the environment. The visualization techniques differ based on this availability, We differ between methods that use cues extracted from the live video and methods that use models representing the environment with different grades of detail. All these techniques share that they use a representation of the physical world to extract the cues that are physically existing in the environment.

Image-based Physical Cues Image-based techniques achieve visual coherence by extracting physical cues from video images. They are the first choice for creating physical cues in situations where the depth order of virtual and physical world is known (for instance through a semantic meaning as we have it for the visualization of subsurface infrastructure) and no accurate and precisely registered 3D model of the occluding physical world object is available. Since such an accurate model of the physical context may be

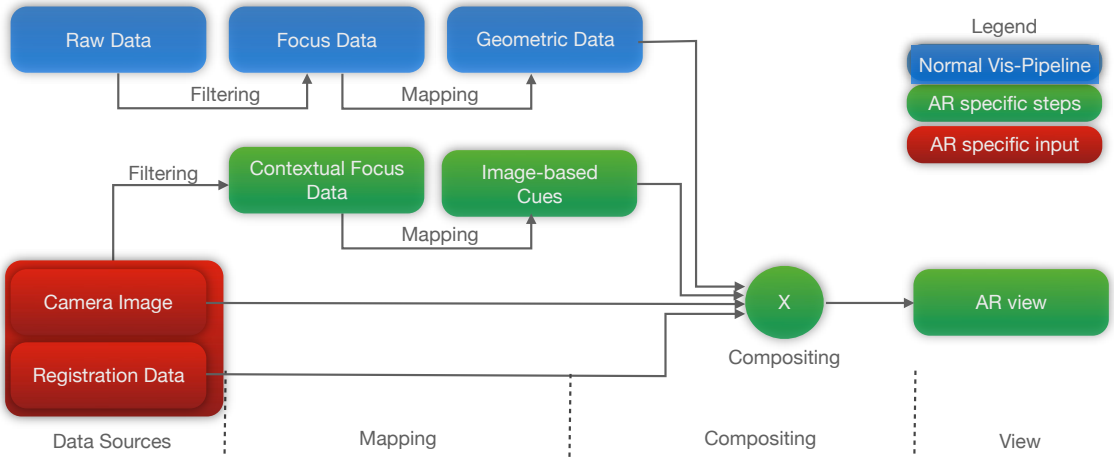


Figure 2.5: AR visualization pipeline for extracting image-based physical cues. (Data sources) Important elements from the camera are extracted and (Mapping) mapped to an importance map that represents the physical cues. (Compositing) These cues are then combined with the camera image and virtual geometries to create the final AR view.

Figure 2.6: Examples for using image-based physical cues. Left) Using edges for creating a ghosting in a medical application (Kalkofen et al. [64]). Middle) Using edges for visualizing remote views (Avery et al. [5]). Right) Using saliency for a ghosted view (Sandor et al. [105])

not available in every scenario, image-based techniques focus on creating physical cues based on 2D physical world data giving by the camera image. In Figure 2.5, we show the process of extracting physical cues from the camera image using the our adapted AR visualization pipeline. The AR pipeline reflects how important elements from the camera image are filtered and mapped to an importance map (Figure 2.5, (Left)). These cues are then combined with the camera image and virtual geometries to create the final AR visualization.

Such an approach has been introduced by Kalkofen et al. [64]. In their work, they extracted edges from a camera image and used them to create edge-based ghostings. In this approach, the edges are rendered on top of the video image and the virtual content (Figure 2.6, Left). The AR visualization pipeline in Figure 2.5 reflects this: 1) the camera image is used to extract edges as contextual focus data (filtering), 2) the edges are mapped to a ghosting mask that is 3) used in the final compositing step. Bichlmeier et al. extended

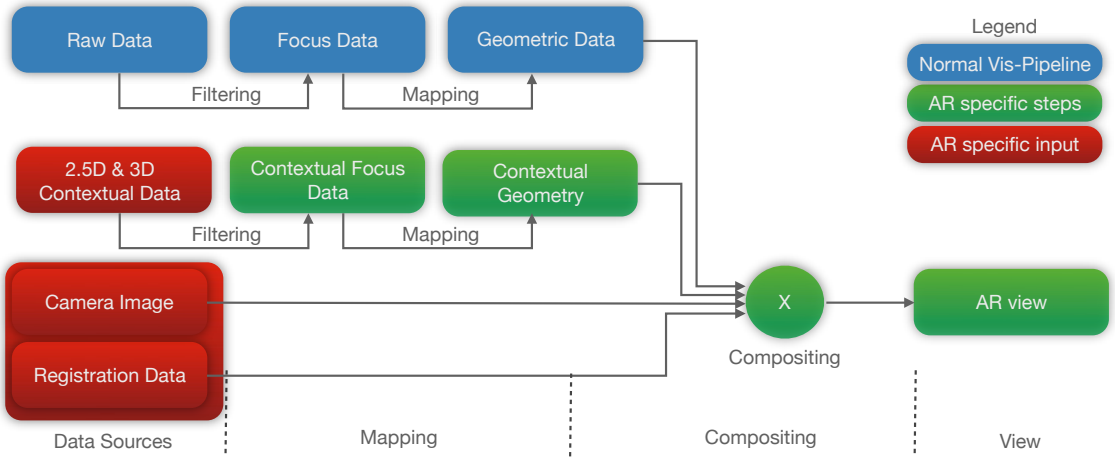


Figure 2.7: Pipeline for creating model-based physical cues. A 2.5D or 3D representation of the physical world is used as additional input for the visualization to create physical cues.

this approach by using a combination of edges and bright pixels as physical depth cues [12]. Another approach that uses edges as input to create physical cues is the method of Avery et al. [5]. They applied edges to improve their X-ray vision system in outdoor environments (Figure 2.6, Middle). Based on this work, Sandor et al. later on defined the physical cues as being saliency information. They computed saliency masks from the camera image and the layer of virtual content to decided which information should be preserved in the final rendering [105] (Figure 2.6, Right). All these methods work well in situation were enough meaningful data is available in the camera image, but will fail for poorly textured scenes.

Model-based Physical Cues If a model representation of the environment is available, this representation can be used to derive pictorial cues. In Figure 2.7, we depict our AR visualization pipeline using contextual data to derive cues for the scene integration. Contextual data as additional data source is only useful, if an accurate 3D registration and meaningful models are available.

Some previous approaches use contextual data directly for occlusion culling to provide occlusion cues. One of the earliest approach in this field used an interactive method to manually align models of physical world objects and applied these models for occlusion culling [15]. In the same paper, Breen et al. proposed to use stereo vision to create a 2.5 depth map for occlusion culling. A similar approach was applied by Fischer et al., who used a time-of-flight camera to create a depth map as input for occlusion culling (Figure 2.8, Left) [34].

More recent approaches use 3D models of the physical environment for increasing the visual coherence in X-Ray AR by deriving physical cues from the geometric or visual properties of the model. For instance, Lerotic et al. [81] presented an approach to maintain salient details of an occluder model from a pq-space-based non-photorealistic rendering. Bichelmeier et.al used ghostings from registered volumetric data to improve depth perception in AR applications in cases where hidden structure is of interest [13]. For this

Figure 2.8: Examples for using model-based physical cues. Left) Using time-of-flight camera data for occlusion culling (Fischer et al. [34]). Middle) A ghostings based on edges extracted from a registered 3D CAD model (Kalkofen et al. [64]). Right) Using importance maps applied on a 3D model (Mendez et al. [90])).

purpose, they used the curvature, the angle of incidence and the distance falloff to compute the final transparency in the ghosting. Kalkofen et al. demonstrated how to create ghostings based on an analysis of registered 3D CAD models [64] ((Figure 2.8, Middle)).

These last three model-based approaches for X-Ray visualization only work well if the models of the occluding object show interesting features in their geometry. Mendez and Schmalstieg presented an approach that allows to create comprehensible ghostings for rather simple shaped and sparsely textured physical objects [90]. By mapping a predefined importance map on the model of the occluding physical object, selected areas of the physical object can be preserved (Figure 2.8, Right).

However, existing model-based approaches focus nearly exclusively on indoor applications, since in this case it is easier to either build a model of the environment or capture it. For outdoor environments, it is more difficult to apply model-based approaches since the environment is changing dynamically and it is more complicated to model these complex environments accurately.

2.3.2 Supporting Depth Estimation

The previously described techniques primarily use occlusion cues to achieve a seamless integration of virtual content into the physical world. Nevertheless, these cues only provide ordinal depth information. In this section, we will discuss a group of visualization techniques that focus on supporting relative or absolute depth estimation. Depth estimation is complicated if the presented objects do not follow expected principles [23]. This is often the case in AR visualizations. For instance, it may happen when visualizing occluded objects, floating objects, or in general objects that are too abstract to hold normal physical characteristics. In order to support the depth estimation for these objects, additional cues are required. We refer to these cues as *virtual cues*, since they are not naturally available in the physical world. In the literature, they are also called *graphical aides* [83]. The AR visualization pipeline has to integrate these cues additionally.

We can mainly differentiate between three ways of adding virtual cues to the AR visualization:

- Using external geometries

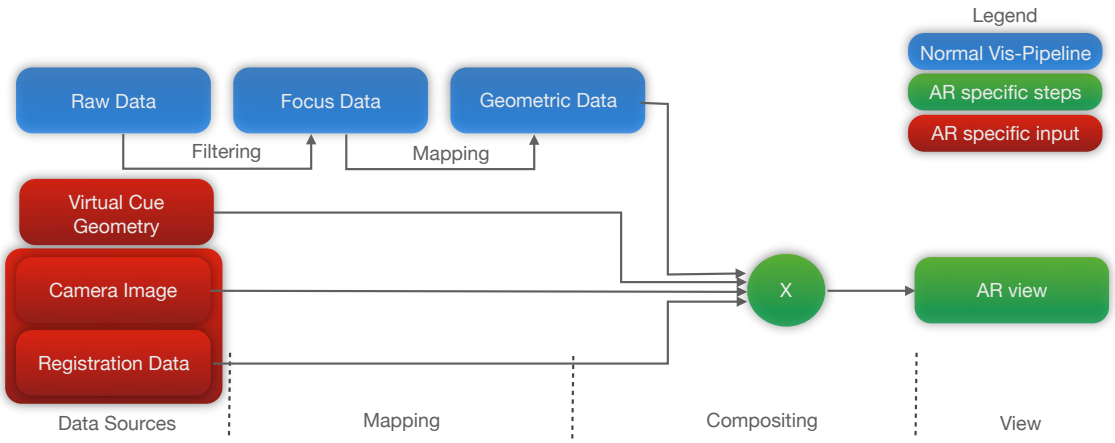


Figure 2.9: Pipeline for creating external virtual cues. An external geometry is added to the compositing step in order to create additional depth cues.

Figure 2.10: Examples for using external virtual cues. Left) Adding a set of parallel lines to improve the depth perception (Image courtesy of the U. S. Naval Research Laboratory, Livingston et al. [83]). Middle) Using a virtual ground grid to show absolute distances (Image courtesy of the U. S. Naval Research Laboratory, Livingston et al. [84]). Right) Virtual Shadow Planes for visualizing absolute depths (Wither et al. [131]).

- Mapping from virtual data
- Using cutaways

These techniques differ in the way the virtual cues are created, but they follow the same goal of supporting the user in estimating relative or absolute depth of virtual objects. Furthermore, we can differentiate them by their reference frame. Some techniques provide additional depth information about distances between the user's position and the virtual data. Others integrate additional hints about the spatial relationship between data and physical world. We divide them into *User-centric Depth Cues* and *Data-centric Depth Cues*, respectively.

External Aid Geometries Predefined virtual geometries, such as virtual ground planes or parallel lines, support the depth comprehension by providing additional depth cues (Figure 2.9). Usually, these additional cues are available in a predefined geometric representation. For instance, Livingston et al. included a set of parallel lines (called tram lines)

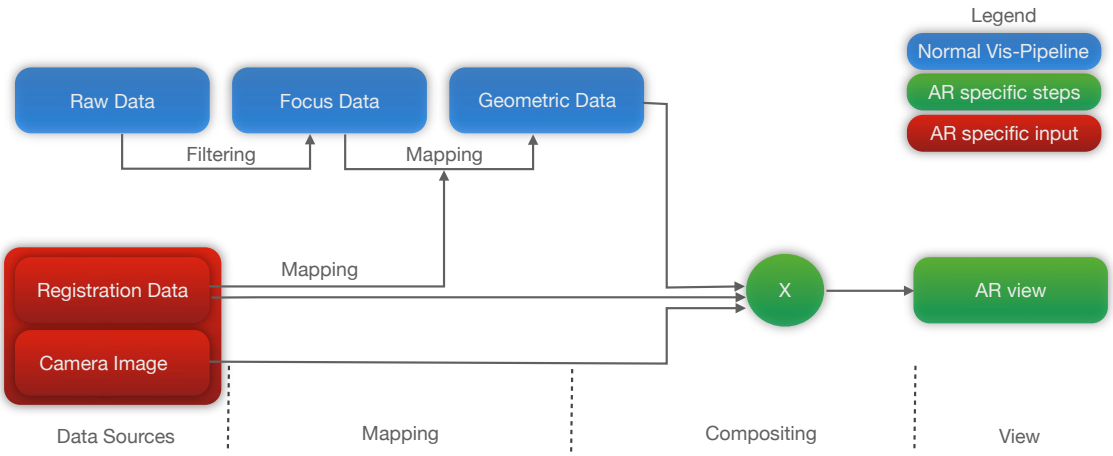


Figure 2.11: Mapping distance to appearance.

into their visualization of colored makers to improve the depth perception in an indoor and outdoor scenario [83]. Additionally, they added grid points to the tram lines (Figure 2.10, Left). The authors conducted a user study investigating this visualization technique and confirmed on a positive effect for depth estimation outdoors. It seemed that the users were tending to decrease overestimated depth judgments in outdoor environments. For indoor usage adding the tram lines was counterproductive, since it decreased the already underestimated depth.

Livingston et al. also introduced other examples of using external geometries to improve the depth perception in AR. For instance, they implemented a ground grid visualizing a virtual plane on the ground that either shows the distance to the user with concentric circles or with parallel lines [84]. Their graphical aid restore the visual cues of height in visual field, and relative size. The ground plane geometry can be extended by ties that show the connection between the virtual object of interest and the ground plane. This feature is in particular interesting for floating or subsurface objects, since it shows the connection between these objects and the ground. Wither et al. introduced a similar concept with the *Shadow Planes* [131]. The shadow planes consist of two orthogonal planes with depth measurements that are used to project shadows of virtual objects onto it (Figure 2.10, Right). The shadows in combination with a virtual distance scale on the planes was introduced to support the user in judging distances. Nevertheless, first study results showed no significant improvement using this technique.

Mapping distance to appearance Less obstructive, but also less direct are methods that encode the distance into the visual appearance. These methods form the second group of virtual cues. In Figure 2.11, we show an instance of the AR visualization pipeline that reflects this mapping. The pipeline adds the distance from the user to the virtual object to the mapping process and includes it into the visual appearance of the object. Thereby, the registration data helps to compute the distance between virtual object and user. Visual characteristics that are used to encode distance are transparency, color, frequency of stipples or density of virtual edges.

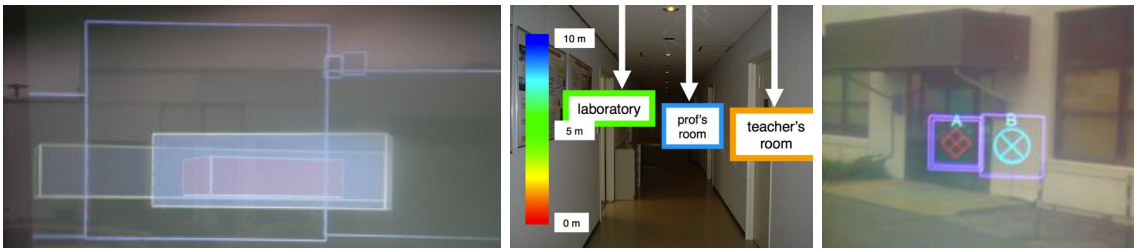


Figure 2.12: Methods that apply a mapping from distance to appearance. Left) Using intensity and opacity for encoding depth (image courtesy of the U. S. Naval Research Laboratory, Livingston et al. [85]). Middle) Using color to encode distance (Uratani et al. [123]). Right) The Tunnel Tool varies the amount of rectangles depending on the distance to simulate a virtual tunnel (image courtesy of the U. S. Naval Research Laboratory, Livingston et al. [84]).

This kind of mapping was discussed by Livingston et al. [85]. In their work, the authors suggested to change opacity and intensity of building renderings based on their distance. They compared this visual mapping to constant opacity and constant intensity and found a significant effect of using decreasing opacity on depth estimation (Figure 2.12. Left). Uratani et al. discussed how to map monocular depth cues to the appearance by using the distance of labels such as [123]:

- Depth of field by blurring the frame of the label depending on the distance.
- Relative size by changing the size of the label's frame.
- Aerial perspective by changing the saturation of the label as a function of distance.
- Texture gradient by including a texture pattern into the label.

In their final implementation they encoded the absolute distance into a color pattern (Figure 2.12, Middle). More recently, Livingston et al. used a set of mapping techniques to encode depth of virtual targets and compared them to each other [84]. Mappings that they used to encode the distance comprise:

- Stipples around the target, whereby the frequency increases with the distance.
- Opacity of the target that decreases with the distance.
- Synthetic edges around the target, whereby the distance is encoded in the spatial frequency of the edge pattern.
- Tunnel metaphor that uses squares around the target, whereby the number of squares depends on the number of occluding layers to the user (Figure 2.12, Right).

In a user study with professional users from the military, Livingston et al. compared the cues by asking the participants to map the virtual targets to five depth zones. The results of the study showed the Tunnel metaphor to be the most helpful cue.

While the mapping metaphors can provide absolute or relative depth measurements, they are usually not so well designed for maintaining visual coherence. In the next paragraph, we will show virtual cues aim support both.

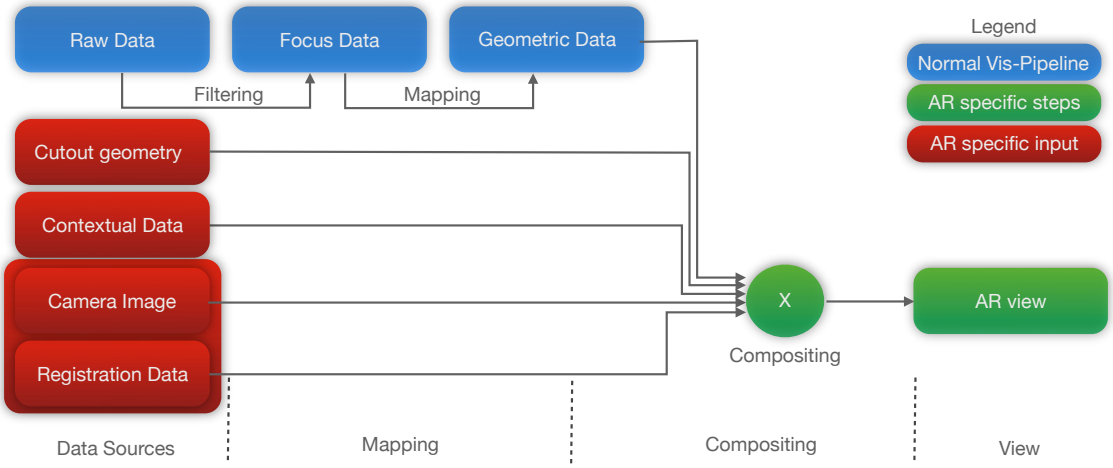


Figure 2.13: Creating additional virtual cues with cutaway geometries.

Figure 2.14: Cutaways as virtual cues in AR. Left) Virtual cutaways rendered on a wall to reveal a hidden green target object (Furmanski et al. [36]). Middle) Virtual cutaways used to visualize the interior of a car (Kalkofen et al. [62]). Right) A virtual excavation with a depth scale is used to visualize subsurface infrastructure in an urban civil engineering scenario (Mendez et al. [89]).

Cutaways Cutaways are visualization techniques that focus on supporting depth estimation for X-Ray visualizations. For this purpose, a part of the occluding object is cut out and reveals the hidden content. Cutaways are often considered as being a part of the group of Focus&Context techniques, since they allow one to inspect data in the cutaway area more in detail. But actually they can do more than filtering. They are also able to provide virtual depths cues, such as a box around the hidden object that shows measurements or perspective cues given by the shape of the cutout geometry. In contrast to using external geometries or mapping distance to appearance, the creation of cutaways requires extensive information about the physical world. Similar to the ghosting techniques, cutaways have their origin in illustrations and technical drawings, where the artist wants to reveal hidden parts of an object to the observer.

The input that is required to create a convincing cutaway in AR comprises a cutout geometry as well as a model of the occluding object. Since the occluding object is in this case the physical world, we need contextual data about the physical world (Figure 2.13). This data could be a rough surface model or a phantom geometry. By combining the

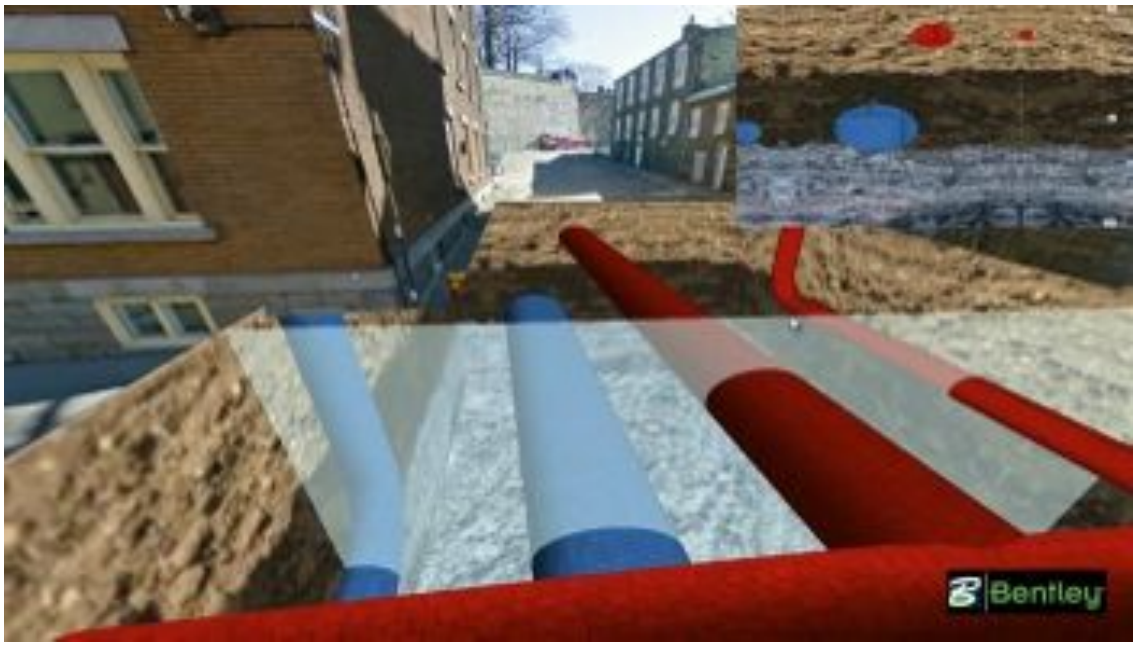


Figure 2.15: Vertical Slicing Tool. User can interactively control the slicing plane. The inset shows a 2D vertical section of the cut to support the comprehension of the spatial relationship between virtual elements (Courtesy of Stéphane Cote, Bentley)

cutout geometry and the phantom model, we can compute the correct cutout by aligning the cutout to the surface of the physical world object.

In their research from 2002, Furmanski et al. discussed general guidelines for designing X-Ray AR systems [36]. Among different suggestions for visual aids (ground planes grids, distance marker and temporal distance markers), they showed how to render virtual cutaways on a wall to reveal hidden information (Figure 2.14, Left). In a user study, they compared the visualization of a target inside a wall with and without cutaways. In contrast to the expectations, the study showed that the virtual cutaways do only help to understand the location of the virtual target for a dynamic video sequence, where the target was partially occluded by the frame of the cutaway box. But the authors stated that the findings from their study can be influenced by technical limitations of the AR system. This was confirmed by the participants reporting that their perception was influenced by the jitter from the registration.

Later on, Kalkofen used cutaways to visualize the interior of a miniature car (Figure 2.14, Middle). He used a phantom representation of the occluding object (the car) to compute the bending of the contour of the cut-out area. Based on this information, he was able to preserve the shape of the occluding object [62]. Further, the work of Kalkofen showed that the cutout is not enough to transfer the depth of a detached hidden objects. In this case, the visualization technique should provide additional hidden aids or geometries. Kalkofen, addressed this by rendering the cutout volume to add depth cues.

In the Vidente[1] project, Mendez et al. showed how to include such additional visual

[1] `www.vidente.at`

hints in a cutaway visualization (Figure 2.14, Right) [89]. They rendered a virtual excavation with depth measurements to visualize subsurface infrastructure in an urban civil engineering scenario. The virtual box allows the user to estimate the depth of the hidden objects. Furthermore, occlusions between the virtual pipes and the textured box allow to support the depth estimation, since it shows the exact spatial relationship between the cutout geometry and an object of interest.

The benefit of using a virtual excavation was extended with a virtual slicing tool in the work of researchers at Bentley[2]. Their slicing tool allows one to inspect the space between subsurface pipes more accurately by moving the slicing tool in the virtual excavation and showing a 2D vertical section of the cut in a separate view (Figure 2.15).

2.3.3 Reducing Visual Clutter

With the increasing amount of omnipresent information, the presentation of it is more likely to become subject to clutter. Consequently, researchers in the field of *Human Computer Interaction (HCI)* and Information Visualization investigate the issue of information clutter for a long time. In 2005, Rosenholtz et al. provided a definition of clutter in visualization systems:

> "Definition: Clutter is the state in which excess items, or their representation or organization, lead to a degradation of performance at some task." [103]

In the research field of Information Visualization several techniques have been developed that aim to reduce information clutter, such as filtering the amount of objects or view distortion techniques that allow to magnify or rearrange objects of interest.

In AR visualization, complex data is often embedded in complex physical environments that are crowded with information by nature. Thus, information clutter is a big issue in AR visualization. In order to approach this problem, researchers introduced methods that focus on the problem of information clutter in AR environments. Similar to the methods available for Information Visualization, research groups proposed methods that either reduce the amount of information by filtering the presented content or by using spatial distortion techniques to rearrange the objects in a more comprehensible way.

Filtering The main goal of information filtering is to reduce the complexity by decreasing the amount of displayed information based on a defined logic. In AR, location, user objectives and user-defined focus areas were used to control the filtering (Figure 2.16). One of the early research works that investigated filtering in AR is the work of Julier et al. [61]. They proposed a system for reducing information clutter in a mobile AR system by calculating a focus and nimbus area based on the user's location and objectives. Based on this information, they calculated the importance of virtual buildings and used it to decide whether a virtual building should be culled or not. A fading function provides smooth transitions between filtering levels to avoid that small changes in user's positions extremely change the displayed content. Later, Livingston et al. used a similar filtering approach based on focus and nimbus areas of objects of interests for removing clutter in military AR operations [84].

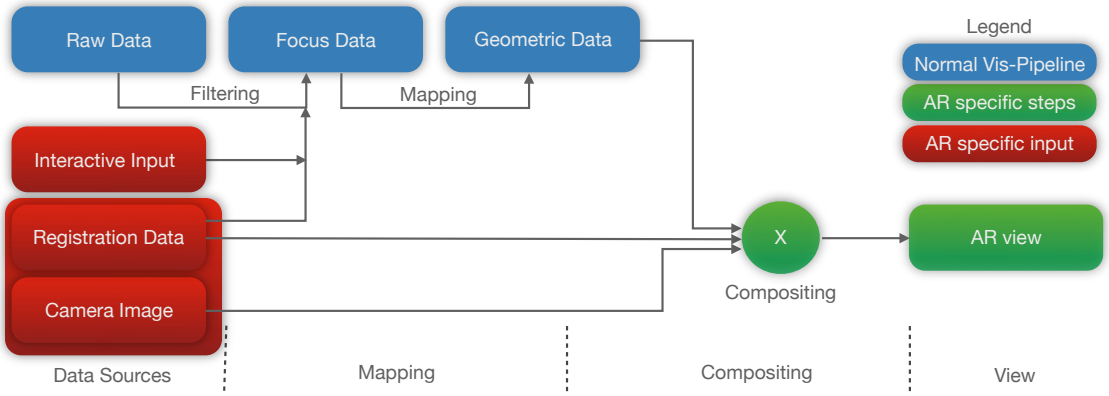


Figure 2.16: Information filtering pipeline.

Figure 2.17: Focus&Context techniques for information filtering in AR. Left) Magic lens that defines a focus area for displaying defined virtual content (Looser et al. [86]). Middle) Interactive X-Ray tunnel (Bane et al. [10]). Right) Focus&Context tools allow to explore occluded information in an X-Ray view (Kalkofen et al. [63]).

Instead of applying a global filtering, Focus&Context techniques in AR allow to filter virtual information based on an interactively defined spatial logic. For instance, Looser et al. introduced an interactive magic lens for defining a focus area [86]. Users of their system can control the lens with a physical marker. Based on this selection, only the area inside the lens displays virtual data (Figure 2.17, Left). This allows the user to inspect the virtual data while avoiding a cluttered context area. Additionally, the filtering criteria of their magic lens tool can be configured during run-time. Other interactive Focus&Context tool are the interactive X-Ray tunnel and the room-selector tool from Bane and Höllerer [10]. These tools allow defining a focus area that display virtual data, such as heat distribution of a building (Figure 2.17, Middle). Kalkofen et al. used Focus&Context filters in a scene graph to allow users to explore occluded information in an X-Ray view (Figure 2.17, Right) [63].

View Management The drawback of the filtering techniques is that they eliminate information to avoid information clutter. In contrast, view-management techniques do not completely remove this information. For instance, distortion methods reduce the space of non-relevant information in the visualization by distorting this. Such a distortion allows

[2] `http://communities.bentley.com/`

Figure 2.18: View management techniques for reducing clutter. Left) Rearranging labels (Bell et al. [11]). Middle and Right) View distortion based on a model of the environment (Sandor et al. [106]).

to keep the non-relevant information available for a fast overview [119]. View management techniques were developed in the context of information visualization, but were also already applied for AR visualizations. In AR, either the physical world information or the virtual information is transformed to create a more clean visualization. For instance, inspired by illustrative techniques Kalkofen et al. created explosion views to remove occluding areas from a hidden object of interest [65]. Their technique translates occluding parts to a new position to reveal occluded information. Recently, they extended these techniques by using compact explosion views to avoid that transformed content infers with the environment [121]. Sandor et al. used a distortion of occluding physical world objects to reveal occluded objects [106] (Figure 2.18, Middle and Right). While these methods manipulate the appearance of the physical world, approaches that focus on view management for label placement usually apply different methods for moving the labels in such a way that all information is readable. While the techniques of Bell et al. and Azuma et al. manipulate the label placement based on the label characteristics or the relationship between labels [7, 11], the techniques by Rosten et al. and Grasset et al. apply transformation to virtual annotations in order to rearrange them based on an analysis of the current environment [43, 104].

2.3.4 Summary

In this section, we discussed different groups of related work and described how they improve the comprehension in AR. We described different instances of our AR visualization pipeline that reflect the way how these visualization techniques combine information in order to address different perceptual issues in AR. Nevertheless, from the discussion we learned that a lot of problems are still not solved. In particular, for the visualization of data in nearly unknown outdoor environment, there are still several challenges.

So far, existing methods for seamless scene integration in AR often focus on indoor usage. According to this, several methods assume that a complete model of the physical world is available, which is complicated to achieve for outdoor environments. This is in particular a problem if the environment is dynamic or a large operation range is needed. Other methods assume that scenes contain a lot of edges that can be used for providing occlusion cues. Since, urban outdoor scenes often contain more important information,

Table 2.2: Overview of related work in comparison to our contribution.

Addressed Problem	Technique	Required Data	Benefits
Coherence	Model-based Physical Cues	Accurate 3D model required (difficult for outdoor environments)	Preserves geometric characteristics as depth cues
	Image-based techniques	Camera image, known depth order	Provide edges or single blobs
	Our contribution	Camera image, known depth order	Preserves important image structures as occlusion cues
	Our contribution	Camera image, GIS data	Extracts depth information as occlusion cues
Depth Estimation	External geometries	Predefined 3D geometries	Provide relative or absolute depth information
	Distance Mapping	Virtual data and user location	Provide relative depth information
	Cutaways	Predefined cutout geometry, context information	Provide relative or absolute depth
	Our contribution	GIS data	Provide relative or absolute depth
Visual Clutter	Information Filtering	User position or Interactive Input	Reduce clutter by reducing amount of information
	View management	Virtual data or camera image	Rearrange content to reduce clutter
	Our contribution	GIS data and 3D reconstruction data	Visualize multiple 3D datasets

we need methods that address these characteristics as well. Instead of being limited to accurately modeled environments and to edges and bright spots as as source for occlusion cues, in this thesis we show how to provide depth cues and a convincing scene integration in sparsely modeled environments and how to extract a combination of several important image features from the video images.

Visualization techniques that aim to support the depth perception in AR are also often either limited to indoor usage or require to create additional depth cues manually or interactively by the user for selected data only. For professional applications working on larger databases, methods that automatically create additional virtual cues for supporting depth estimation are needed. In this thesis, we will show how we can derive such additional cues from georeferenced data.

Finally, there is only little work that investigates the visualization of complex data in AR. The most of the related work focuses on information that has three dimensions. If we want to visualize information that has more dimensions, for instance visualizing construction site progress with a 4D representation, these methods struggle to enable the user to understand the relationship between these multiple data sets. In this work, we

Figure 2.19: Traditional methods for information query of digital assets. Left) Accessing information with a paper printout. Right) Accessing information with a mobile GIS interface.

will address this issue and propose visualization techniques that allow the visualization of complex data even if it has more than three dimensions.

2.4 Applications

Several industrial applications, and in particular applications from the ACE industries can benefit from the presentation of an integrated view of virtual data and physical world. By providing on-site feedback to the user and visualizing information in relationship to the physical world the mental workload can be decreased and thus outdoor working hours can be reduced. For instance, several research groups have shown different application areas for the on-site AR visualization of GIS data, such as for infrastructure maintenance [108], for agricultural data [72], or scientific data [95, 128].

In this section, we will describe several industrial outdoor applications that can benefit from an AR interface. It is important to note that in particular professional applications require a careful visualization design in order to be successful. Problems such as information clutter or wrong depth perceptions could produce wrong interpretations and can turn the advantages of the AR visualization into disadvantages.

2.4.1 Information Query for Digital Assets

For professional workers from the ACE industries in many tasks it is important to access information about assets in the field. For instance, they need information about infrastructure that is subject to maintenance or spatial information where to dig for assets and how to ensure safety during digging. Even if private workers want to create excavations on their private ground, they are supposed to query information about subsurface objects in the proximity to avoid damages on public infrastructure.

This kind of information is often presented on paper prints or on stationary computer

Figure 2.20: Traditional surveying methods. Left) A user surveys point of interest with the theodolite. Middle) He has to note down information about the surveyed point (such as type and id). Right) Sketching a set of surveyed points.

systems. Mobile GIS systems already extend the way of information presentation and allow one to access information in the field (Figure 2.19). Nevertheless, this way of information presentation does not provide the spatial relationship between the digital asset and the physical world. This relationship has to be built mentally by the users themselves, which requires a lot of experience and is often a source of mistakes.

In order to reduce the mental workload for professional workers, and to allow less experienced users to understand information about buried assets, a registered AR visualization can be beneficial. It supports the fast and accurate localization of subsurface objects and provides the spatial relationship to the physical world automatically.

2.4.2 As-built Surveying

As-built surveying is the process of capturing georeferenced information about newly built objects on-site. Companies use the surveyed information for as-built/as-planned comparison and for documentation purposes. In traditional surveying, the field worker captures reference points with a theodolite (Figure 2.20, Left) and has to move to each surveying point to place the device at its position for measuring. Only georeferenced measurements are stored on the device. Additionally, the worker draws the layout of the surveyed points in relationship to the construction site and between previously surveyed points (Figure 2.20, Middle and Right). Since the surveyed 3D points consist so far only of purely geometric information, they have to be combined with the meta information from the drawings in an additional step in the office. This is a lot of additional effort and requires expert knowledge. Furthermore, it produces a high mental workload. Using AR as an interface for surveying can avoid the need for this post-processing step in the office. The post-processing time could be reduced by directly surveying the georeferenced geometry on-site in relationship to the physical world with interactive tools integrated in an AR interface.

2.4.3 Planning Applications

Planning is an essential task for all construction projects. Planning allows for the inspection and calculation of the projected structures and helps stakeholders to understand the relationship between planned and existing structures. It helps to understand conflicts before the new structures are realized and prevent these conflicts. Traditionally, planning is done by using paper plans or desktop applications. The responsible construction site staff

Figure 2.21: Traditional paper plans vs. an AR interface for planning. Left) Paper-based planning, newly planned lamps are visualized in orange. Right) AR planning interface visualizing planned lamps.

takes the paper plans or printouts to the physical environment to inspect this information on-site. For integrating modifications, he has to go back to the office to apply the required chances.

AR as an interface for planning tasks allows one to inspect planned objects directly in relationship to existing structures. Interactive modifications of the as-planned objects can be applied directly on-site. There is no need to put additional effort to integrate modifications in the office. Furthermore, AR planning also allows the visualization of planned structures in a way that external parties can be integrated into the decisions process, since no special knowledge is required to understand the as-planned information (Figure 2.21, Right).

2.4.4 Construction Site Monitoring

Automated documentation and monitoring is an important topic for the construction industry since it improves the supervision of contractors achievements, as well as the detection of schedule derivations or the search for sources of defects and workers in charge. In particular, the last point is interesting for compensation requests and responsibilities. Adequate progress monitoring methods help supervisors and workers to document the current status of the construction work as well as to understand origins of defects.

Nowadays, construction site staff already uses digital photography to document the progress of a construction site. Typically, responsible staff members capture individual photos of the construction site on a regular basis and store them together with the construction site plans in a database (Figure 2.22). This enables the supervisors to relate possible errors or bottlenecks to certain dates. The disadvantage of this approach is that a staff member has to take the photographs manually, being time-consuming and leading to areas which are not covered very well. Another aspect is that the relation between acquired photographs, nor the relationship to the physical construction site is available. This creates a high mental workload for the supervisor when she has to map the photographs

Figure 2.22: Construction site monitoring using camera images. The image sequence represents different steps of the construction progress.

to the actual construction site.

AR in combination with adequate visualization techniques can support the on-site construction site monitoring by providing the required information in direct relationship to the physical construction site. AR visualizations have already been applied for displaying construction plans on-site [132]. Furthermore, Golparvar-Fard et al. discussed AR visualization for supervising the progress on construction sites within the scope of the 4DAR project [39, 40]. Their system computes a single value representing the status of the construction site, such as the current level of completion. The corresponding value is visualized by using a color coding of the 3D real world object. While this approach allows one to study multiple differences between planned data and the current real world situation, it does not allow for detailed analysis of the data.

2.4.5 Flight Management and Navigation of Aerial Vehicles

Micro aerial vehicles (MAVs) such as quad- or octocopters are an emerging technology. While small commodity devices are designed to perform simple movements in the near-field controlled by a simple remote control, professional devices such as octocopters equipped with automatic balancing technology, professional GPS and inertial sensors are focusing on mid- and far-distance applications. These devices are built to even transport minor additional payload such as a camera. There are several professional applications that can benefit from professional unmanned aerial vehicles, such as collecting a set of aerial views for reconstructing an area of interest. These 3D data sets can be used for industrial applications such as for construction site monitoring.

For these applications it is important to obtain a high reconstruction quality within a limited flight time. Although, automatic flight path planning methods can compute a flight path including a high number of images from meaningful viewpoints [54], they usually only plan ideal viewpoints and send them as a list of waypoints to the *Micro Aerial Vehicle (MAV)*. They do not consider how the MAV is exactly moving from one waypoint to the next. In order to address this problem, a lot of research on autonomous flying has been done [35]. However, methods for autonomous flying are still a field of research and have so far not the ability to fully replace the human in the loop supervising a flight session on-site avoiding collisions with physical obstacles.

Professional MAVs come with a remote control that allows the supervisor to interfere in an emergency. 2D map interfaces allow one to inspect the complete flight path (Figure 2.23, Middle and Right). However, the user still has to establish the spatial relationship

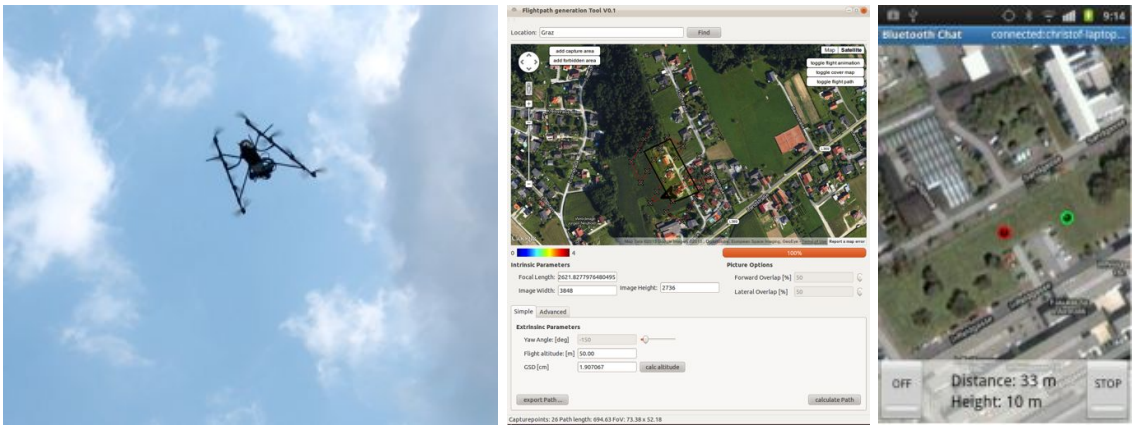


Figure 2.23: Map-based navigation interfaces for MAVs. Left) Desktop application shows flight path in red with white crosses. Right) Mobile applications can be used on-site by the user, but still requires him to map the 2D locations to the physical environment. The current position of the vehicle is marked with green, the next waypoint visualized in red.

between the positions on the 2D map and his physical environment. With such a workflow it can be hard to avoid obstacles since the user has to either mentally map the physical obstacles to the 2D map or to transfer the flight path from the map to the physical environment. Furthermore, it is hard to understand the distance of the MAV, if it is to far away and depth cues are not available (Figure 2.23, Left). Using Augmented Reality (AR) as an interface for supporting the navigation of aerial vehicles has the advantage that this relationship is provided automatically by overlaying the waypoints onto a camera image in real-time. Obstacles on the path are visible in the camera image and conflicts can even be highlighted, if depth information is available.

This benefit of AR was already exploited by Kasahara et al. in the exTouch project where a robot was navigated by physically moving an AR interface in relation to the robot [67]. This approach was demonstrated to work well in near range to the user. Nevertheless, when it comes to mid- and far-range navigation, this approach can not be used in this way, since 1) the MAV's positions are often outside the reaching range of the user, and 2) the depth estimation for distanced floating objects is more difficult. For this purpose, it is important to develop AR visualization techniques that address these problems.

Chapter 3

Methods and Systems

Contents

Before we can start to investigate visualization techniques in outdoor *Augmented Reality (AR)* environments, we need a system that allows displaying registered data and provides an easy integration of new visualization techniques. This chapter describes how we achieve this by 1) discussing the registration methods we apply, 2) discussing the methods to access the data that is relevant for the aforementioned applications as well as the data that provides context information about the environment, and 3) how we integrate all these methods into one system to finally display the data.

3.1 Registration

The first prerequisite for visualizing data in AR is the registration technology. The registration assures that virtual objects are correctly aligned to the physical world. There are different possibilities to achieve a correct registration, varying from simple maker-based registration techniques [68] to tracking methods that use natural-features [127], sensor fusion [111] to localization-based approaches [4, 125]. All these technologies use different approaches to achieve the same goal; aligning the virtual data in relationship to the physical world for a coherent AR visualization.

Marker-based and natural-feature-target based techniques are often used for indoor AR applications. For the outdoor applications, they are usually no option due to larger working environments and environmental influences. To achieve a reliable registration in unknown outdoor environments, we integrate more sophisticated sensors into the AR setup and apply a sensor-fusion-based approach combining panoramic mapping and tracking with an IMU and a GPS receiver. For application where a 3D model of the physical environment

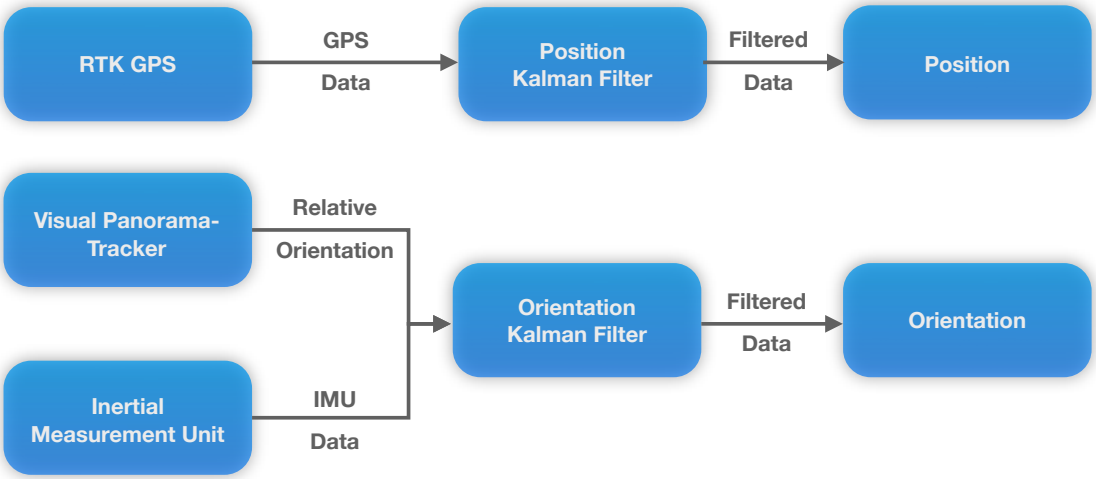


Figure 3.1: Multi-sensor fusion system architecture. Data provided by the GPS sensor is filtered by a *Position Kalman Filter*. The *Orientation Kalman Filter* fuses data from *IMU* and *Visual Panorama Tracker*.

is available, we apply localization-based methods that relay on a server-client structure in combination with panorama-based or model-based tracking.

3.1.1 Multi-Sensor Outdoor Registration

For AR applications that focus on entertainment and tourism in outdoor environments, researchers and companies typically use a combination of built-in sensors of mobile phones or tablet computers. For professional applications, the accuracy that these sensors provide is not sufficient. They often show positioning inaccuracies in the range of several meters, in addition orientation sensors are often a subject to drift [109]. In order to provide accurate overlays of the virtual data, we implement a registration method that is able to achieve registration accuracy in the centimeter and subangle range.

To achieve such a highly accurate position and orientation estimate of the AR system in outdoor environments, we combine the measurements of different sensors:

- L1/L2 *Real-time Kinematics (RTK)* GPS
- Inertial Measurement Unit (IMU)
- Vision-based Panoramic Tracker

In outdoor AR applications, GPS is often used for positioning measurements. Nevertheless, even professional devices are only able to deliver sufficient accuracy under perfect conditions, such as in unoccluded areas with a high number of visible satellites. In order get accurate measurements and to compensate noise, we use a professional GPS receiver in combination with a Kalman filter.

For achieving a high GPS localization accuracy, the GPS receiver performs dual frequency measurements and applies RTK for accurate positioning. We use correction data

Figure 3.2: Panorama generated by panorama tracker.

from one of our industrial partners (WienEnergie AG) for differential corrections. With an update rate of 1 FPS, a reference station delivers the correction signal to the device in a format following the RTCM standard[1]. A requirement for receiving the correction signal is a network connection as well as a correct configuration of the correction signal. For this purpose, we use an Open Source NTRIP application[2].

We apply a Kalman filter for the positioning as shown in Figure 3.1 to compensate the position measurements of the GPS receiver. This filter produces smooth movements and can be adjusted to the amount of smoothness of the expected movement [111].

For estimating their orientation, AR systems often use inertial sensors with gyroscopes, magnetometers and accelerometers. Unfortunately, these sensors are subject to drift and also sensitive to environmental influences, such as electromagnetic interference that often occur in urban environments. We combine the orientation measurements from an *Inertial Measurement Unit (IMU)* with relative measurements provided by a vision-based panorama-tracker [111] to avoid these problems.

The panorama tracker is based on feature detection and matching. During a mapping step, it creates a panoramic representation of the environment that is stored for further tracking. As the panoramic map is based on a cylindrical representation of the environment, the panorama tracker assumes only rotational movements. Similar to *Simultaneous Localization and Mapping (SLAM)* approaches [25], the idea is to 1) determine the pose of a new frame relatively to already mapped data, and 2) add features from this newly localized frame to the existing map that is then again used for further pose estimation [71, 100]. In order to determine the pose of a new image relatively to the panoramic map, we extract feature from the camera image and match them against the map.

By combining the measurements of IMU and panorama tracker in an Orientation Kalman filter as shown in Figure 3.1, we are able to achieve robust absolute orientation measurements that are mainly drift-free [109]. Other inaccuracies can result from the magnetic deviation, a value that describes how much the absolute orientation measurement differs from the geographic north. This measurement depends on the current location and has to be configured manually or can be computed from the current GPS location.

[1] `http://www.rtcm.org`
[2] GNSS Surfer `http://igs.bkg.bund.de/ntrip/download`

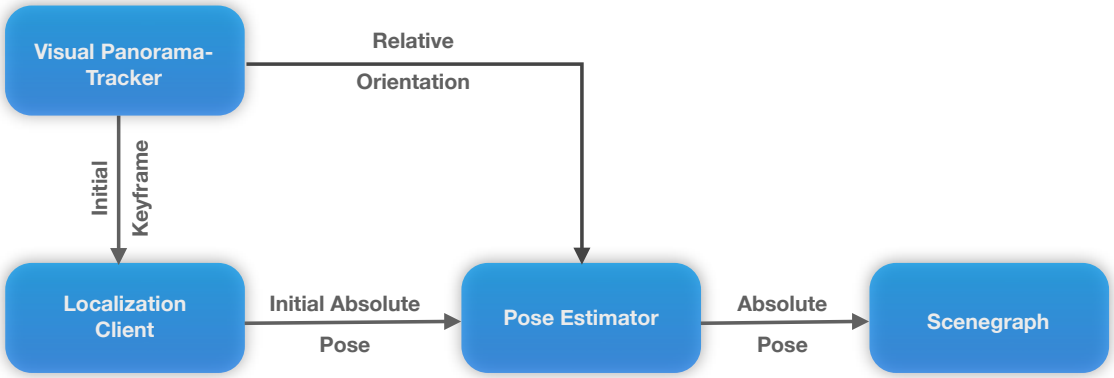


Figure 3.3: Model-based localization and panoramic tracking. The panorama tracker updates the pose estimation with relative orientation measurements. For an determining the absolute pose, an initial keyframe is send to the localization client. This is done, each time the panorama tracker can not relocalize itself.

3.1.2 Model-based localization

If a 3D model of the environment is available (Figure 3.4, Left), an accurate localization can be achieved by using the model data as input for localization. We can create such accurate 3D models by 3D reconstructions using a set of camera images. Thereby, it is important that these images cover the area of interest to allow for a robust localization. Additionally, we can extend the area by integrating new images into the 3D reconstruction.

In order to integrate a model-based localization into our system, we implement a talker-listener structure based on the Robot Operating System (ROS[3]). On the AR client-site, either a visual panorama tracker or a model-based 6DOF tracker receives absolute pose measurements and use camera images as tracking input.

Panoramic tracker The panoramic tracker uses the incoming camera images to calculate its orientation relative to an initial keyframe. By sending this initial keyframe to the localization client, an initial absolute pose can be computed in relationship to the georeferenced 3D model. The localization client sends the localization information back to the panoramic tracker. The tracker uses this absolute information to update its relative orientation measurements.

The accurate registration in relationship to the georeferenced model allows for rendering accurate AR visualizations (Figure 3.4, Right) as long as the user performs purely orientational movements. If the user moves to a new position, the motion model of the panoramic tracker that assumes orientational movements, loses the tracking. Accordingly, the tracker starts to create a new panoramic representation of the environment with a new initial keyframe. Therefore, we have to perform a relocalization and send the new initial keyframe again to the localization client (Figure 3.3). The process of sending of the image data as well as receiving the localization information is implemented in an extra thread.

[3] `http://www.ros.org`

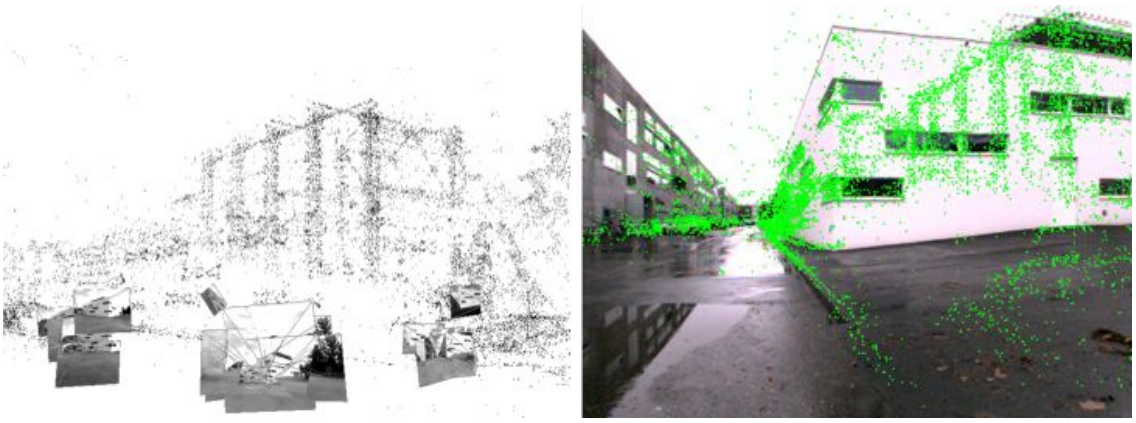


Figure 3.4: Model-based localization. Left) 3D point cloud with localized camera frames. Right) Localized camera frame overlaid with registered point cloud.

Model-based tracker The model-based tracker is based on the talker-listener concept of ROS as well. The AR client publishes an initial camera frame and waits for answers that contain a localization matrix of this frame in relationship to the georeferenced point cloud. Based on this image data, the localization matrix is calculated by the remote localization client that is connected to a *Structure from Motion (SfM)* database. After receiving the localization matrix, the AR client can initialize a model for model-based tracking. The model is initialized with the initial pose, the localization image frame, and the 3D points of the environment (given by the SfM application). For this purpose, we create a *Point Feature* for each 3D point that is visible in the localization frame. A Point Feature contains the 2D image data, the 2D location in the image frame and 3D information about this point. Based on this initialized model, movements relative to the initial localization matrix can be calculated as soon as new camera images come in.

We compute correspondences between the incoming camera image and the model and use this correspondence information to compute a relative transformation between them. This information is then used to compute an absolute transformation matrix. As long the model-based tracker finds enough correspondences between the model and new incoming camera frames, the initialized model can be used.

However, if the AR client moves to far away from the initial pose, there are not enough correspondences available to compute the transformation. Thus, we have to compute a new model. The process starts again by publishing the camera frame and waiting for localization answers (Figure 3.5). As long as the model-based tracker computes accurate registration values in relationship to the georeferenced model, the AR client can create accurate georeferenced AR overlays (Figure 3.5, Left).

3.2 Data Sources

The main goal of this thesis is to develop visualization techniques for industrial application and professional data. Therefore, we spend some effort in accessing and processing expert data, such as data from *Geographic Information Systems (GIS)*, *Building Information*

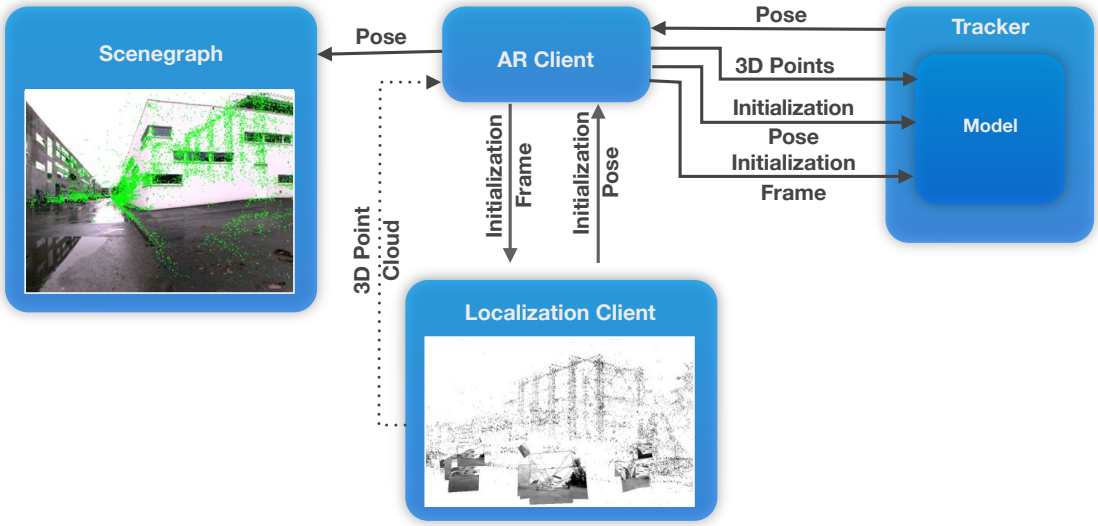


Figure 3.5: Model-based tracking. A localized camera frame and the 3D point cloud data is used to initialize a model for tracking. The model is then used for tracking and allows for an accurate registration.

Modeling (BIM) and 4D data from aerial reconstruction. In this section, we will describe the methods used to acess this data.

3.2.1 Geographic Information Systems

Geographic information systems (GIS) have a long tradition in supporting architecture, engineering and construction industries (ACE) in managing existing or future infrastructure. In civil engineering and construction industries, the fast access to inventory is essential. Architectural applications are mainly supported by managing as-planned data. Companies from the ACE sector have usually extensive GIS databases of their infrastructure comprising pipes, cables and other installations, as well as objects in the surroundings, including street furniture, trees, walls and buildings. But also open source tools and user generated map data, such as OpenStreetMap[4], provide access to detailed 2D information on building outlines, street curbs and other features in the environment. Efficient utility location tools and computer-assisted management practices can largely reduce costs and hence are subject to continuous improvements.

Recent developments brought GIS tools to mobile devices for on-site inspection (e.g., ARCGIS for Android[5]). However, current visualization techniques implemented in these tools do not show the relation of GIS data to the physical world context. Accordingly, they still involve the tedious task of referencing assets correctly to the physical world.

Using Augmented Reality (AR) as an interface to extend mobile GIS systems has the potential to provide significant advances for the field of civil engineering by supporting the

[4] `http://www.openstreetmap.org`
[5] `http://www.arcgis.com`

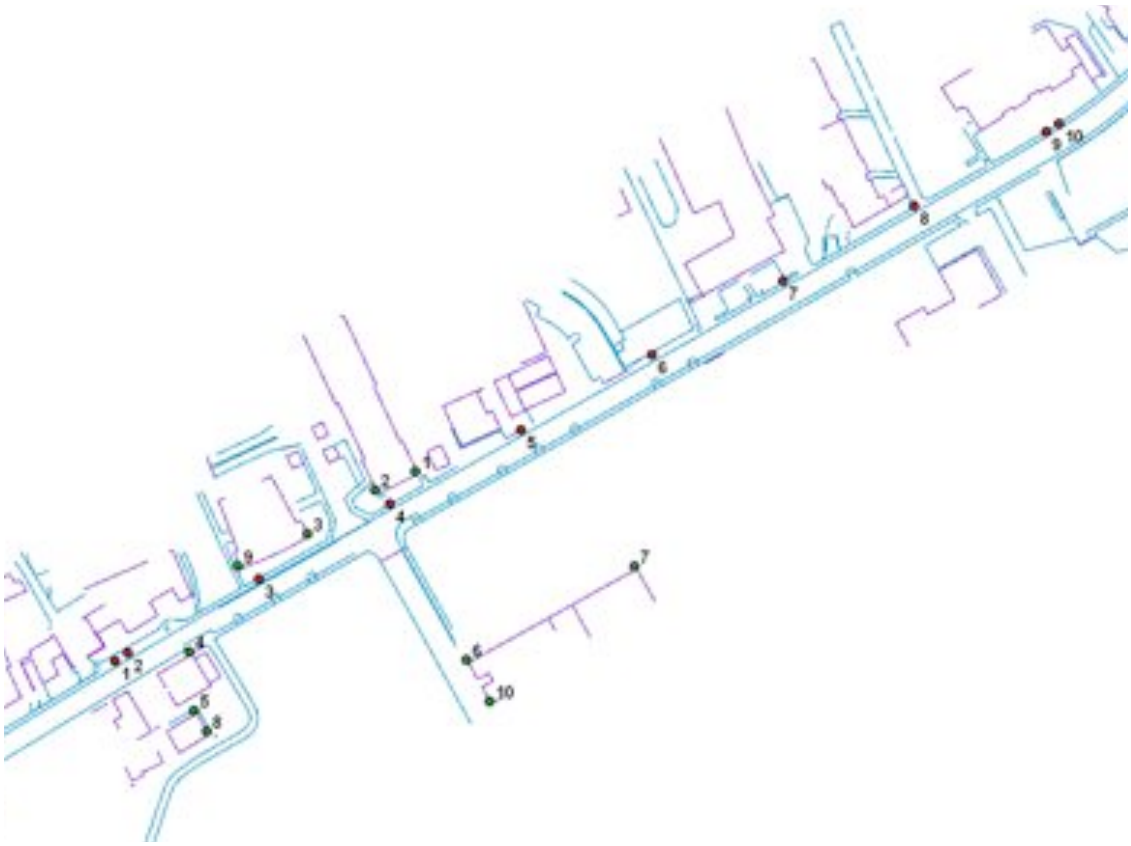


Figure 3.6: GIS information of a street. Purple represents building outlines. Blue shows curbstones.

visual integration of existing assets into the psychical world. The visualization of both, physical and virtual geospatial information, at the same time in reference to each other has a big potential to avoid errors and to decrease workload. Since information in GIS is usually stored as a 2D or 2.5D representation, an additional conversion step to create 3D information has to be performed, before the data can be displayed in an AR visualization.

Firstly, we extract the data from the GIS database. For this purpose, we use FME[6], an integrated collection of tools for spatial data transformation and data translation. FME is a GIS utility that helps users converting data between various data formats, as well as process data geometry and attributes. The user interactively selects objects of interest in the back-end GIS (Figure 3.6). These objects can then be exported to an external file by the FME software. For the file format, we decided to use *Geography Markup Language (GML)*, since GML is already widely used in order to store and exchange geographic objects and provides flexibility for extensions.

The exported GML-based file represents a collection of features. In this collection, each feature describes one physical world object. Since some geometric data is only available in 2D, this data has to be converted to 3D representations. For the conversion, we use a

[6]The Feature Manipulation Engine: `http://www.safe.com`.

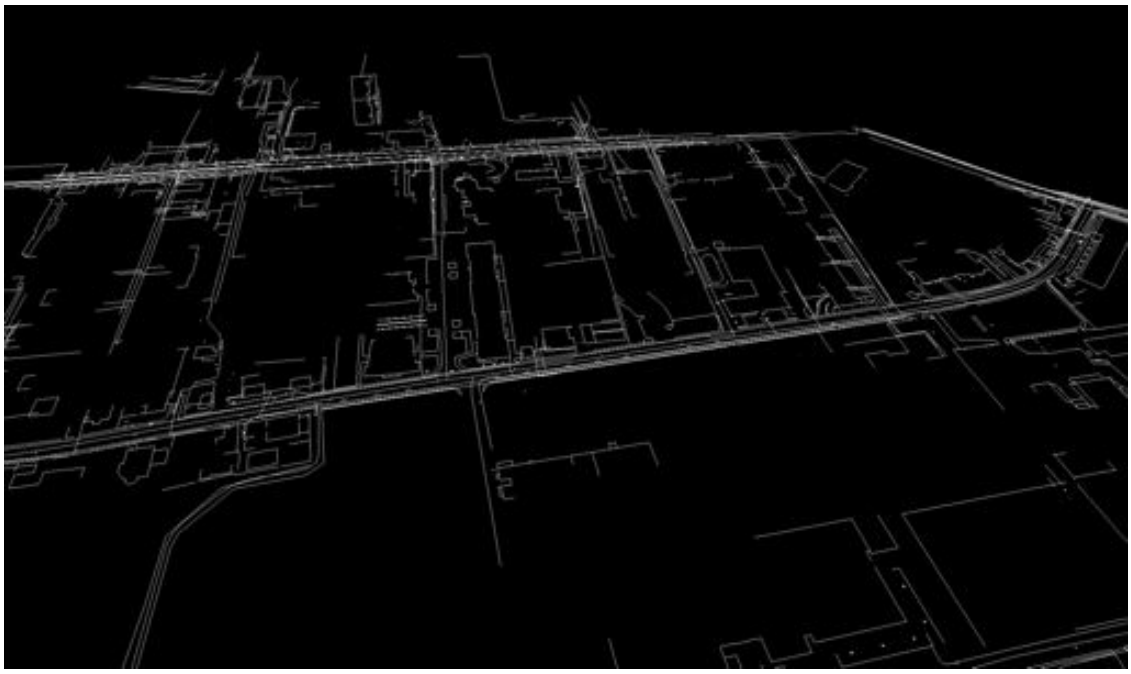


Figure 3.7: Semantic scenegraph representation. Each VidenteGML feature is converted into a 3D scenegraph object. The scenegraph object includes the geometric properties, but maintain the semantic information as well.

digital elevation model (DEM) and known laying depths of subsurface objects. From this information, we compute 3D information for each 2D feature.

Exporting GIS data to a GML file is a step that has to be performed offline, before starting the AR system, as it requires external software and interactive selection of the export area. We restrict the number of supported features by using own GML-based format (VidenteGML). In order to be flexible for extensions, the design is based on the following concepts:

- Each GML document is a collection of features (Listing 3.1).
- One feature includes all geometric and semantic information that is required to describe a single physical object.
- Attribute information is modeled as an XML property tag.
- Geometric information is modeled as an XML property tag.
- Meta information is represented as an XML tag attribute.
- A feature is composed of an arbitrary number of XML property tags describing semantic and geometric properties.
- Property tags are self-descriptive and encoded similar to feature tags.

- Geometry property tags support geometry types according to the GML format (supported types are gml:Point, gml:LineString and gml:Polygon for encoding common geometries and gml:RectfiedGrid for transferring DEM information)

Listing 3.1: Feature in VidenteGML format.

```
<Feature id="13168176" group="water" source="Salzburg␣AG">
    <property name="status" type="string">in Betrieb</property>
    <geometry name="as_built_position" alias="B␣Position">
            <gml:Point srsName="EPSG:31258" srsDimension="3">
                <gml:pos>-21215.433 296985.947 423.516</gml:pos>
            </gml:Point>
    </geometry>
</Feature>
```

VidenteGML supports the description of a wide variety of objects without requiring additional definitions for new types. This is a big advantage, since geospatial databases often consists of a very high number of different feature classes from different sources (such as electricity, gas, water or heating). For instance, for our field trials, we worked with datasets from three different companies. It would be a lot of effort to describe all existing feature classes separately. Furthermore, the VidenteGML format supports modifications of the data and a data round-trip by flagging changes, additions or deletions of features in an attribute. Such changes can later be parsed in the FME tool and can be used to update the geospatial database.

In order to render VidenteGML data, the data has to be converted into a scene graph format. For each feature, we create a scene graph object representing the semantic attributes and geometric properties of the feature (Listing 3.2). A local reference point, helps to transform the data in a local coordinate system, in order to avoid double precision problems in the rendering. We support the main standard features of GML such as GMLLineStrings, GMLLinearRings, GMLPoint and GMLPolygon in the conversion step. In our current implementation, we use COIN3D[7] to implement the scene graph object, as it is easily extendable. An example for the scenegraph representation of the GIS data from Figure 3.6 is shown in Figure 3.7.

Listing 3.2: Scenegraph format.

```
DEF ID_ SoFeature {
    fields [ SFString attribute_id, SFString attribute_name,
    SFString attribute_alias, SFString attribute_group,
    SFString attribute_groupAlias, MFInt32 attribute_level,
    SFString attribute_mod, SFString attribute_source]
    attribute_id ""
    attribute_name "enclosure"
    attribute_alias "NAT␣Gebaeudewand"
    attribute_group "topo"
    attribute_groupAlias "Basisdaten"
    attribute_mod "none"
    attribute_source "Grazer␣Stadtwerke"
```

[7] http://www.coin3d.org

```
    GmlLineString {
        fields [ SFNode vertexProperty, SFInt32 startIndex,
            MFInt32 numVertices, SFString attribute_name,
            SFString attribute_alias, SFInt32 attribute_lod,
            SFString attribute_mod, SFInt32 attribute_srsDimension,
        SFString attribute_srsName ]
            vertexProperty
            VertexProperty {
                vertex [ 442.44901 665.43701 7.2090001,
                443.37399 663.38098 7.6960001,
                443.48199 663.14001 7.7729998,
                444.50101 660.87701 7.7470002 ]
        }
        startIndex 0
        numVertices 4
        attribute_name "position"
        attribute_alias "Position"
        attribute_lod 1
        attribute_srsDimension 3
        attribute_srsName "EPSG:31258"

    }
}
```

3.2.2 Building Information Modeling

Building Information Modeling (BIM) describe systems that combine various kinds of information about the life cycle of a building, such as construction plans, but also plans for building management. Information from BIM can provide as well geometric as semantic information about the physical world object, similar to information from GIS. Accordingly, BIM can be a helpful source of information for AR visualizations. They contain geometric as well as semantic information. Furthermore, the general concept of BIM supports 3D as well as 4D representations for construction sites.

Nevertheless, a lot of companies still work with 2D CAD plans. In this case, we have to apply a data conversion step in order to create 3D data for visualization, similar as for the GIS data. Often BIM or CAD data is not even georeferenced. This means, for outdoor usage we have to register the data according to the physical world before we can display it using an AR system. For this purpose, we select at least three 3D point correspondences in the unreferenced model and in an georeferenced model. Based on these correspondences, we apply the Absolute Orientation algorithm to compute the transformation between both [55].

3.2.3 Aerial Vision

In contrast to GIS and BIM that either present an as-planned or a surveyed as-built status, aerial vision allows capturing the actual as-built status of a defined area. Nevertheless, due to the high costs and the high effort, manned aerial vision was traditionally only used to capture large scale areas such as Digital Surface Models (DSM) of complete cities. With the recent developments in the sector of micro aerial vehicles (MAVs), aerial vision

Figure 3.8: Aerial 3D reconstruction of a building.

also became interesting for small scale reconstruction and can be applied for construction site monitoring to capture the as-built status. Autonomously flying MAVs are equipped with standard consumer digital cameras and capture images automatically of the area of interest [22].

Thereby, one flight session usually results in sets of 200-300 high-resolution and highly overlapping images. This set of images is used as input to a SfM approach [58]. The SfM approach computes the scene geometry as a sparse point cloud. Since sparse geometries contain only limited data for visualization, we apply state-of-the-art methods for model densification. We apply the approach of Furukawa et al. to calculate an oriented semi-dense point cloud [37] (Figure 3.8).

For some applications, mesh data is needed. In order to obtain a 3D mesh, we use *Poisson surface reconstruction* [69] to create a mesh from the point cloud data. Additionally, we include available GPS information from the MAV into the reconstruction workflow to reduce computation time and to obtain a georeferenced 3D model at a metric scale.

Using this workflow we are not only able to create as-built data sets, but also to create

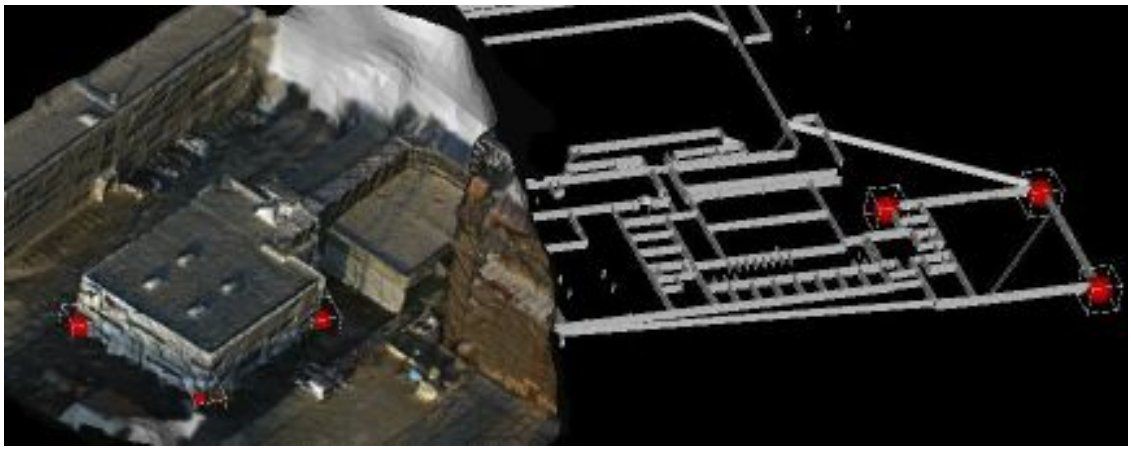


Figure 3.9: Computing the absolute orientation. Left) Selecting 3D points in the 3D mesh (red spheres). Right) Selecting 3D points in the CAD data (red spheres).

4D or so called time-oriented data sets that represent the as-built status over time. To obtain such 4D data it is important that the meshes are accurately aligned to each other. We perform accurate registration in a multi-step approach. Having initial georeference information available allows for a coarse registration of the individual models over time. In a second step, we apply a matching procedure on accumulated feature descriptors available for the sparse 3D points (resulting from the SfM).

3.2.4 Interactive Geometry Abstraction

3D meshes or point clouds are often very complex, which decrease the ability of users to understand them. Interactive methods for geometry abstraction allow one to create a more simplified representation of the complex data. For instance, a building wall represented by 3D point cloud data can be converted into an abstract rectangular wall representation.

Such abstract descriptions are already available from different data sources, such as GIS or BIM databases. GIS databases, for example, store outlines of buildings and even time-dependent representations are available for modern construction sites from BIM systems. However, such data is not always available, often is simply not accurate enough, or it does not represent the time component. In these cases, we need an additional method to create this kind of data. The main goal of this subsection is to describe a semi-automatic tool for creating abstract representations of 3D models.

For this purpose, we implemented an editor that integrates a set of semi-automatic tools for creating abstract 3D models from point clouds and a 2.5D plan. Complete manual modeling of this kind of data would require a high effort, especially when it comes to multiple 3D datasets representing different points in time. Our tools allow the fast and intuitive creation of an abstract representation with a minimal user interaction. As input we use

- 2.5D as-planned data,
- Georeferenced point cloud data of the as-built situation,

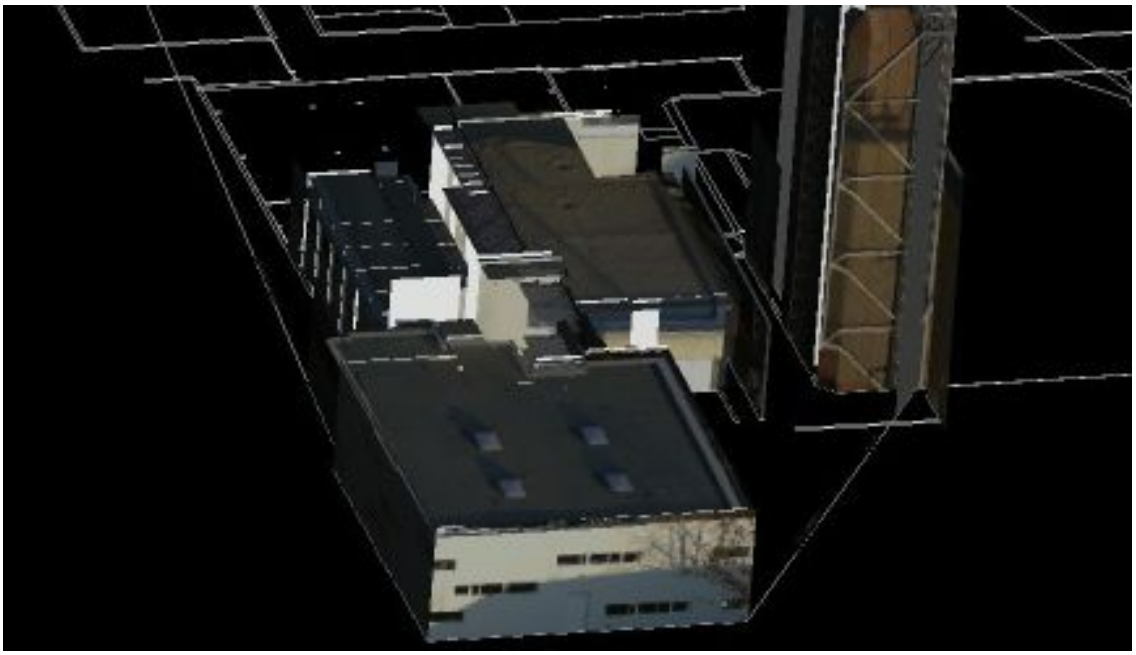


Figure 3.10: Result of creating abstract models. Textured blockmodel after adjusting the dimension of the as-planned data to the dimensions of the as-built 3D data.

- Georeferenced camera images that are registered to the point cloud data.

In a first step, we transcode the 2.5D input data into a 3D representation using an automatic extrusion transcoder. After the transcoding, every polygon is represented as an extruded line with a certain height and depth. If an object is not represented in the 2.5 data, the editor allows one to interactively creating an outline in the scene. As input for the 3D point cloud, we use georeferenced data created by aerial vision (Section 3.2.3).

The as-planned and the as-built dataset have to be registered to each other in order to adapt the abstract as-planned data to the dimensions of the as-built point cloud. To register both datasets, we provide an interactive tool performing a rough manual registration based on the *Absolute Orientation* [55]. For this purpose, the user selects corresponding points from both datasets. Based on these points a transformation matrix is computed. The transformation matrix is used to transform the as-planned data to the as-built data to allow a visual comparison. After achieving this kind of rough registration, the user can apply automatic methods for adapting the 3D extrusions to the exact dimensions of the point cloud.

Automatic Methods for Adjusting Models In order to automatically adapt the abstract as-planned automatically to the dimensions of the as-built point cloud, we analyze the data of the point cloud and derive height, depth and width measurements. For this purpose, we compute histograms based an the bounding box given by the as-planned data. To compute the height, we project all 3D points within this bounding box on the y-axis (up-vector of the bounding box). This projection gives a histogram, that allows us to

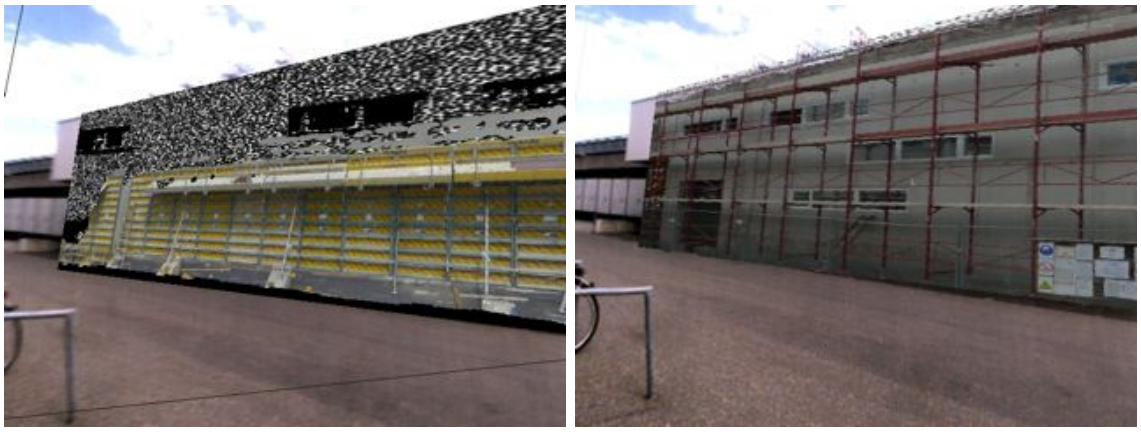


Figure 3.11: Abstract representation of former points in time.

compute the maximum using the histogram entries. Similar methods are used to compute depth and width of the objects.

The result of this adaption step are rectangular abstract objects. In order to texture these objects, we compute the most appropriate camera for texturing and apply projective texture mapping.

Additionally, the editor provides a set of interactive methods for manual improvements:

- Texturing objects manually by selecting registered camera images.
- Creating and deleting objects.
- Manually increasing and decreasing the height or depth of objects.

Results After performing the semi-automatic adaptions using our interactive editor, the results of adapting a 3D point cloud is an abstract representation that consist of several 3D blocks (Figure 3.10). This kind of data can then be displayed in an AR visualization to show former points in time in relationship to the physical world (Figure 3.11).

3.3 Mobile Augmented Reality Setup

For testing AR visualization techniques in an outdoor environment, we developed a system that is appropriate for working in outdoor environments and integrates all sensors that are required for registration.

For this purpose, we equip a ruggedized powerful 1.6GHz tablet PC (Motion J3400) with a set of sensors for registration and data capturing. To support outdoor usage, we use a tablet that provides a screen that is specially built to be viewable outdoors, even under sunlight conditions. The set of sensors consists of a camera (VRMagic VRM FC-6 COLOR), an IMU (XSense MTx 3DOF), a laser measurement device (for surveying tasks) and a GPS receiver. The camera is equipped with a wide-angle lens and is combined with the IMU and mounted on the back of the tablet to point away from the user. The IMU consists of gyroscopes, accelerometers and 3D magnetometers and provides 3 DOF orientation

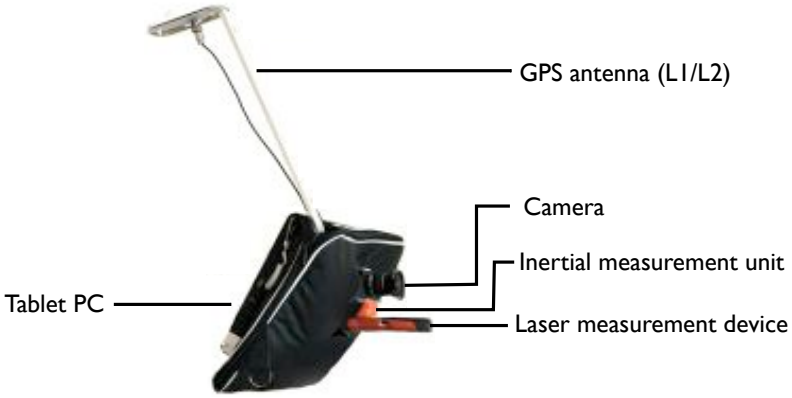


Figure 3.12: Augmented Reality Setup.

measurements. As GPS sensor, we use a L1/L2 RTK receiver that provides measurements of the device's position within centimeter accuracy (Novatel OEMV-2 L1/L2 Real-Time Kinematic).

A laser measurement device is integrated into the system to allow a precise surveying of 3D points in the environment for interactive surveying tasks. All sensors are connected via USB. A leather bag covers all cables and the GPS sensor, to shield them from weather influences (Figure 3.12). The system itself can be carried using a shoulder strap that is connected to the bag or can be used in a fixed position by mounting it to a tripod.

At runtime the sensors work in separate threads and are used to feed the registration methods from Section 3.1 with input data. The output of the registration is used to update the transformation matrix of the virtual content. The rendering itself is performed each time a new camera image arrives.

3.4 Summary

In this chapter, we introduced the methods and systems that are used to implement and test the visualization techniques developed within this thesis in different outdoor environments. We showed that we can use different registration techniques depending on the sensors and the data that is available. Professional sensors are highly expensive, but they allow registrations in nearly completely unknown environments. If more knowledge about a test site is available, e.g., a 3D point cloud, it is possible to reduce setup expenses by using the 3D data as input for model-based localization and registration techniques. In this case, there is no need for high-accuracy GPS and IMU sensors.

Furthermore, the visualization techniques strongly depend on the available data. Convincing visualizations are based on realistic data sources. For this purpose, we described different data sources that we use as input for the visualization of professional data in outdoor applications. We showed how to access this data and how to convert it into displayable 3D geometries.

Finally, we described how registration techniques and display functionality can be combined in a mobile AR system that focuses on outdoor usage and industrial scenarios.

Chapter 4

Physical Pictorial Cues from Camera Imagery

Contents

4.1 Introduction

In this chapter, we address the issue of achieving a seamless integration of virtual content into the physical world by detecting and maintaining physical pictorial cues from camera images. If natural pictorial cues are not sufficiently presented in an AR visualization, the scene will either look unnatural or produce a wrong perception of the order of objects in the scene. Missing occlusion cues in an X-Ray view may lead for instance to perceiving virtual subsurface objects as being floating over the ground.

In Chapter 2 we discussed different cues that are important for for depth perception. Since humans use these cues for building a mental model of the scene, they are also important for a seamless integration of virtual and physical content. If some of them are not available, spatial information is not coherent and the user will note that something is wrong in the scene. Several depth cues are already provided by the general rendering pipeline. But some cues have to be added additionally.

Occlusion cues are assumed to be the strongest of all pictorial depth cues and to work on all distances [23]. Unfortunately, they are not automatically provided by a naïve AR overlay. Occlusion cues give an ordinal measurement of objects. Due to the power of occlusion cues, we will focus on these cues in the following, but we will also discuss some examples where we integrate shadows of virtual objects in the next chapter.

Figure 4.1: Random cues vs. relevant occlusion cues. Left) Random occlusion cues can not transport the depth order. Right) Using important image regions as occlusion cues creates the impression of subsurface objects.

There are two main problems that have to be addressed to provide an adequate occlusion management. Firstly, if there is no accurate 3D representation of the scene available, the order of depth of virtual and physical objects has to be estimated. Secondly, AR allows not for only natural compositions, but also provides the possibility of having an X-Ray view inside physical objects. In this case, it has to be decided which information of physical occluding object is preserved in the final rendering. This decision depends on the visibility of the occluded objects, but also on the minimum number of occlusion cues that has to be preserved to achieve a seamless scene integration. A convincing occlusion management finds the best compromise between sufficient number of occlusion cues, preserving the occluder's structure and object visibility. Thereby, it is important that the cues preserve not only the appearance of the occluding object but also its structure. In Figure 4.1, we show the problem that appears when only random image elements are used as occlusion cues to preserve the appearance of the street.

The compromise between visibility of occluder and occluded object will result in a visualization technique called *ghosting* [31]. The basic idea of ghostings in AR is to preserve selected information from the physical object by rendering it on top of the virtual objects. This then raises the questions what information should be preserved and in what amount (opacity). Former work describes model-based approaches that use a 3D representation of the occluder to determine important scene structures [63]. The disadvantages of model-based ghostings are that often no occluder model exists or that the registration of this model is not accurate. This is in particular a problem in outdoor environments. Even if a perfectly registered 3D model of the occluder exists, the exact texture of the model may be missing. This makes the computation of the adequate amount of preserving difficult. For the case of exact registered data Mendez et al. proposed a method that is based on using pre-defined mask for context preserving [90]. There have also been attempts at using edges extracted from camera images in the case that there is no model of the occluder available [64].

In this chapter, we will describe a technique that extracts occlusion cues solely from

Figure 4.2: Image-based ghostings. Left) Naïve overlay of virtual pipes on top of a video image of an urban scene. Middle) Extracted important image regions. Right) Ghosting preserves important image parts in the final rendering and provides a convincing integration of the virtual pipes into the street scene.

video images. It is based on the assumption that the depth order between virtual objects and physical world is known. This assumption is valid for scenes where all virtual objects are located under or behind the physical objects seen by the camera and would be normally occluded (Figure 4.2, Left). We refer to this assumption as *single layer occlusions*, which applies, for example, in underground infrastructure visualizations.

Our method decides which information of the physical environment should be preserved by analyzing camera images and heuristically extracting key information. For this purpose, we analyze edges, salient locations and texture details from the camera stream. These features are then used as input for the ghostings. Whenever there are too few features to preserve (such as the uniform surface of a table), we add synthetic details that maintain appearance characteristics [99] for compensation. Figure 4.2 contrasts a naïve augmentation with our suggested solution in an outdoor AR application. The left image presents the problem of augmenting virtual data without considering the underlying camera image. The right image illustrates our approach and that it is easier to infer the spatial positions of objects.

4.1.1 Approach

A basic ghosting approach based on alpha-blending would preserve both virtual content and video content in equal measure by using equally half transparent objects. However, this approach would disregard the fact that each image region may require a different amount of preservation due to properties and importance of each region. Our approach addresses the question about what has to be preserved in each image region and in which amount. We analyze the video image of the scene which has to be augmented and calculate a transfer function that maps the video image into a *ghosting map* (see Section 4.3.2). The ghosting map indicates the importance of each pixel in the scene and whether it should be preserved or not. The map incorporates both per-pixel image features and features computed from larger regions (see Section 4.3.1). These regions are computed as superpixels to preserve perceptual grouping. If a region is found to be less important and lacks important structures, we use synthetic region-dependent structures to preserve a sketch-like representation of the region. The user can modify the ghosting map through setting a generic transparency parameter controlling the overall amount of pixels retained

(see Section 4.3.3). Finally, we compute the ghosting map on a larger representation of the environment such as a panoramic image, instead of for each video frame (see Section 4.4.1). This enables us to provide real-time performance and temporal coherence.

4.1.2 Contribution

The main contribution of this chapter is twofold; firstly, to provide a heuristic approach to nominate the information that should be preserved in the ghosting. And secondly, to analyze the amount of said information and complement it with synthetic structure if necessary. We demonstrate the technique using examples from outdoor visualization of infrastructure data such as electricity lines and gas pipes. The examples shown in this section are part of the SMARTVidente[1] project, which investigates AR visualizations of subsurface features. Our main interest lies in outdoor AR applications, which draw from a rich database of virtual representations of the physical world infrastructure.

Finally, we conducted a user study investigating the effect of the image-based ghosting technique on the ability of users of perceiving subsurface object subsurface.

4.2 Foundations for Creating Physical Cues from Camera Imagery

Traditional techniques in medical and technical illustrations [52] already address the conflict between showing hidden structures and preserving the context of the foreground. Artists identify important structures of the scene and preserve them in the illustration. In model-based approaches, automatic feature extraction from the 3D model of the scene replaces the artist's understanding of the scene. In unknown environments, we have to rely on information that we observe of the environment. Here, the video background image provides the first and most direct observation of the scene and therefore, we will analyze the video image to infer context worth preserving.

4.2.1 Transfer functions

To formalize our analysis of the video image of the background scene, we turn to the concept of transfer functions. In the area of volume rendering, transfer functions are used to describe the contribution of every data element to the overall scene. Inspired by traditional illustration techniques, Bruckner et al. adapted transfer functions in volume rendering to preserve context [17].

In volume rendering a transfer function f maps every voxel with the coordinates x, y, z of the volume to a transparency (A or α) and a color value (RGB):

$$RGBA(x, y, z) = f(x, y, z) = p_0 \otimes p_1 \otimes p_2 \otimes \dots \tag{4.1}$$

Bruckner et al. used volume data dependent parameters p_i such as shading intensity, gradient magnitude and distance to the eye to determine the transfer function. Since

[1] Subsurface Mobile Augmented Reality Technology for Outdoor Infrastructure Workers (`http://www.vidente.at/`)

the blending of virtual content and video images in an X-Ray AR scene also requires the mapping of a transparency value to one layer (namely the video layer), the blending function can be seen as a reduced transfer function for transparency values $\alpha(x, y)$ applied to a 2D domain:

$$\alpha(x, y) = p_0 \otimes p_1 \otimes p_2 \otimes \dots \tag{4.2}$$

We will call this reduced transfer function the *ghosting transfer function* and adapt the problem of preserving context in volume rendering to preserve context in X-Ray augmented reality. Volume data dependent information, such as shading intensity and gradient magnitude, are usually not given in X-Ray augmented reality applications. In particular, there is no volume data, nor a model of the scene, but only a flat representation of the world, which is the video image. Therefore, we have a look at parameters that describe the importance of image regions in video images. Given this set of parameters, the transfer functions can be described as a combination of image domain dependent parameters $p_i(x, y)$:

$$\alpha(x, y) = p_0(x, y) \otimes p_1(x, y) \otimes p_2(x, y) \otimes \dots \tag{4.3}$$

We reduce the complexity of the ghosting transfer function by constraining the X-Ray visualizations to *single layer occlusions*. Single layer occlusions assume all virtual objects to be spatially located behind objects in the camera image layer. Thus it is not necessary to determine the depth ordering of physical world objects. The result of the ghosting transfer function is a transparency value for each pixel in camera image space. These transparency values are stored in a ghosting map for application as an alpha mask.

The parameters $p_i(x, y)$ depend on different importance measurements of image regions in the camera image space. These measurements will be identified by investigating the importance of image regions for the human visual system. In order to reflect the Gestalt Laws in our approach, we compute image regions based on perceptual grouping.

4.2.2 Importance of Image Regions for the Human Visual System

Importance of image regions can be divided into two types of factors [96]. High-level factors consider previous knowledge, goals and tasks. An example for high-level importance is a user that focus on red objects in a visual search task for red targets. High-level factors offer useful measurements of the importance of a region to scene perception, but require a detailed understanding of the scene in terms of individual objects. Such information is often not available. Thus we decided to not use high-level factors based on previous knowledge in our approach. In contrast, low-level or bottom-up factors are stimulus-based for fast information processing. Osberger et al. [96] identified several low-level factors; and used them in MPEG encoding to code visually important regions more accurately. These comprise

1. **Contrast**. Osberger et al. state that the human visual system starts processing with converting luminance into contrast. Thus, they assume contrast to be a strong low-level visual attractor and regions with high contrast are likely to have high visual importance. We can analyze contrast at a global scale, at a neighborhood scale and at a regional scale.

2. **Shape**. Osberger et al. further discussed the influence shape on attracting attention. They described edge-like shapes to have a high visual attraction. Compared to regions of the same contrast but with other shapes, edges-like shapes seem to be more likely to attract attention.

3. **Color**. From previous literature, it is known that humans are more sensitive to some colors then to others due to the spectral sensitivity of rods and cones [38, 60]. In photopic vision for instance, we are most sensitive to the wavelength 555nm (light-green). Osberger et al. further discuss that some colors attract the attention of humans more than others. However, the effect of the visual importance of a color depends strongly on the global color of an image. In particular, a strong influence can be measured if the color of a region is different to the background color.

4. **Size**. Another factor that influences the visual importance is the size of a region. Osberger et al. state that large regions are more likely to call attention.

5. **Motion**. We are more likely to notice objects in our proximity if they are dynamic in front of a fixed background. Osberger et al. state that motion has one of the strongest influence in attracting attention.

Other low-level factors include brightness, orientation and line endings. All these factors can be used as input for image space dependent parameters of the ghosting transfer function in Equation (4.3).

4.2.3 Perceptual Grouping

As mentioned in Chapter 2, the Gestalt laws state that perceptual grouping plays an important role for the human visual system. These principles have been widely used for different computer vision applications such as depth estimation [53] and segmentation [82]. Since the success of visualization techniques is strongly connected to the visual perception of humans, we suggest applying visualization techniques not arbitrarily on a per-pixel base but per perceptual groups. In order to use such a grouping, we compute a superpixel representation of the input image, since superpixels satisfy the demand to perceptual grouping in images [102]. A useful side effect of using superpixels instead of pixels is that we can compute meaningful statistics of these groups. Furthermore, the per-superpixel processing is faster than a per-pixel processing. The pixel-dependent parameters $p_i(x, y)$ of Equation 4.3 are then extended to be dependent on the corresponding superpixel r. The function $sp(x, y)$ maps a pixel to its corresponding superpixel r:

$$\alpha(x, y) = p_0(x, y, sp(x, y)) \otimes p_1(x, y, sp(x, y)) \otimes \dots \tag{4.4}$$

One aspect of information preserving that has so far been omitted from our considerations, are the user's intentions, interests and previous knowledge when exploring an *Augmented Reality (AR)* scene. In order to include user preferences, we want to allow the users to adapt the visualization interactively according to their preferences. In their context-preserving approach, Bruckner et al. introduced parameters that allow the user to interactively explore the data sets [17]. For example, the user controls the amount of

Figure 4.3: Different grades of preservation. Left) The road markings are completely transparent. Right) While the road markings are opaque, texture details of the concrete are extracted and preserved.

revelation of the interior of the volume data by modifying one of the parameters. We want to apply a similar concept for our image-based ghosting approach. For this purpose, we introduce a parameter $g_i(r)$ that enables the user to change the preservation grade of an image region (Figure 4.3). The ghosting transfer function from Equation 4.4 is then extended to

$$\alpha(x, y) = p_0(x, y, g_0(sp(x, y)) \otimes p_1(x, y, g_1(sp(x, y))) \otimes \dots \tag{4.5}$$

The final function takes different image space dependent parameters as well as the preserving grades into account and assigns each pixel an alpha value. These values are stored in a ghosting map that is later used to determine the transparency for each pixel.

4.3 Image-based Ghostings

In order to fill the ghosting map with meaningful values, we set up a ghosting transfer function based on various image-dependent properties. Figure 4.4 gives an overview of the involved process. The current view of the environment is captured by the camera. The captured camera image is analyzed on a per-pixel and a per-superpixel-based representation of the image for important information, such as edges, salient regions and texture details. Information that has been considered to be relevant will be preserved by the ghosting transfer function. The function assigns the according values in the ghosting map (Figure 4.4, Middle). In regions that contain no important information, synthetic structures from tonal art maps, such as stipples or hatchings, will be included. The result of the analysis is a ghosting map, which is interactively adjustable by the user. For this purpose, he or she selects reference segments in the image and adjust the preservation grade. The final ghosting map is used together with the camera image and the virtual geometry to create the final ghosting (Figure 4.4, Right and Figure 4.5, Right).

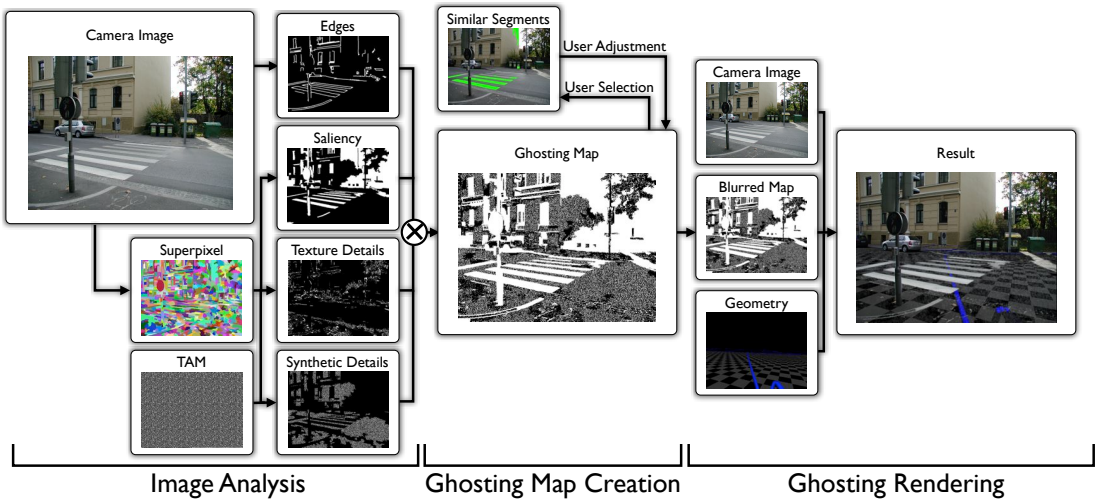


Figure 4.4: Overview of image-based ghostings. The camera image is analyzed for important information that has to be preserved. Important information includes edges, salient regions and texture details. In regions without important information, synthetic structure from tonal art maps (TAMs) will be preserved. The combination of this information results in the ghosting map, which creates together with the camera image and the virtual geometry, the final ghosting.

4.3.1 Importance of Image Regions

The first step is the analysis of the image regarding the importance of each pixel to the overall scene. In Section 4.2.2, we discussed several indicators for the importance of image regions. For our approach, we have decided to focus on contrast, shape and color as importance measurements, since they can easily be computed from one image, without knowledge about image objects. Even though, we only use a subset of features, the approach is easily extendable to more parameters.

Edge-like features are highly important for scene understanding (Section 4.2.2). For this reason, we compute edges in the image to identify visually important parts of the image. We apply the Canny edge detector and use edge chaining to extract connected edge segments [19]. In order to avoid clutter from small edges, we eliminate segments with a size smaller than 10 pixels. This processing step defines a pixel-wise function E(x,y):

$$E(x,y) = \begin{cases} 1, & \text{if a pixel } (x,y) \text{ is on an edge} \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

To measure the influence of color differences we use a method similar to the one described by Achanta et al. [1], who computed the saliency map S for an image as

$$S(x,y) = \|I_\mu - I_\omega(x,y)\|\,, \tag{4.7}$$

where I_μ is the mean image color and $I_\omega(x,y)$ is the image color at (x,y) in a Gaussian blurred version of the image. Instead, we compute the saliency on a per-superpixel basis

Figure 4.5: X-Ray visualization of a virtual room inside a physical building. Left) Simple overlay. The virtual room is often perceived to be located in front of the building. Right) Image-based ghostings. Occlusion cues are extracted from the video image and preserved in the visualization. The virtual room is perceived to be located inside the building.

by using the average image color of a superpixel $I(r)$:

$$S(r) = \|I_\mu - I(r)\| . \tag{4.8}$$

In this computation, we use the CIELAB color space to represent the color vector, since CIELAB takes the nonlinearity of human color perception into account. This consideration allows the usage of the Euclidean norm as a distance measure.

Another importance measurement that we use is the local contrast within a superpixel. The local contrast is computed as a root mean square contrast and can also be interpreted as texturedness $T(r)$ of a defined region

$$T(r) = \sqrt{\frac{\sum_{i=0}^{N-1}(I_i - \overline{I})^2}{N}}, \tag{4.9}$$

where I_i are the intensities of each pixel of the superpixel r, $\overline{I}$ is the average intensity of the superpixel r and N is the number of pixels of r.

4.3.2 Transfer Function for Ghosting

The ghosting transfer function $\alpha(x, y)$ uses the results of the importance analysis as input parameters:

$$\alpha(x, y) = E(x, y) \otimes S(x, y) \otimes D(x, y). \tag{4.10}$$

In our approach the function maps each pixel to a value between preserve (*1*) or replace (*0*). If at least one of the importance measurements gives a positive response for a pixel, the pixel has to be preserved. This composition is represented by the following equation:

$$\alpha(x, y) = \begin{cases} E(x, y), & \text{if } E(x, y) > 0 \\ S(x, y)), & \text{else if } S(x, y) > 0 \\ D(x, y), & \text{otherwise.} \end{cases} \tag{4.11}$$

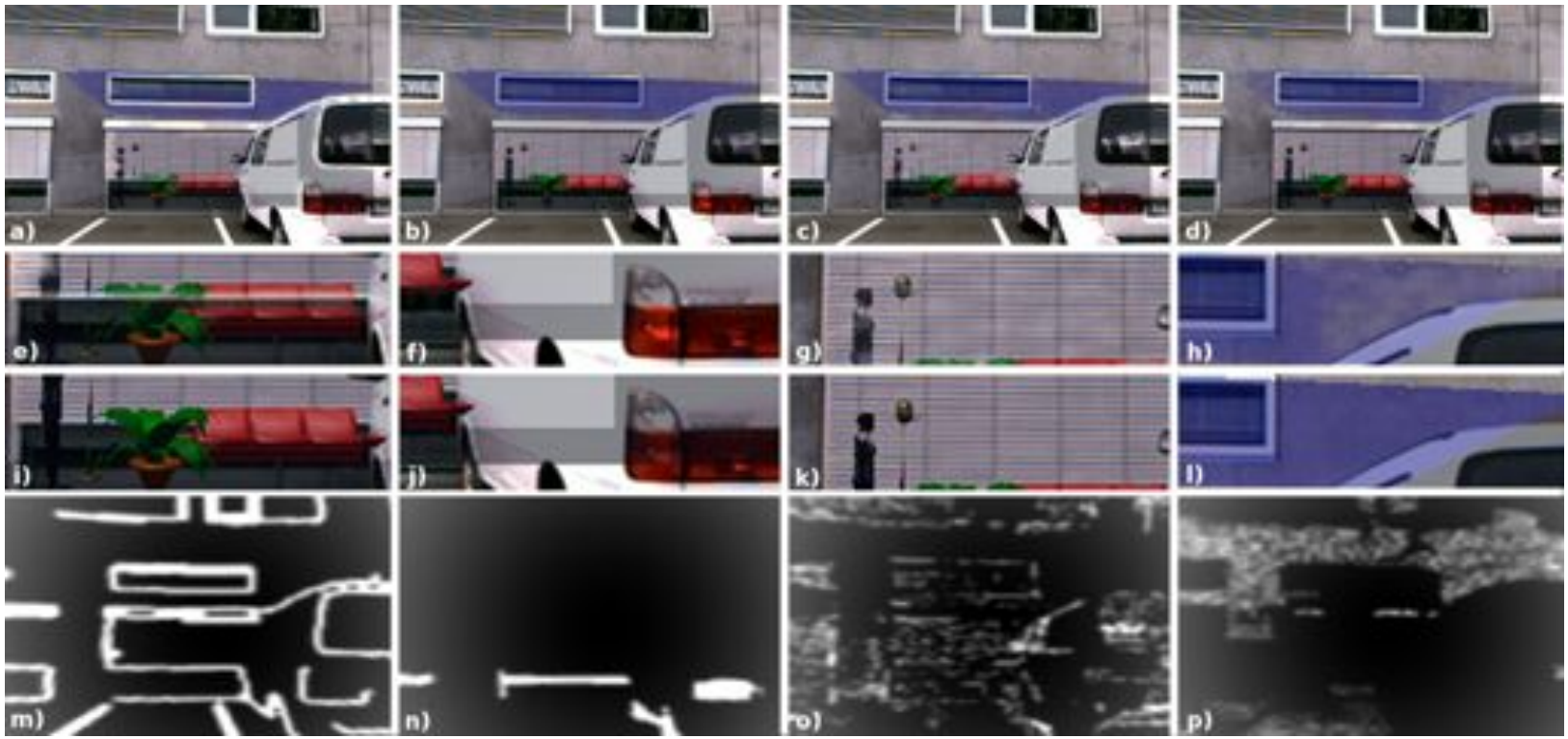


Figure 4.6: Different stages of preserving video information in image-based ghostings. Edges: a) shows a ghosting created with edges m) from the video image. The closeup e) shows that parts of the window frame are preserved. These occlusion cues are not available in the simple overlay i). Salient regions: A ghosting created by the extracted salient regions n) is shown in b). The closeup f) shows that the red car light is preserved as a salient region. This occlusion cue is not available in the simple overlay j). Texture details: Extracted texture details o) are used to create the ghosting c). The closeup g) shows that parts of the curtain are preserved and slightly occlude the virtual man and the light. That provide the occlusion cues that these objects are occluded by the curtain, whereas in the simple overlay k) the virtual objects also may be in front of the curtain. Synthetic details: In d) the parts of the image that contain no important information are preserved by using synthetic structures p) from tonal art maps. The closeup h) shows that the concrete of the building is preserved, instead of being simply replaced by the virtual content l).

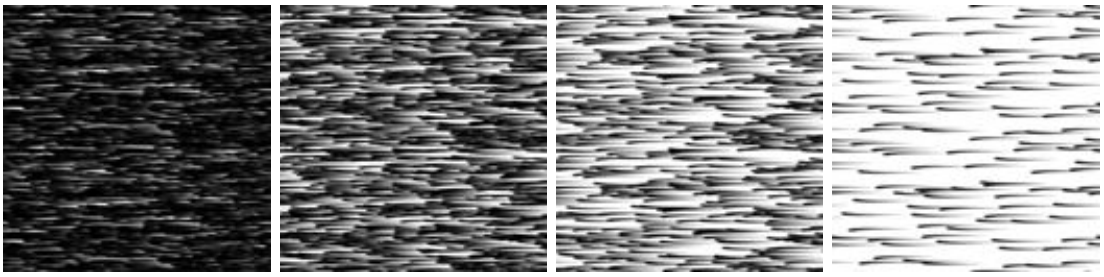


Figure 4.7: Tonal art maps with hatchings. Subsequent textures contain same elements to provide smooth transitions over the intensity ramp.

- $E(x, y)$ is the edge representation of the image and is 1, if a pixel (x, y) belongs to a edge. Figure 4.6 m) shows an example for the edge representation and 4.6 a) the usage of edges as input for creating a ghosting image.
- $S(x, y)$ describes the saliency measurement for each pixel and is given by saliency value of the corresponding superpixel $S(x, y) = S(sp(x, y))$. The value is binarized by a threshold. Figure 4.6 n) shows extracted salient regions for an image and 4.6 b) shows the corresponding ghosting image.
- $D(x, y)$ determines if a pixel should be preserved as a detail. $D(x, y)$ depends on the texturedness of the corresponding superpixel $T(sp(x, y))$ and will be defined in the following. An example for the preservation of details are shown in Figure 4.6 c) and b). The corresponding preserving information are shown in Figure 4.6 o) and p).

The function $D(x, y)$ preserves details for regions with a high level of texturedness by the texture extraction function $D_T(x, y)$ (Figure 4.6 c) and o)) and for regions with a low texturedness by synthetic detail extraction $D_S(x, y)$ (Figure 4.6 d) and p)). Low texturedness is thereby defined by a threshold T_{min}

$$D(x, y) = \begin{cases} D_T(x, y), & \text{if } T(sp(x, y)) > T_{min} \\ D_S(x, y), & \text{otherwise.} \end{cases} \tag{4.12}$$

For image regions with a high texturedness the details $D_T(x, y)$ are extracted from the texture. We define texture details as pixels with a high difference in luminance to the average luminance of one segment. Thus, for preserving the texture details we compute the details by

$$D_T(x, y) = \begin{cases} 1, & \text{if } |L(x, y) - \overline{L}| > (1 - g_d(sp(x, y))) \\ 0, & \text{otherwise,} \end{cases} \tag{4.13}$$

where $L(x, y)$ is the luminance of a pixel, $\overline{L}$ is the average luminance, and $g_d(sp(x, y))$ is the preservation parameter of the corresponding region. The preservation parameter controls the amount of preservation and will be discussed in Section 4.3.3.

Image regions that contain no important image information have a problem to transport occlusion cues. This means in flat untextured regions occluded objects are often perceived to be in front. In order to address this problem, we use synthetic structures for

preserving occlusion cues, such as hatchings and stipplings, similar to the techniques for illustrative renderings of Interrante et al. [57]. For this purpose, we detect image regions with a low level of texturedness and add these synthetic structures to the rendering using $D_S(x, y)$ in Equation 4.12.

We define the density of the synthetic structures to be dependent on the intensity of the image pixel [52]. Smooth transitions between regions with different intensity levels are provided by pre-computed *tonal art maps* (TAM) [99]:

$$D_S(x, y) = TAM((L(x, y) + g_d) * n_{tam}. \tag{4.14}$$

TAMs are a sequence of n_{tam} textures; used to represent different levels of intensity that were originally applied to render hatched and stippled drawings in non-photorealistic rendering. Since each texture consists of the same elements (in our case of either stipples or hatches) as all of its subsequent textures (Figure 4.7), TAMs allow smooth transitions over the entire intensity ramp.

The result of the ghosting transfer function is an alpha value for each pixel. In a final step, we smooth the output of the ghosting transfer function using a Gaussian filter in order to reduce high frequency artifacts,

4.3.3 Adjustment

As described in Section 4.2.2 our method so far considers only stimulus-based information for preserving content of the video image. A possibility to include the users objectives, personal preferences and previous knowledge, is to allow the user to interactively control the overall amount of pixels retained by the generic transparency parameter $g_i(r)$ from Equation 4.5. While the user abstractly controls the amount of occlusion by the parameter, our approach takes care of adding more relevant or synthetic structures to the overlay. For this purpose, we use the parameter g_d in Equation (4.13) and (4.14) to change the amount of details. We define the range of the parameter to be 0 to 1. Large values of g_d preserve more texture details, since the threshold for using a pixel as texture detail is getting lower. Similarly, the large values increase the offset of Equation 4.14, which means the selected level of the tonal art map is higher. This results in a higher density of the synthetic structures and finally more synthetic details are preserved. If the preservation parameter is 1 all pixels will be preserved and if the parameter is 0 no pixel will be preserved. In order to change the generic transparency parameter, the user selects one image segment and adjusts its parameter.

Since the individual selection and adjustment of the preservation parameter for each segment may be very time-consuming, we provide a possibility to select similar image regions together at once. A descriptor of each region, computed from some of its normalized characteristics helps us to find similar regions. The characteristics are texturedness, luminance, a* and b* from the CIELAB color space, entropy and edges. The similarity for regions is defined as Euclidean distance between the descriptors of the two regions. We identify all regions as being similar whose descriptor distance to the selected region descriptor is below a certain threshold. These segments will be added to the selection (see Figure 4.8). After selecting a set of regions, the user increases or decreases the parameter

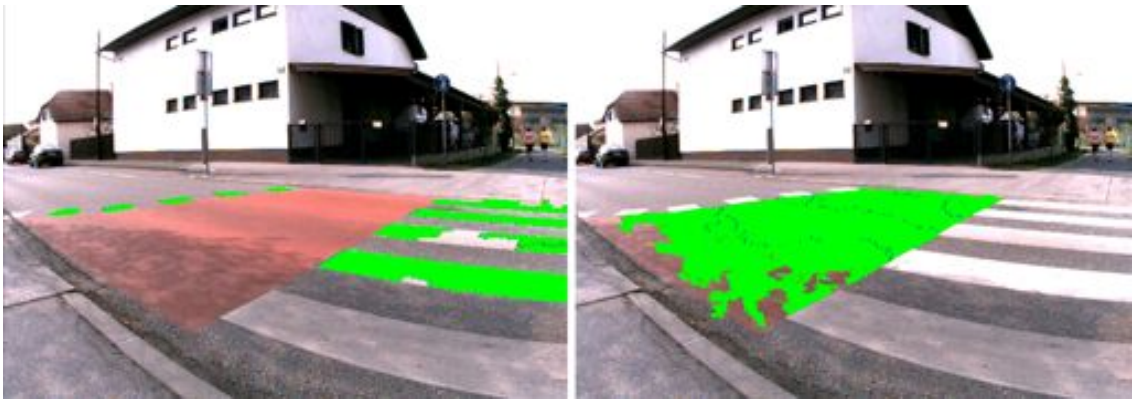


Figure 4.8: Examples of the selection of similar image regions. Left) Selection of white street markings. Right) Selection of red street markings.

g_d to change the amount of video foreground that is preserved.

4.4 Implementation

We tested our method with the mobile outdoor AR system described in Chapter 3. For registration, we used the multi-sensor approach (Section 3.1.1). In the following, we discuss how we integrated the image-based ghosting approach into this system for visualizing underground infrastructure.

4.4.1 Panorama-based Ghosting Map

As discussed in Section 4.2.3 we use a superpixel representation of the video image to enable perceptual grouping. The calculations of importance values are done partially on the superpixel representation. We decided to use the EGBIS algorithm of Felzenszwalb et al. [32] for computing a superpixel segmentation, since it preserves details in low-variability image regions, ignores details in high-variability regions, and it is fast compared to similar approaches.

Unfortunately, the EGBIS algorithm is still too slow for real-time applications. Furthermore, even if it could provide real-time performance, a lack of temporal coherence in the segmentation between consecutive frames can lead to flickering as segments change due to image noise and small motions. Since the orientation tracker of the multi-sensor registration approach uses a panoramic image of the environment to track the orientation, we decided to use this panoramic map to compute the ghosting map once in advance. The ghosting map is calculated on the panoramic image and remapped for each video frame based on the current orientation.

Given the tracked orientation for the current video frame, we transfer the ghosting map from the panorama image into the video frame through an inverse mapping. Every pixel p_c in the video frame is projected into a global 3D ray and intersected with the panorama cylinder (see Figure 4.9). The corresponding pixel p_m in the panorama ghosting map is

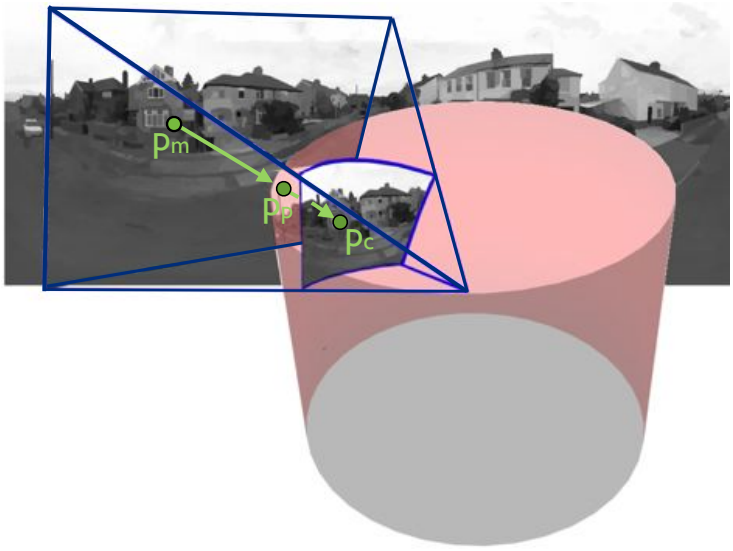


Figure 4.9: Panorama remapping. This illustration shows the mapping from the panoramic map p_m over the panoramic cylinder p_c into the current camera image p_c.

read and mapped into the video frame. Bilinear filtering is used to avoid aliasing in the resulting map.

For selecting image regions for adjusting the transparency parameter, we apply the same projection. Here, the user selects a reference superpixel by selecting a pixel in camera image space. The mouse coordinates are also projected from camera image space to panorama map space given the current camera orientation. The coordinates in the panorama image are then compared with the list of superpixels and the superpixel at this position is selected.

The main advantages of the panorama-based method of creating image-based ghosting are that it allows real-time processing and preserves frame-to-frame coherence in the ghosting. However, it is important to note that the panoramic remapping process can introduce small inaccuracies depending on the size of the panoramic map. If the map is too small, too many pixels of the camera image space share the same pixel of the panoramic map. Therefore, the panoramic map should have an adequate resolution. For our tests, we used a map have a resolution of 1000x500 pixels.

4.4.2 Creating the Ghosting Image

A series of steps composites the final ghosted view. A ghosting image is generated combining RGB channel of the current video frame with the calculated ghosting mask as the alpha channel. The ghosting itself is then created by rendering the ghosting image on the top of the rendered view of the 3D content with blending. Blending is set to overwrite the virtual 3D image, if the alpha mask is opaque, and to not modify it, if the mask is transparent. The complete rendering process is:

Listing 4.1: Compositing of Ghosting.

```
Acquire video image
Track orientation
Transfer ghosting map from panorama space into camera space
Create ghosting image from video image and ghosting map
Render 3D model
Render ghosting view with blending.
```

Video image pixels are rendered with an alpha value between 0 and 1. The alpha value is defined by the ghosting mask.

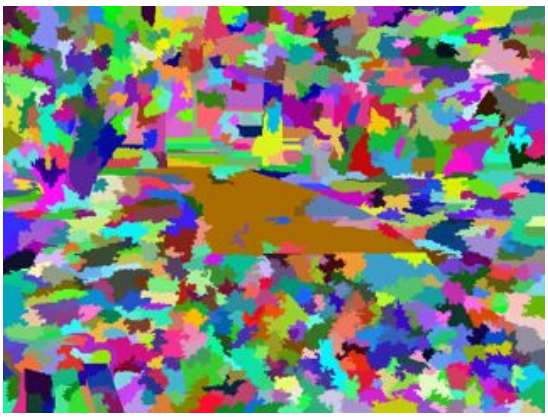
4.5 Results

We applied the image-based ghosting approach in the project SMARTVidente for subsurface infrastructure visualization. Underground infrastructure such as electrical and communication cables, water and heating pipes are shown to users in a registered AR display. Often, no curate surface structure model of the environment is available for these kinds of applications. Thus, traditional model-based approaches to X-Ray visualization are not applicable.

Figure 4.2 shows an example result of our approach. In a simple blended overlay (Figure 4.2, Left) the underground infrastructure appears to hover over the street, since essential perceptual cues are missing. Using our approach, we analyze the camera image and create a ghosting map (Figure 4.2, Middle). The ghosting map determines which camera image pixels have to be preserved or replaced by virtual content. The final compositing uses the ghosting map, virtual content and the video image as shown in Figure 4.2 (Right). In this picture important structures and thus essential perceptual cues of the camera image are preserved. The virtual pipes appear to be located under the street. We display an additional background behind the pipes (here a checkerboard structure) to provide additional occlusion cues. Figure 4.5 shows another example of using image-based ghostings. Instead of displaying underground infrastructure information, in this picture a room inside a building is revealed. In this case, the simple blended overlay (Figure 4.5, Left) could lead to the impression that the virtual room is located in front of the building. By preserving the extracted occlusion cues (Figure 4.5, Right) the scene comprehension is supported and the room appears to be located inside the building.

A problem with our current solution is that we use only bottom-up analysis of image regions and no complete object recognition. Hence, one object can be cut into different regions and be preserved in different styles. Figure 4.10 shows a ghosting where this problem occurs. The pavement in this scene is not detected as one object and is rendered with different styles. The regions of the pavement in the front are preserved by hatchings, since the regions have been found to be lowly textured. On the other hand, the large part of the pavement in the back is preserved by texture details, because it was found to be textured. The superpixel representation in Figure 4.10(b) shows the reason for the different selection of ghostings styles. While the front of the pavement is represented by small superpixels, the back is represented by only one large superpixel. Shadows on the pavement with different intensities increase the texturedness of the region. Generic object recognition could solve this problem and ghostings could be applied to each object

(a) Original AR scene.



(b) Superpixel representation of image.

(c) Image-based ghosting map.

(d) Ghosting.

Figure 4.10: Problems of image-based ghosting. The pavement is preserved in different styles, because it is not recognized as one object. In the front the pavement is preserved using hatchings, for regions in the back texture details are used.

individually. However, to work in general scenes, the recognition method would essentially need to provide a full understanding of the scene, otherwise it would only deliver a sparse interpretation of a subsets of objects. This is still a topic of research.

4.5.1 Computation Times

We also analyzed the computation times of our approach. We compared the computation on a desktop PC (Intel Core 2Quad CPU Q9400, 2.6GHZ) and a tablet PC (Motion J3400 with 1.6GHz Pentium CPU). As discussed before so far the computation runs not in real-time (Table 4.1). This is mainly due to the computation of the superpixel representation. At the time writing this thesis, methods became available that allow a computation of

Table 4.1: Computing times for image-based ghostings.

Image Size	160x120	320x240	640x480	1000x500
Desktop	0.113s	0.225s	0.758s	1.89s
Tablet PC	0.172s	0.343s	1.186s	2.917s

superpixel representations in real-time on the GPU [101].

4.5.2 User Evaluation

In order to understand if our techniques has a benefit compared to existing ghosting techniques, we investigated the effect of technique on depth perception in a user study. Within the study we compared our image-based ghosting method (G_{IB}) with a simple alpha blending (A) and a state-of-the-art ghosting technique using edges for preserving image features (G_E) [6]. The goal was to investigate if the image-based ghostings perform better than alpha blending and edges in terms of depth perception. Furthermore, we analyzed if the user is still able to understand the shapes of the hidden virtual objects in the scene, even if these objects are partially occluded.

Hypotheses We hypothesized that participants understand the subsurface location of virtual objects better using image-based ghostings (G_{IB}) than using alpha-blending or the state-of-the-art edge-based ghosting technique (G_E). Furthermore, we hypothesized that the visualization technique has no influence on the ability of perceiving the occluded shape.

- H1: Image-based ghostings will outperform edge-based ghostings and simple alpha-blending in terms of a convincing depth perception. Using image-based ghostings, the user will have a stronger perception that objects are located subsurface.
- H2: The choice of visualization technique has no influence on the correctness of perceived shapes. The HVS of the users will complete shapes automatically.

Experimental Platform The comparability between the test scenes and the possibility to perform the study on a set of different test scenes with different characteristics had a high priority during the design of the study. Furthermore, we wanted to preclude external influences such as an unstable tracking from our study. In order to achieve these goals, we decided to prepare a set of static AR scenes in advance using the three different visualization techniques. This further has the advantage that we are able to nearly exclusively investigate the occlusion cues, since depth cues resulting from motion could be excluded.

All scenes contained urban scenarios that are common for the inspection of subsurface infrastructure. We differentiate between street scenes that contain a lot of important information that seems to be important such as cross walks and scenes containing less important information such as plain streets or grass. In addition, we used two different types of content, content that belongs to a scenario of inspecting subsurface infrastructure (red pipes) and abstract content (red spheres with different sizes).

Figure 4.11: Test scenes for the survey. Left) Condition Alpha Blending A, Middle) Condition Edge-based Ghostings G_E, Right) Condition Image-based Ghostings G_{IB}.

The settings for the visualization techniques were fixed for all scenes. For the alpha blending the composition is computed as follows $C_A = \alpha V + (1 - \alpha)P$. We set the value for α to a fixed value of 0.5 for the study. For computing the composition for G_E we used edges that were extracted with the same methods as described in Section 4.3.1 and binarized in an alpha map ($\alpha(x, y)$). The composition for G_E as well as for G_{IB} is then given by $C_G = \alpha(x, y)V + (1 - \alpha(x, y))P$.

Task and Procedure We divided the study in two tasks. At first, the user had to inspect each scene and give a rating about her depth perception by using the input of keyboard. Thereby, the depth perception was on a Likert scale ranging from 1 = strongly underground, 2 = underground, 3 = rather underground, 4 = undecided, 5 = rather overground, 6 = overground and 7 = strongly overground. We told the participants before, that the scenes may contain subsurface as well as overground objects. Nevertheless all scenes contained subsurface objects. We decided to do so in order to give the user no previous knowledge about the scene configuration and giving them the complete freedom of choosing the spatial location of the virtual objects.

After completing this task, the user was asked to draw an outline of the virtual objects for scenes that contained virtual pipes. We compared the filled outlines with a binary mask of the virtual objects. The difference between both mask resulted in the contour difference that was used to determine the ability of users to correctly understand the shape of the object. To compute the contour difference C_{Diff}, we used the amount of pixels that differ from the ground truth mask n_{diff} and divided it by the amount of pixels n_{gt} from the ground truth mask $C_{Diff} = \frac{n_{diff}}{n_{gt}}$.

This task was repeated for 12 different scenes using the same visualization technique, but showing different content. After finishing these scenes, participants were asked about their experience with the applied visualization technique. Afterwards, the technique was altered and used for the same scenes as before. The order of the scenes was thereby randomized. The order of the visualization techniques was randomized using Latin Squares. In a final questionnaire, we asked the participants to give a rating on their preferences according depth perception, coherence and general comprehension. The overall study duration for each participant was approximately thirty-five minutes.

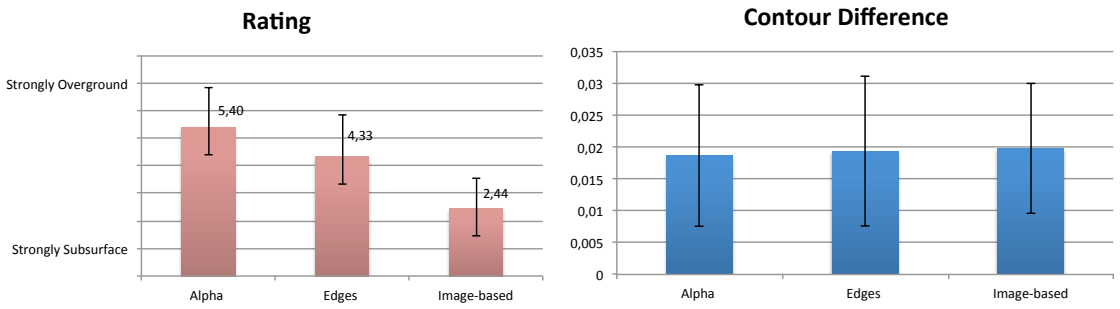


Figure 4.12: Results from the pilot study. Left) Rating the subsurface location of the virtual content. Right) Contour difference measurement in %.

Pilot Before we started with the main study, we conducted a pilot with five users to find out, if our experimental design is sound and to understand if the test is too exhausting for the participants. From the user feedback during the pilot study, we learned that we should remove the abstract shape condition for the contour drawing since there the participants reported that these shapes were too simple and to easy to complete and on the other hand quite exhausting, due to the sphere shape. The pilot also showed that the participants seem to perceive the subsurface objects more being located underground when using our technique G_{IB} (compare with Figure 4.12, Left, average rating 2.43). Contrarily, for G_E they seem to be rather undecided (average 4.33) and for A they seem to rate the location more being overground (average 5.39). While the rating for the spatial arrangement of the objects in the scene seemed to be different, these findings encouraged us to proceed with the given study design with the described minimal adjustments.

Participants We invited 15 people from different universities to take part in this experiment (5 female, 10 male, age ranging from 22 to 35. The experience with AR of the participants ranged from not familiar at all to very familiar. We used a repeated measure design for the study. Each participant performed all three visualization techniques A,G_E and G_{IB} for all 12 scenes.

Results For each participant we averaged the depth perception rating and the contour deviation for each technique, resulting in an overall depth perception rating and an overall contour deviation. We performed a repeated measure ANOVA on this data in order to analyze the effect of technique on overall depth perception rating and overall contour deviation.

The output of the ANOVA for overall depth perception rating shows that the F-statistics is 71.685 with a p-value ≈ 0. We can clearly reject the null hypothesis of equal means for the overall depth perception rating of all three visualization techniques. This shows that there is an significant effect of technqiue on depth perception.

In order to find the significant differences between the single techniques, we used a post-hoc test. The pairwise T-Test (P value adjustment method: bonferroni) showed that there are significant differences between all three methods. G_{IB} showed a significant better perception (M = 2.95, compare with Figure 4.13, Left) of the subsurface location of the

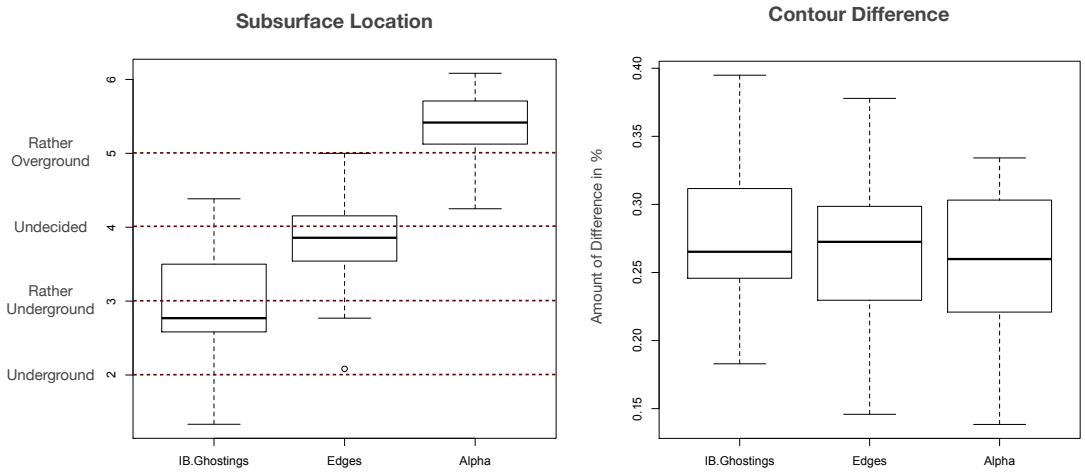


Figure 4.13: Results user study for ghosting techniques. Left) Results of the depth perception task. Right) Results of the accuracy test.

virtual objects than the simple blending A (M = 5.37, A - G_{IB}: p= 4.8e-12) and state-of-the-art ghosting G_E (M=3.79, G_{IB} - G_E: p = 0.0042). G_E also performs better than A(A-G_E: p= 2.6e-07). This confirms hypothesis H1 that the image-based ghostings are outperforming edges and alpha blending in terms of transferring the subsurface location of objects. Users have a stronger perception that objects are located subsurface.

The output of the ANOVA for overall contour deviation shows that F= 1.204 and has a p-value p=0.31502. We cannot reject null hypothesis of equal means for the accuracy of outlines during usage of the three visualization techniques. Consequently, there is no difference between the techniques, which confirms hypothesis H2 that the visualization technique has no influence on the shape perception and users can find the outline with the same accuracy.

We could confirm the findings on depth perception test with results from the questionnaires of the study. After working with each technique the user were asked to rate this technique according to the questions "A: The subsurface visualizations using the X-Ray technique was confusing", B: "The subsurface location of virtual objects in the scene was hard to understand" and C: "The shape of the virtual objects was complicated to understand.". For question A and B, participants rated our technique G_{IB} better than the other techniques (Figure 4.14). G_{IB} was rated with 2.3 for question A and 2.3 for question B. The rating reflects an average rating between "disagree" and "rather disagree" for our technique. In contrast, G_A was rated with M=4.2 for question A and M=4.9 for question B, which reflects a value between undecided and rather agree. For the method G_E, the participants seemed to be rather undecided, since they rated the technique with M=3.2 for question A and M=3.4 for question B, a value between rather disagree and undecided. We found an significant effect between technique and both questions (ANOVA for question A: F=7.188, p=0.00302 and B: F=10.334, p=0.000435). The pairwise T-Test shows only significance between our technique G_{IB} and the naïve overlay G_A (question A: p=0.004 and question B: p=0.00013). This means on the one hand, that G_E performs not

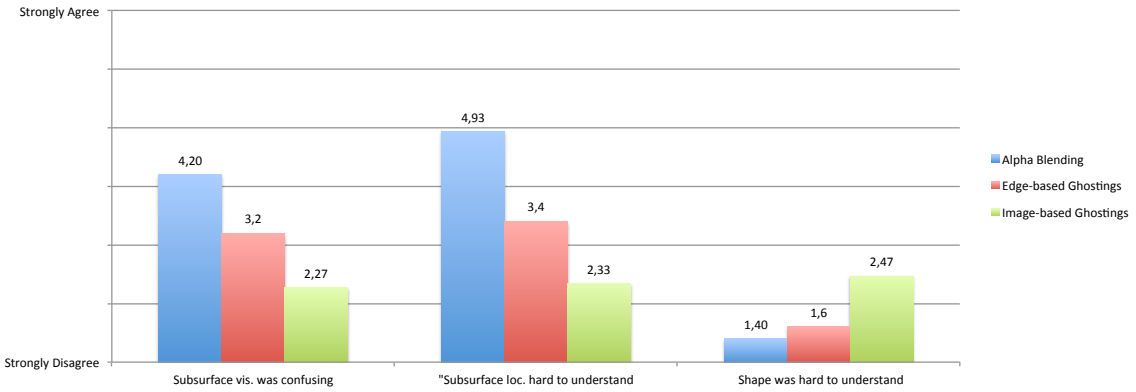


Figure 4.14: Results user study: User ratings for each technique for the following questions. "A: The subsurface visualizations using the X-Ray technique was confusing", B: "The subsurface location of virtual objects in the scene was hard to understand" and C: "The shape of the virtual objects was complicated to understand."

significantly better then the naïve overlay. On the other hand, it shows that our techniques does perform significantly better.

The third question shows that even if the participants showed similar performance on understanding the shape, they rated the comprehension of the shape of the virtual objects slightly more complicated than with the other techniques (question C: $G_A = 1.4$, $G_E = 1.6$, $G_{IB} = 2.5$). This measured difference between our technique and the alpha blending techniques was significant (ANOVA F=10.334, p=0.000435, T-Test $G_{IB} - G_A$: p=0.034). Nevertheless, the rating still indicates that they disagreed or rather disagreed that the shape was complicated to understand. Together with the quantitative measurements, it seems that it was more complicated compared to the other techniques, but still possible.

In a final questionnaire, we asked the participants to rate their preferences in terms of depth perception, coherence and general comprehension. They were asked to give their ratings according to their preferences starting with one for the favorite. As shown in Figure 4.15, the users preferred our technique for all the questions over the other techniques (G_{IB}: depth perception M=1.2, coherence M=1.4, general M=1.4, G_A: depth perception M=2.7, coherence M=2.7, general M=2.6, G_E: depth perception M=2.1, coherence M=1.9, general M=2,). The ANOVA showed significance for the effect of technique on the rating for (depth perception: F=20.24 p=3.65e-06, coherence: F= 11.82 p=0.00019 and general comprehension F=7.875, p=0.00193).

For the depth perception the pairwise T-Test shows a significance between the ratings for all three technique ($G_{IB} - G_A$: p= 3.5e-09 , $G_{IB} - G_E$: p=0.00022, and $G_A - G_E$:p=0.00475). This means that the participants clearly prefer image-ghostings for depth perception. The pairwise T-Test for the question asking for the coherence of the presented X-Ray techniques shows significant effects between both ghosting techniques G_E and G_{IB} against the naïve overlay G_A ($G_{IB} - G_A$: p=1.8e-06 and $G_E - G_A$: p= 0.0013). Although G_{IB} was rated with a higher preference and shows better performance during the depth estimation tests, the user ratings have no significant difference in the perceived coherence between G_E and G_{IB} (p= 0.1388). Finally, for the rating on general comprehension the

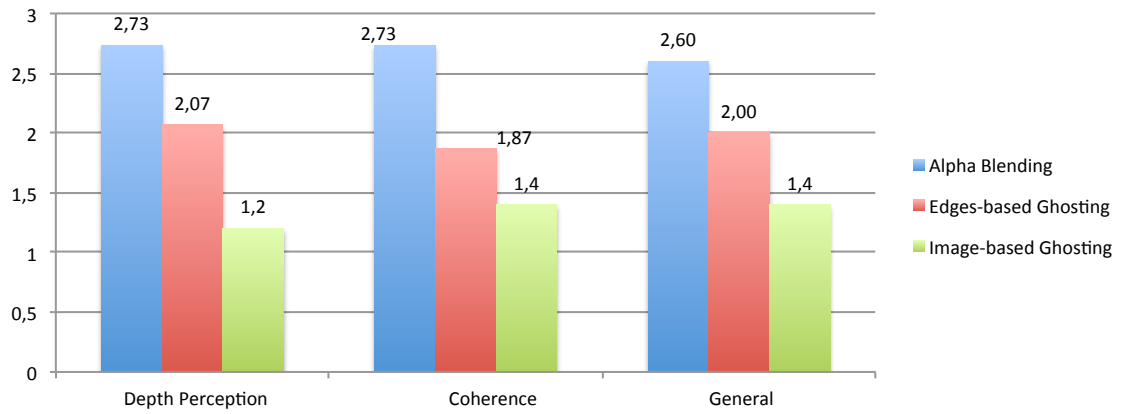


Figure 4.15: Results user study: User preferences on depth perception, coherence and general comprehension.

pairwise T-Test (P value adjustment method: bonferroni) indicates that there is only a significant difference between G_{IB} and our techniques G_{IB} (G_{IB}-G_A:p=5e-05). It seems that the participants preferred our technique over the alpha blending. The edge-based technique was in general not rated as being significantly better than the naïve overlay. This is interesting, since it confirms our assumption that for these urban scenes with an AR overlay showing subsurface object, the edge information provides not enough depth cues to improve the comprehension in comparison to a naïve overlay as given by the alpha-blending.

4.6 Summary

In this chapter, we presented an automated ghosting strategy by creating ghostings from video images. For this purpose, we extract information that is important to scene understanding from the video image. We combine this information into a ghosting transfer function and use the output of the function to map video image pixels into transparency values to control the overlay in the AR view. By performing a user study, we were able to show that this technique helps the user to understand the subsurface location of virtual objects better than using just a simple overlay or state-of-the-art ghosting techniques.

The current approach relies on simple bottom-up features of images to provide a heuristic decision between ghosting methods. More high-level concepts such as recognizing whole objects, or using 3D information could make the grouping of image regions more robust. Here, we are looking at adding dense depth information to panorama maps to allow segmenting whole objects based on depth coherence.

Furthermore this approach is only working for single-layer occlusion where the depth order is known, for instance by a given semantic. If the depth order is not available, we need to apply different methods for occlusion management that extract the depth information first. This will be discussed in the following chapter.

Chapter 5

Physical Pictorial Cues from Sparse Models

Contents

5.1 Introduction

In this chapter, we continue our work on methods for creating physical pictorial cues for AR visualization. In contrast to the methods of Chapter 4 that use no additional source of information but the camera images, in this chapter, we will describe methods that benefit from a sparse description of context information, such as given by a GIS database. These databases often contain no accurate 3D models, but sets of georeferenced features describing infrastructure elements. We refer to this information as *sparse models*, similar to sparse point clouds created from image-based reconstruction. In the following, we will start with discussing methods that directly use these sparse representations as visual cues. For instance, by rendering sparse virtual representations of physical world objects as occluders (Section 5.1.1) or using the sparse data for occlusion culling (Section 5.1.2).

However, we will show that in a lot of cases the sparse information is not sufficient for providing pictorial cues, since it misses physical world detail. We will address this issue in Section 5.3 by presenting methods that allow supplementing sparse information with additional data from camera images. This is only possible, if an accurate registration of the sparse data is available.

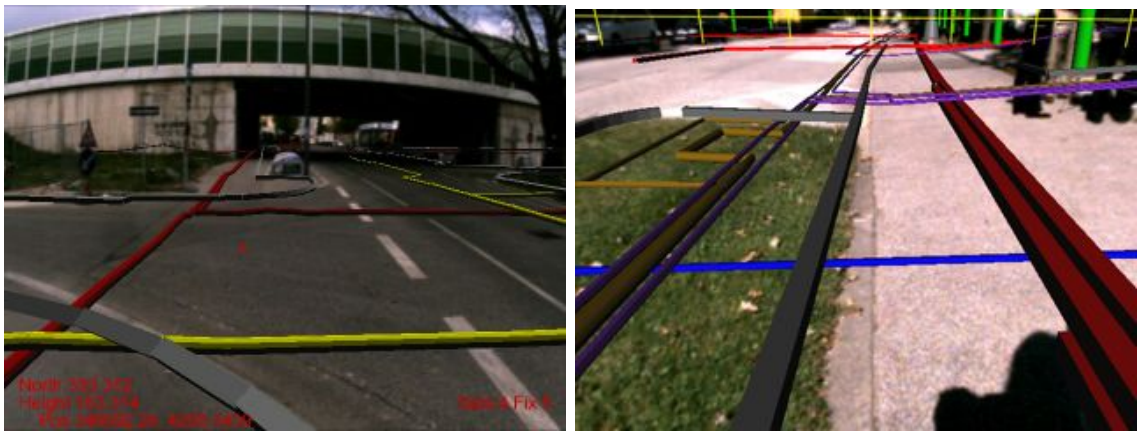


Figure 5.1: Virtual representations as physical pictorial cues. The virtual counterparts of the physical curbstones and grass boundaries are rendered in grey on top of the occluded pipes.

5.1.1 Sparse Geometries as Physical Pictorial Cues

Professional geographic databases such as GIS provide a rich set of information about physical world objects. In the last chapter, we used this information to visualize subsurface infrastructure. But these databases provide geometric information about overground objects as well.

We can use information to support the scene comprehension by rendering sparse geometric representations of important physical objects in addition to the virtual subsurface objects. This is demonstrated in Figure 5.1 (Left), where the virtual grey curbstone is rendered as overlay onto the urban street scene and the subsurface objects. The resulting occlusion yields depth cues and creates the impression that the pipes are located beneath the curbstone. Physical objects from GIS databases that can be used for providing physical cues are for instance curb stone edges, building outlines or manholes.

In the compositing step that combines virtual and physical information, the correct occlusion is achieved by using a rendering with enabled depth buffer testing. During the rendering process only those fragments of the virtual subsurface geometry are rendered that pass the depth test against the sparse geometries representing physical world objects.

GIS databases contain a huge amount of information. Accordingly, it is important to select the infrastructure elements that should be used for occlusion culling. This decision has to be provided by an external logic, such as by an application designer. A filtering method on the GIS data allows us to only include selected geometries in the rendering. Only for those objects a virtual counterpart is created and rendered on top of virtual subsurface objects. The advantage of using sparse geometries as physical cues, is that it is also working in cases where the accuracy of the data is slightly offset. In this case, the geometries of physical objects as well as the geometries for subsurface objects are mislocated with the same offset. The main drawback is that the user has to mentally restore the relationship between the physical world objects and their virtual representation.

Figure 5.2: Using sparse data for occlusion culling. Left and Middle) Hidden virtual information that is located under the curbstones is culled from the rendering to preserve the physical information of the curbstones. Right) For this purpose we create a mask based on the virtual representation of physical objects of interests.

5.1.2 Sparse Geometries for Occlusion Culling

Instead of rendering sparse geometries for creating occlusion cues, sparse data can also be used as input for occlusion culling. In this case, we compute an occlusion mask M containing the sparse geometries. The final view is composed by combining virtual subsurface geometry V with the video image representing the physical environment P based on the occlusion mask M:

$$C = (1 - M)V + P. \tag{5.1}$$

In order to compute the masking, we use the stencil buffer. The virtual representations are rendered into the stencil buffer to create a stencil mask (Figure 5.2, Left). The stencil buffer is then used in the next rendering step to cull the rendering of virtual hidden objects at the fragments where the mask entries are positive. Finally, this results in the final compositing C as shown in Figure 5.2 (Left and Middle). Here, at fragments representing the curbstone the video image is rendered instead of the virtual hidden pipe.

There are two requirements to apply this method successfully. 1) The sparse geometries have to be accurately registered as otherwise wrong occlusion cues are created. 2) Important elements have to be accurately stored in the database. If the information is too inaccurate, important occlusion cues will get lost.

When we use physical object representations for occlusion culling, it may happen that too much information of the underlying content is discarded. Dynamic occlusion masks can help to avoid this problem. Instead of rendering the video image completely opaque at fragments containing important physical world objects, this method generates an importance mask that varies over time. Dynamic occlusion masks were introduced by Mendez to reveal the hidden content of a paper box [89]. We apply this method for sparse geometries given by GIS databases (Figure 5.3). During run-time, a vertex shader and a fragment shader create a Perlin noise function [98] to create a dynamic mask. The advantage of this technique is that it reveals hidden structures over time. Thus, the user sees different parts of the hidden object at different points in time. Furthermore, it does not depend on the amount of features of the occluding object, this means it can also be applied for less textured regions.

Figure 5.3: Using sparse data in combination with Perlin noise for occlusion culling. Left) Perlin noise mask mapped to the objects of interest. Middle and Right) Hidden virtual information that is located under the curbstones are culled from the rendering to preserve the physical information of the curbstones based on the mask.

5.1.3 Dense Depth Maps from Sparse Geometries

Both previously presented visualization techniques often suffer from the sparseness of the GIS data. Objects in GIS databases are represented by geometric primitives such as points, lines or polygons, usually representing only the 2D projections of the real objects onto a map ground plane. Even if a height value is stored, a GIS feature only represents the point or line on the surface of the earth but does not describe the 3D shape. If the data is too sparse, for instance if information about the height, dimensions or extent of an object is missing, it becomes difficult to use this model data for presenting physical pictorial cues.

For example, correct occlusion management between objects in the real environment and virtual geometry (Figure 5.5, Right) requires a more detailed model of the real environment. Ideally, it is accurate enough to create a pixel-accurate occluder mask or depth map of the scene. Likewise, the rendering of realistic shadows and lighting effects between the physical objects and virtual objects requires a more detailed model. The sparse data provided by GIS databases often lacks these details. As shown in Figure 5.4, the virtual representation of the house is slightly misplaced and does not have the right height. This leads to a wrong perception of the object alignment.

If an accurate registration of the physical world and the GIS data is available, as it is the case for an high-precision AR system (Section 3.1), we are able to complete the sparse information with information coming from the video images. In this section we describe such a method that combines sparse GIS data with information derived from real world imagery in order to infer a dense depth map. This allows to derive more detailed information that can be used to provide additional visual cues in the final AR rendering.

For this purpose, we transform sparse 3D model data into 2D shape priors in the current camera frame guiding an appearance-based image segmentation (Section 5.3.1). We describe different image segmentation methods and their integration with the prior data to compute a segmentation of the real world objects corresponding to the model prior (Section 5.3.2). The result of our approach is a dense depth map representing real world objects (Section 5.3.3 and Figure 5.5, (Middle)). Finally, we use the resulting dense depth map for supporting the scene comprehension, such as for occlusion management or shadow rendering.

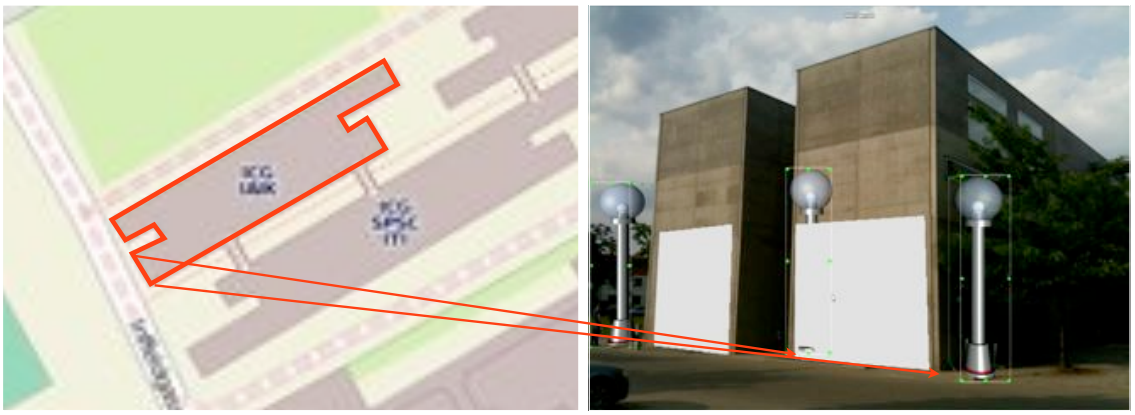


Figure 5.4: Problems in occlusion management resulting from inaccurate GIS data. Left) GIS data of building. Right) The sparse geometries given by the GIS do not fully cover the wall of the buildings. Thus, occlusion cues can only be created incompletely.

5.2 Background

Creating 3D model information from images is a traditional research area in computer vision. Many techniques require specific input such as multiple images from meaningful viewpoints [51] or specialized hardware such as stereo cameras or depth sensors to create dense depth information. Consequently, these techniques are often not suitable for outdoor AR systems. Therefore, we aim to determine depth maps from single images without specialized hardware but by reusing other information.

There is some previous work that focuses on determining depth information from single images. For instance, Hoiem et al. [53] proposed a method to automatically create photo pop-up models. The result of their method is a billboard model, which is created by mapping discriminative labels to parts of the image and applying different cutting and folding techniques to them. Since this method is limited to work only for selected scenes without foreground objects, Ventura et al. proposed a method for supporting the cutting and folding by an interactive labeling of foreground objects [124]. Another technique to create pop-up objects was introduced by Wither et al. [129]. For their approach the authors used depth samples produced by a laser range finder as input for a foreground-background segmentation. Using the segmentation and the known depth sample, they are able to propagate depth to each image pixel. The idea of our method is to create pop-up models automatically, without being limited to selected scenes, user input or laser range sensors.

In 3D reconstruction, several research groups already used external input to improve the results of *Structure from Motion (SfM)* approaches, for instance 2D outlines drawn by the user [118] or prior knowledge about the architecture [26]. Grzeszczuk et al. replaced the interactive input with information from GIS databases allowing for a fully automatic computation of compact architectural 3D models [46]. However, in outdoor AR users are usually observing the scene from one location performing only rotational movements and afterwards moving on to the next position; moving from one location to another location while using the *Augmented Reality (AR)* view is rather rare [45]. Thus, it is not possible

Figure 5.5: Combining sparse information with image coherence. Left) Sparse GIS model data overlaid over a registered view. Middle) Segmentation of the registered view derived from the sparse model. Right) Correct occlusion with virtual geometry using the derived depth map.

to rely on large image sets from different viewpoints as input for a SfM approach as SfM requires translational movement. But Grzeszczuk's work motivated us to use GIS data as additional input for a depth from single image approach.

The basic idea of our method is to combine sparse depth information from a GIS database with a single camera image to map depth information to selected parts of the image. In order to find the corresponding pixels in the registered camera view, we compute a segmentation on the video image. Typically, segmentation methods work with user input that provides sample information about foreground and background [107]. In our approach, we replaced the user input with input from the GIS database. This allows for an automatic computation of the segmentation.

The results of such a dense depth map estimation can be used to achieve a seamless integration of virtual content into physical scenes, for instance by using them as input for occlusion culling or for shadow rendering.

5.3 Combining GIS Database Information and Image Coherence

Our method uses a combination of GIS information and a geo-registered image given by an AR system to segment the input image (see Figure 5.6 for an overview). We start (1) with generating sparse prior 3D models that are projected into the camera image. The projection seeds (2) an automatic segmentation process with shape and appearance priors. The segmentation process (3) returns a 2D image area that corresponds to the 3D object and is taken as the true object outline in the image. This object outline is back-projected onto a plane representing the depth of the sparse 3D feature. From multiple such features and assumptions on the ground plane and the scene depth, (4) we create a coarse pop-up like depth map of the input image (Figure 5.6, Right).

A building, for example, is usually only represented as the 2D projection of the walls onto a ground plane in the GIS. This means, we only have information about the location and extend of the intersection between walls and ground, but not the actual height or overall 3D shape of the building. Assuming a certain minimal building height, we can create an impostor shape that represents the building up to that height (e.g., 2m). This shape will not correspond to the true building outline in the input image, but is enough

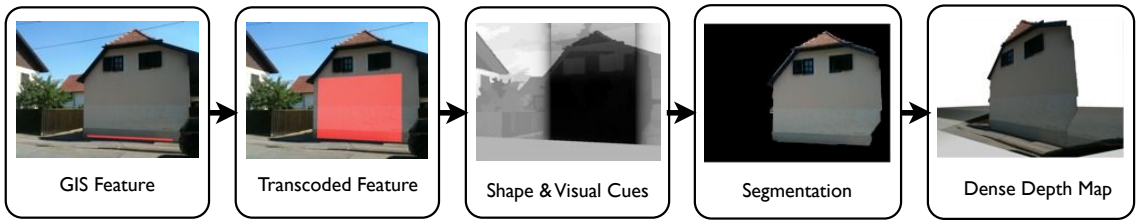


Figure 5.6: Overview of dense depth map creation. The AR system provides a registered camera image, which is combined with GIS model data to derive visual and shape cues and guide a 2D segmentation process. From the segments and the corresponding 3D location a dense 3D depth map is formed.

to provide both shape and appearance priors. Its shape and location already provide a strong indication where in the image the segmentation should operate.

In general, segmentation separates the image into foreground and background based on visual similarity, i.e., similar colors in the image. Through sampling the pixel colors in the re-projected shape, we create an initial color model for the foreground object. The re-projected shape serves as input similar to the user input in interactive segmentation methods as it provides foreground and background color models. The result of the segmentation labels image pixels as either foreground, or background. If the segmentation was successful, the foreground region represent the object of interest. Together with the depth information, the foreground pixels are used to create a flat model of the object of interest which can be textured by the camera image or used to render a depth map for occlusion effects.

5.3.1 Cues from Sparse Models

We start our method with extracting shape cues and appearance cues from the GIS. The sparse 3D model obtained from a typical GIS database will provide the following inputs:

- A supporting shape onto which the segmentation is projected
- Seed pixel locations that form a foreground color model for the segmentation
- Additional energy terms to steer the segmentation

Support Shape GIS databases contain only sparse and abstract geometric representations of physical world objects. Therefore, we need to use additional information about the represented object and the application to create an approximate support shape.

Despite the sparseness of the stored geometric features, they usually contain a lot of semantic information, such as a label representing a type (e.g., wall, street curb, tree, fire hydrant) and specific parameters, such as the diameter of a pipe or the type of tree. We use this additional semantic information to arrive at an approximate geometric shape.

For naïve AR overlays, GIS database features are usually transcoded using the georeferenced data and their semantic labels to assign some predefined 3D geometry and

parameterize it with the available position data as shown by Schall et al. in [108]. Additional parameters that are required for the visualization are configured in advance by expert users. For instance, all features with the semantic label "tree" are transcoded to a 3D cylinder and positioned at the corresponding position. The radius and height of the cylinder are fixed parameters configured for the particular transcoding operation.

We employ a similar transcoding step, to create a support shape from a GIS feature. In contrast to pure visualizations, we are only interested in obtaining an estimate for the shape and location of the object in the image. For example, a building wall is defined as a horizontal line in the GIS. Because it is a wall, we extrude it vertically to create a vertical support plane for it.

Due to the missing information in the database, some dimensions are less constrained then others. For the building wall in the example given above, the ground edge of the wall and the extension in the horizontal direction are well defined, while the height is essentially unknown. Therefore, we create a geometry that follows conservative assumptions and should only cover an area in the image that we can be certain belongs to the physical object.

Visual Cues The 2D location information provided by the support shape alone is not enough input for a good segmentation. Good visual cues are essential to obtain an accurate segmentation of the image. Appearance information cannot be inferred from the database, because semantic labels give no further information about the appearance of a feature. We can compute appearance information by projecting the transcoded geometry into the image. This allows for sampling pixels in the coverage area to extract a foreground appearance model of the object's segment. Such an appearance model can include gray values, color values in different color spaces, texture information, or binary patterns as described by Santner et al. [107].

In order to avoid the influence of noise and inaccurate shape priors we apply an over-segmentation to derive visual cues. For that purpose, we use the superpixel algorithm described by Felzenszwalb et al. [32], as it provides an over-segmentation based on a perceptually natural grouping and it is fast to compute. Based on the oversegmenation, we compute image statistics for each individual superpixels. We use the average L*a*b-value and the texturedness of each superpixel to form the appearance model.

Spatial Cues Besides an appearance model and a seed location from the support shape, the location and area potentially covered by a feature in the image should also be taken into account. We assume that objects have a compact and simple-connected shape and are not spread over multiple disconnected areas in the image.

To model this assumption, we introduce a spatial energy term in the segmentation that only selects pixels as foreground, if the energy for their location is low. The simplest of such terms is the indicator function for the projection of the support shape itself. However, a sharp boundary would defeat the purpose of segmenting the object from the image, therefore we create a fuzzy version of the projected shape by computing a distance transform of the projected shape (see Figure 5.7, Left). This fuzzification ensures that the segmented boundary will take the appearance model into account and not simply coincide with the reprojected shape.

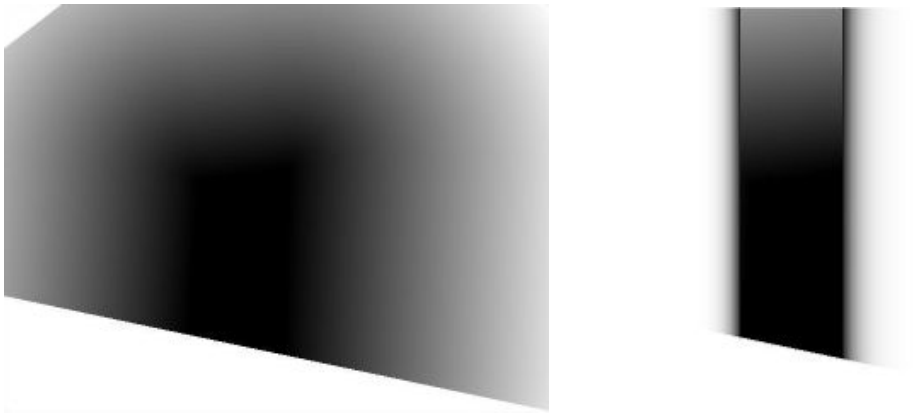


Figure 5.7: Distance-transform of shape cue. Left) Isotropic distance transform of shape cue. Right) Anisotropic distance transform of shape cue.

Uncertainty The shape and location of a feature is often not accurately known, therefore we have to deal with shape and location uncertainties. We model this uncertainty through blurring the spatial energy term accordingly. For example, tracking inaccuracies might lead to wrong re-projection of the feature shifting the object boundary arbitrarily within a certain radius. To model this effect, we apply an isotropic distance function with radius σ representing the tracking error in image coordinates.

Also the shape of a feature is often not fully known from the GIS information alone. For example, walls and buildings have unknown heights, even though the baseline and horizontal extend is well defined by the 2D outline. Hence, we assume a reasonable minimal height (e.g., 2m) and create extrusions of the foot print up to that height. Additionally, we perform a distance transform only in the unknown direction - up in this case. This creates a sharp spatial energy term in the known directions, but a soft energy in the unknown directions and results in an anisotropic distance transform of the shape cue (Figure 5.7, Right). Overall, this makes the segmentation robust against these unknown shape and location parameters.

5.3.2 Segmentation

Based on the extracted cues, we continue with the segmentation of the input image into foreground and background corresponding to the data given by GIS. For the segmentation, we experimented with two different methods using the same input from the projected, transcoded GIS data and computing the same output, a segmentation labeling every pixel as part of the object of interest or not. The two methods offer a trade-off between complexity and quality of the segmentation.

The input consists of the visual and spatial cues as described in Section 5.3.1. Each set of cues defines a cost function for every pixel in the image describing the likelihood that this pixel is part of the object of interest.

The re-projected and transformed support shape defines a likelihood function f_{shape}.

Pixels that are located far from the input shape get a lower probability than pixel located close to the input shape or that are located inside the input shape. Uncertainty is encoded in the distance transformed input as well (Section 5.3.1).

Visual cues are represented as the mean feature vector $V(S)$ averaged over all pixels in the sample set S. Based on the selected segmentation algorithm these sets of pixels are defined differently. The likelihood function $f_{vis}(p)$ of a pixel p is given as the squared euclidean distance of the pixel's feature vector to the mean feature vector

$$f_{vis}(p) = \|V(p) - V(S)\|^2. \tag{5.2}$$

Greedy Algorithm The first segmentation method is an approach based on a greedy algorithm that calculates the final segmentation by operating on the superpixel representation. This method uses all image segments that are covered by the support shape to seed a region growing algorithm and labels them as being part of the object of interest. The region growing enforces connectivity to obtain a single object. The set of segments that identified as being part of the object of interest is called O.

For a segment s that is adjacent to one or more segments $n \in O$, we define the likelihood function $f_{vis}(s)$ for the visual cues as the minimal distance between the feature vector of the segment and all neighboring foreground segments

$$f_{vis}(s) = \min_{n \in O} \|V(s) - V(n)\|^2 \text{ where } n \text{ neighbors } s. \tag{5.3}$$

The labeling $L(s)$ is given by summing up the likelihoods of the distance transform $f_{shape}(s)$ averaged over the segment and the visual similarity $f_{vis}(s)$ as defined in (Equation 5.3) to a single cost function $C(s)$ for the segment s

$$C(s) = f_{shape}(s) + f_{vis}(s), \tag{5.4}$$

and thresholding the cost function for determining the label:

$$L(s) = \begin{cases} 0, & \text{if } C(s) \geqslant T \\ 1, & \text{if } C(s) < T. \end{cases} \tag{5.5}$$

This decision is iterated until no new segments can be labeled as foreground. The result of the greedy segmentation method is a binary labeling that describes whenever an image segment belongs to the object of interest or not (Figure 5.8, Left). The disadvantage of this method is that it is not searching for an optimal solution.

Total Variation Approach The second segmentation is based on minimizing a variational image segmentation model [122]. Santner et al. successfully applied Total Variation for interactive image segmentation [107]. Since their method tends to find accurate object segmentations based on minimal user input, we decided to use a similar approach and replace the interactive input with input from the GIS database. For this purpose, we defined a per-pixel data term similar to the cost function in the first method.

The variational image segmentation minimizes the following energy function $E(u)$ over a continuous labeling function $u(x, y)$ that maps every pixel to the interval $[0, 1]$ indicating

Figure 5.8: Segmentation results. Left) Segmentation based on the Greedy algorithm, some parts of the roof are missing. Right) Segmentation based on the Total Variation approach.

if it is foreground (as 0) or background (as 1)

$$\min_{u\in[0,1]} E(u) = \int_\Omega |\nabla u| d\Omega + \int_\Omega |u - f| d\Omega. \tag{5.6}$$

The first term is a regularization term minimizing the length of the border between the segments. The second term is the data term given by the function $f(x, y)$ indicating how much a pixel belongs to the foreground (0) or background (1).

We define the data-term $f(x, y)$ based on the shape cues and visual cues presented in Section 5.3.1:

$$f(x, y) = \alpha f_{shape}(x, y) + (1 - \alpha) f_{vis}(x, y) \tag{5.7}$$

$$= \alpha f_{shape}(x, y) + (1 - \alpha) \|V((x, y)) - V(S)\|^2. \tag{5.8}$$

$V(S)$ is averaged over the set of all segments covered by the initial shape cues. The function $f(x, y)$ is further normalized by its maximum value to serve as a data term in (5.6). The parameter α weights the influence between shape or visual cues. The output of the Total Variation approach is a smooth labeling function $u(x, y)$ that is thresholded at 0.5 to obtain a binary labeling. The final binary labeling provides a pixel accurate representation of the foreground object (Figure 5.8, Right).

5.3.3 Geometry Computation

So far our method computed a 2D segmentation describing the parts of the image belonging to the sparse model given by the GIS database. Before we can use the result as input for AR visualization techniques, such as occlusion culling or shadow rendering, we have to compute a 3D representation.

For this purpose, we have to assign each 2D pixel of the segmentation 3D information from the georeferenced data. First, we extract the 2D outline of the segmentation by

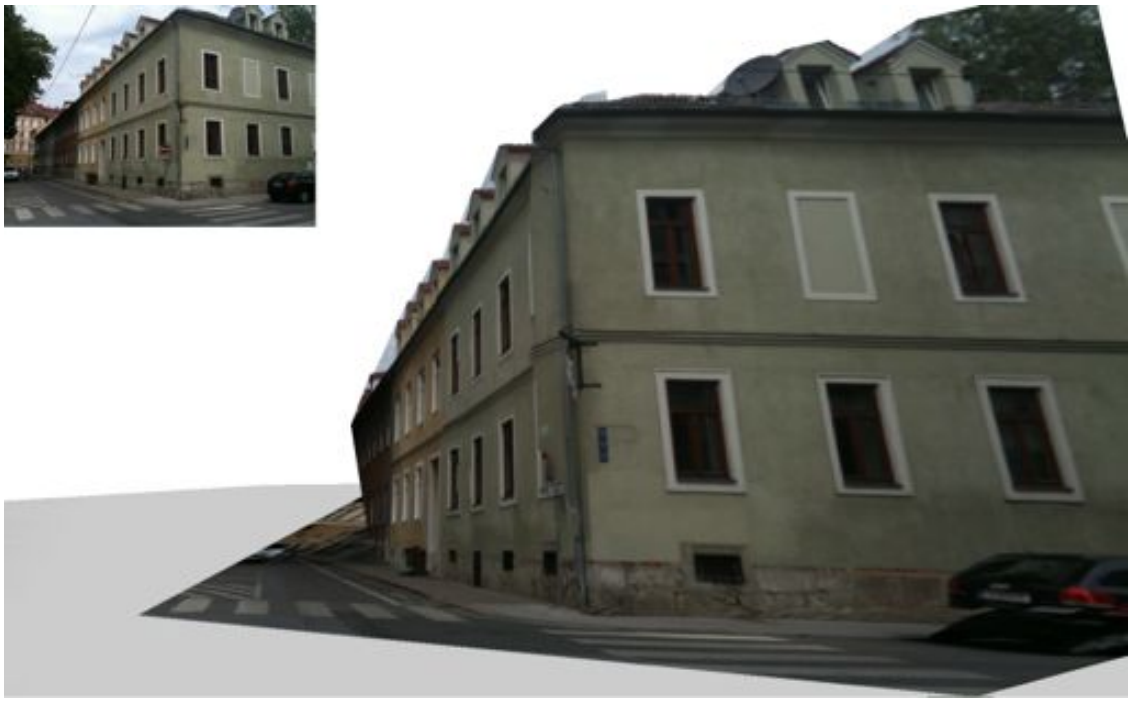


Figure 5.9: Extracted pop-up model created with our method for the image shown in the inset.

tracing the points on the border in either clockwise or counterclockwise order. The result is a list of ordered pixels forming the outline. Next, we set up a plane containing the support shape and aligning the plane in vertical direction to the world ground plane in order to obtain an upright representation similar to a billboard object. The individual pixels of the segmentation outline are back-projected into 3D rays through the inverse camera calibration matrix and intersected with the plane. This intersection establishes the 3D coordinates for the pixels. Finally all 3D coordinates are combined to create a 3D face set. To obtain a full dense representation, we also add a representation of the ground, either derived from a digital terrain model or by approximating the average height with a single plane.

Furthermore, we can create a textured model for visualization purpose. Projective texture mapping allows to correctly projecting the camera image back onto the 3D geometry (see Figure 5.9 and Figure 5.10, Right in an AR overlay). As input for the projective texturing mapping, we set the texture matrix up to use the same camera model as the calibrated real camera.

5.3.4 Results

We analyzed the results that we achieve with of our approach by 1) computing the accuracy of the results under different conditions and 2) comparing the accuracy of the Total Variation approach with the method based on the greedy algorithm.

Figure 5.10: Extracted pop-up model in an AR overlay. Left) Outline of a pop-up model overlaid onto an urban scene. Right) Textured pop-up model overlaid onto the camera image.

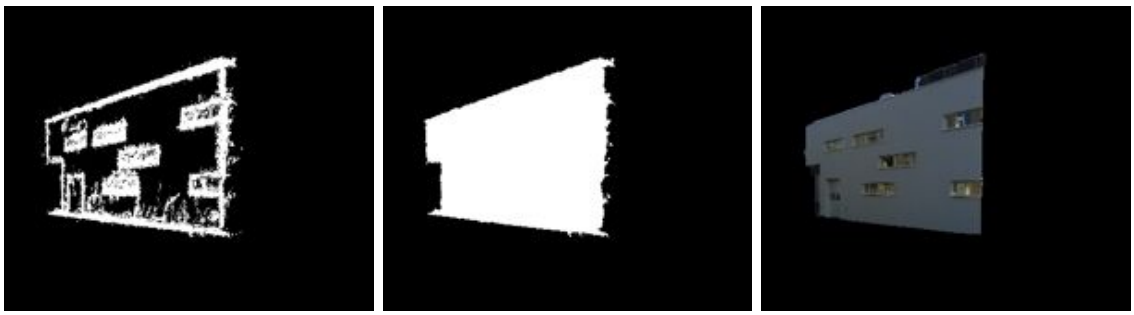


Figure 5.11: Computation of segmentation error. Left) Reprojected 3D points of the point cloud. Middle) Computed filled polygon of the reprojected points. Right) Extracted segmentation.

Accuracy We tested the accuracy of the dense depth estimation against a 3D ground truth model. As ground truth geometry, we used a dense 3D reconstruction created from a set of aerial images acquired with an octo-copter as described in Section 3.2.3. As input to the dense depth estimation we use a georeferenced 2D CAD plan of the building provided by a construction company and georeferenced images of the building.

For testing the accuracy of the segmentation method, we are only interested in the set of ground truth 3D points belonging to our object of interest. We select these points manually from the ground truth point cloud and compare them to the dense depth map computed by the our approach (see Figure 5.11). Firstly, we compute for each point of the object, if its projection on the plane of the estimated pop-up model is inside the polygon describing the outline of the extracted object or not. Secondly, we calculate an accuracy value for the *inlier* estimation by dividing the amount of inliers by the number of points. This measurement can only provide information if the extracted object is big enough. It does not tell if the outline of the extracted object has enough similarity to the reference object. For this purpose, we compute a second value, the *segmentation error*. The segmentation error is computed by calculating the difference in pixel between the

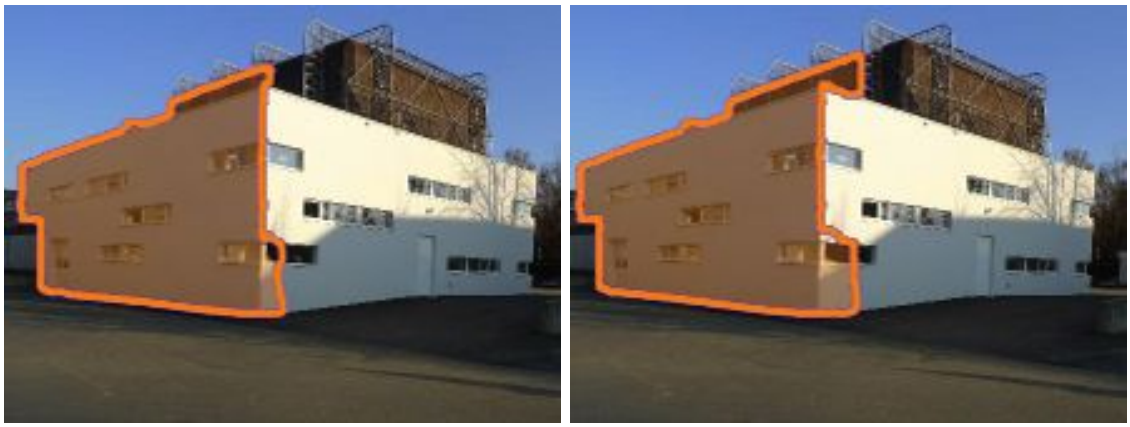


Figure 5.12: Segmentation results for selected offsets. Left) Segmentation result for 0.5m offset. Right) The segmentation with an 1m offset shows a larger error but still a visually acceptable result.

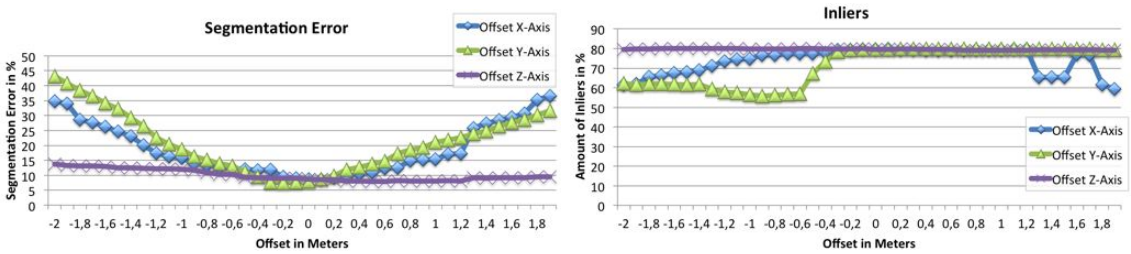


Figure 5.13: Accuracy measurements for different simulated GPS offsets. Left) The segmentation error decreases with decreasing offsets. Right) The amount of inliers is increasing with decreasing offsets.

filled polygon given by the segmentation and the polygon created by re-projecting the 3D points (Figure 5.11, Left) and computing the outline of the resulting 2D points (Figure 5.11, Middle).

These measurements further provide information about the registration accuracy that is required by our approach to produce reliable results. To determine this accuracy value, we created an artificial location offset of the input data and calculated the inliers and the segmentation error in relation to the offset. The results show that with increasing offset errors the accuracy of the result is decreasing (Figure 5.13). We can also show that with an offset of 0.5 m for all directions, segmentation errors below 15% can be reached. Even for an offset of 1m the segmentation errors are below 20%. As shown in Figure 5.12 the segmentation results are still visually acceptable. This level of accuracy can be easily obtained by using a L1/L2 RTK receiver as integrated in our outdoor AR setup (Section 3.3, [110]).

Comparison of Segmentation Methods So far, we determined the accuracy that can be achieved by using the Total Variation approach. Furthermore, we used the accuracy measurements to analyze the difference between the greedy approach and the Total

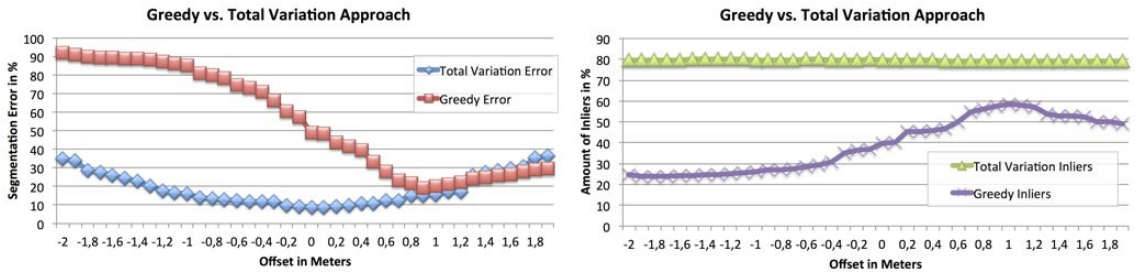


Figure 5.14: Accuracy measurements both segmentation methods. Left) The segmentation error for the Total Variation method is lower than for the Greedy method. Right) The amount of inliers is higher for the Total Variation method.

Figure 5.15: Occlusion management using the dense depth map. Left) Without depth information no occlusion cues can be provided. Middle) Using sparse geometries for occlusion culling can only create partially coorrect occlusion and may result in an incoherent visualization. Right) Using dense depth maps for occlusion culling provides important depth cues.

Variation approach. We used the same methods to determine the segmentation error as well as the amount of inliers as before. As shown in Figure 5.14, the segmentation error for the greedy-algorithm-based approach is much higher than for the Total Variation approach. Also the amount of inliers is much lower than for the Total Variation approach. This shows that the Total Variation method creates more accurate than for the greedy approach.

5.4 Generating Physical Pictorial Cues

The main idea of creating dense depth maps from sparse GIS features was to improve the scene comprehension in AR (Figure 5.15). In this section, we will show how to create physical pictorial cues based on the dense depth maps that we can compute with our method.

Occlusion Cues In Figure 5.15 (Left) we show a naïve AR overlay that lacks occlusion cues. A mix of virtual occluded and virtual non-occluded information is superimposed onto the camera image without culling occluded information. This makes it hard to understand the spatial layout of real and virtual objects. By using the extracted dense depth maps

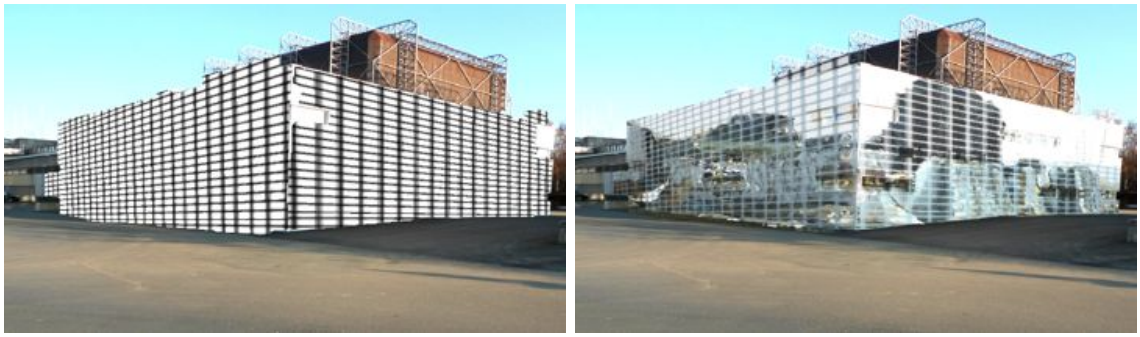


Figure 5.16: Occlusion management using importance maps. Left) The importance map consisting of a grid is mapped to the dense representation of the physical wall. Right) X-Ray visualization using the pop-up model combined with a checkered importance mask.

to decide whether virtual content is occluded or visible, we can achieve a more convincing scene integration (Figure 5.15, Right). In particular, we provide different methods to provide occlusion cues such as

- rendering occluding parts completely opaque,
- rendering occluding parts transparent,
- using an importance mask to render the occluder partially opaque.

If the occluding object only occludes a part of the virtual object or the physical object is regarded being highly important, it may make sense to render the complete object opaque as shown in Figure 5.15. This improves the perception of the virtual lamp objects to be located behind the physical building.

Dense depth maps allow for applying other occlusion management techniques, such as the importance maps presented by Mendez and Schmalstieg [90]. Their technique requires a phantom representation of the scene, which is usually not available for outdoor applications. For this purpose, we can use the dense depth maps of each object as phantom object (Figure 5.16, Right) and map an importance map texture onto it.

Shadow Rendering The dense depth maps provide also information where virtual objects would create shadows on the physical world objects and vice versa. This information is usually not available in naïve AR overlays. Since shadows can provide additional depth cues, the presentation of shadows can also help to increase the scene comprehension of AR renderings. In order to create a shadow of a virtual object onto a physical object, an impostor of the physical object is required. Another requirement for correct shadow computation is that the physical light sources are known.

Shadow rendering is often part of scenegraph APIs. Scenegraphs, such as Coin3D, often provide shadow rendering in a specific environment. For instance, in Coin3D the class *SoShadowGroup* provides real-time shadow casting. All objects within this group

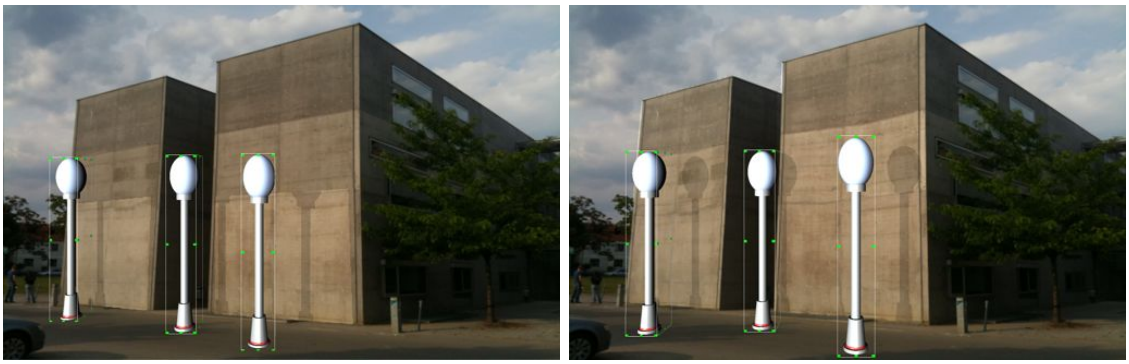


Figure 5.17: Using the dense depth maps for shadow rendering. Left) Sparse map used for shadow rendering can provide shadows only partially. Right) Dense depth maps allow creating convincing shadows over the complete building.

Figure 5.18: Using the dense depth maps in other AR applications. Left) Annotating a building. The dense depth map allows to make annotations spatially consistent. Middle) Labeling. The dense depth maps helps to filter visible labels (marked red) from invisible (marked black). Right) Surveying. The user can interactively survey parts of the building by using the dense depth map as interaction surface.

cast shadows depending on the selected shadow style. For casting shadows on the physical world objects, we simply render a virtual pop-up model on top of the video image.

In Figure 5.17 (Right), we show an example where virtual lamps cast shadows onto the impostor representation of the physical wall of the building.

5.5 Other Applications

There are several other applications that can benefit from dense depth maps, such as the in-place surveying and annotation (Figure 5.18, Left and Right) in outdoor AR, where usually no accurate dense depth model is available. In the following, we will discuss some of these application scenarios.

Annotations Annotations of the physical world can benefit from using our dense depth maps. Methods for annotating the physical environment often work in the 2D image space. Thus, they only allow one to view the annotations from a specified location [80]. Other methods apply additional input devices such as laser range finders [129] to determine the

depth of an object that should be annotated. Wither et al. also proposed an approach using additional aerial views of the scene [130] to determine the 3D position of an annotation. The dense depth maps created by our approach allow creating geo-referenced annotations without additional measurement devices and without interacting in aerial views of the scene (Figure 5.18, Left).

Another problem of label placement, in particular, for AR browsers, is that often all available labels at the user's position are shown since no depth information is available. This results in cluttered views, where the user may have difficulties associating labels with the environment. By using dense depth maps for occlusion culling on the labels, we are able to display only relevant, visible labels for the current view (Figure 5.18, (Middle) shown in red).

Surveying application Surveying is an important task for a lot of architectural and construction site applications. Usually, people have to make use of extensive equipment. Often, the goal is to measure dimensions of objects in the user's proximity. For instance, to compare the size of newly built objects with the as-planned size. Dense depth maps allow users to survey physical world objects in their field of view by using AR. In order to survey the width and heights of windows, the user can select 3D points directly on the facade of the building by clicking in the AR view. Based on the mouse coordinates the surveying application computes the intersection point between the ray starting at the mouse coordinate and the dense depth map of the facade. Multiple surveying points create a set of line segments. The application displays the length of each line segment using text labels (Figure 5.18, Right).

5.6 User Survey

In order to learn more about the effect of using our dense depth map approach for creating correct occlusions in urban scenarios, we carried out an informal survey where we ask 18 people (8 females, 10 males) to fill out an online questionnaire about their perception of a set of AR scenes. From the participants 61% had already experience with any kind of AR system, and 39% did not. Within the survey, we showed them different urban scenes containing different virtual objects. For each scene, we showed them a visualization 1) using no occlusion culling, 2) using an occlusion culling based on sparse GIS data (Figure 5.19, Top Row), and 3) using an occlusion culling based on our dense depth maps (Figure 5.19, Bottom Row). After viewing at each scene, we asked the participants to rate if they understood the spatial arrangement of virtual and physical objects. Their rating was captured using a 6-Point Likert scale (disagree - agree).

The graphs in Figure 5.20 show the distribution of answers according to the applied occlusion culling technique for three of the scenes. It seems that people had the tendency to agree that they understand the arrangement using the dense-depth-based occlusion culling. For scene "A", 66.7%, for scene "B" 55,6% and for scene "C" 38.9% of the participants gave the highest score indicating that they understood the arrangement. In total, only 5.6% of the people gave a rating that indicates that they did not understand the arrangement in all scenes. The lowest rating for the dense depth maps was a 3.0 (5.6%).

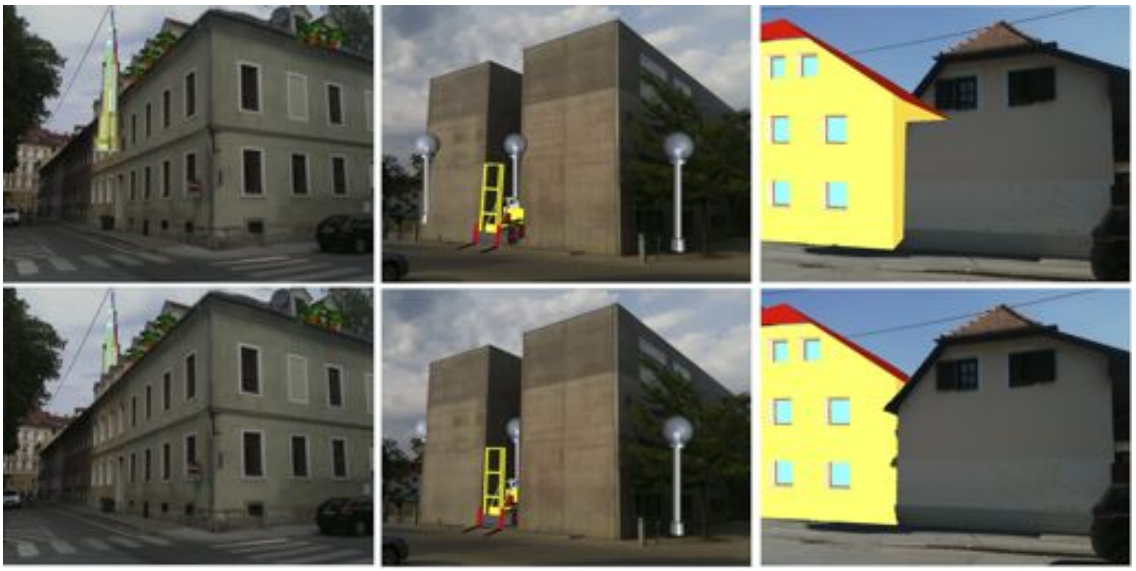


Figure 5.19: Test scenes for the survey. Left) Scene "A": Street with virtual skyscraper. Middle) Scene "B": Buildings with virtual lamps. Right) Scene "C": Street with virtual building. The top row shows the visualization using the sparse GIS data as input for occlusion culling. The bottom row shows the result using the dense depth map estimation.

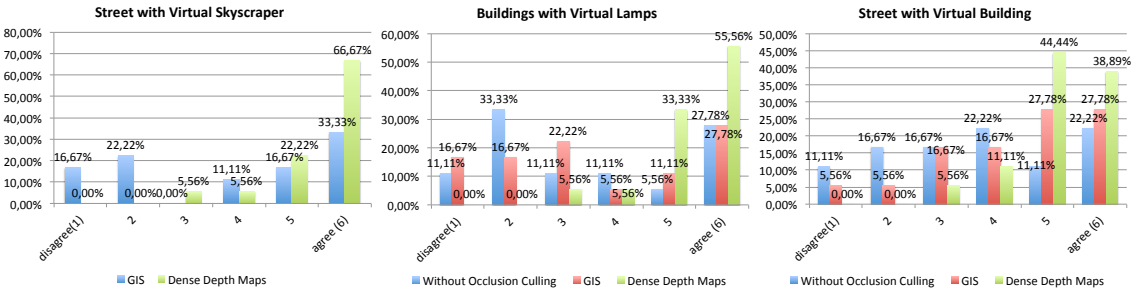


Figure 5.20: Results for different test scenes for the question "I understood the arrangement of virtual and physical objects in the presented image".

In contrast, the population of the participants seemed to be undecided if they understood the arrangement in scenes that applied no occlusion culling. For scene "B" 55.5%, and for scene "C" 44.5% of the participants gave ratings indicating that they did not understood the arrangement of objects. For scene "A" there are no measurements, since the visual result is the same as for the visualization using the sparse GIS data for occlusion culling. For the occlusion culling based on sparse GIS data, the ratings were a similar or a bit better than for the visualization without occlusion culling. For scene "A" 38.9%, for scene "B" 55.5%, and for scene "C" 27.8% of the participant gave the visualization a rating indicating that they did not understood the arrangement (1.0-3.0). This is still a big amount of people that had problems in understanding the arrangement compared to the 5.6% that had comprehension problems in the scenes that used the dense depth maps. In conclusion, this informal survey gives a first feedback that the dense depth could be useful

for improving the comprehension of spatial arrangement for AR visualization in urban scenes.

5.7 Summary

In this chapter, we discussed different methods of using sparse representations from GIS databases to provide physical pictorial cues for AR. Firstly, we applied sparse geometries as direct cues. The direct usage of the GIS data is often subject to inaccuracies due to the sparseness of data. Therefore, we introduced a method that allows combining a sparse representation from GIS databases with camera imagery of the physical environment. By initializing a segmentation algorithm with projections of the sparse data, we can obtain more accurate representations of the 3D objects. This combination allows us to improve sparse features and offers new possibilities to maintain physical pictorial cues.

While the results of the dense depth maps can be used in AR in real-time, the current implementation of the calculation does not operate in real-time as both the super-pixel segmentation and the final variational method do not run at interactive frame rates. We aim to overcome this limitation through incremental computation of the super-pixel segmentation [120] and relying on frame-to-frame coherence to reuse results from the last frame, as well as application of the variational optimization to the superpixels directly. This would reduce the complexity of the problem by several orders of magnitude.

Chapter 6

Virtual Pictorial Cues

Contents

6.1 Introduction

In the previous chapters, we discussed how to derive physical cues for achieving a seamless scene integration of virtual information in the physical environment. A seamless scene integration can be seen as a foundation for comprehensible visualization. It provide us with ordinal depth cues, such as occlusion and shadows, and avoid misinterpretations of spatial relationships. Nevertheless, there is often the lack of absolute depth measurements in AR compositions. This is in particular the case for the visualization of objects that have no connection to the ground plane [23], such as X-Ray visualization or for the visualization of floating objects representing for instance waypoints of aerial vehicles. Our visual system has problems in interpreting relative or absolute depth information since several pictorial cues can not be used (see Chapter 2).

This is an important challenge, since several industrial outdoor applications, such as information query for digital assets or flight management of aerial vehicles, require absolute depth information. The aim of this chapter is to integrate additional graphical hints (virtual cues) into the visualization, in order to address the issue of missing absolute depth cues in *Augmented Reality (AR)* scenes with occluded or floating objects. These additional graphical hints are supposed to support users in deriving absolute depth measurements for these kind of AR scenes.

We differentiate between two kind of cues, user-centric and data-centric. User-centric cues support situations where the distance between the user an the virtual data is important. This is the case in tasks where the user has to judge the depth of objects in relationship to its own position, for instance to start digging for a subsurface pipe. If

Figure 6.1: Physical depth cues vs. user-centric virtual visual cue. Left) Physical occlusion cues only provide ordinal depth information. Right) A virtual magic lens provides relative depth information.

the spatial relationship between data and the physical world is in focus, the visualization technique of choice are data-centric virtual cues.

6.2 User-centric Virtual Cues

The main aim of user-centric virtual cues is to visualize the direct relationship to the user. These cues provides additional information about the relationship between the user and virtual objects of interest. In the following, we will describe two examples for user-centric virtual cues. One that provides absolute depth information for subsurface objects and one that supports the depth estimation for floating objects.

6.2.1 Virtual Cues for Subsurface Objects

Physical cues extracted with the methods described in Chapter 4 and 5 provide a coherent integration of virtual subsurface objects into the physical world (Figure 6.1, Left). However, estimating the depth of these subsurface objects relative to the surface is still complicated, since the available physical pictorial cues are not sufficient. Virtual magic lenses related to the user's pose can provide additional depth hints that address the problem of insufficient pictorial cues.

Magic lenses are usually used as Focus&Context tools, having the main goal of separating the visualization into a context and a focus area. But they can also be applied as virtual pictorial cues to provide additional absolute depth measurements since they display a direct connection between the virtual subsurface object and the surface. This is further supported by combining the magic lenses with a virtual scale visualization. The user-centric magic lens provides users with a view into the ground in front of them. In our case, we use a magic lens with the visual appearance of an excavation. This provides a more realistic visualization of having a view into the ground on a construction site (Figure 6.1, Right).

Figure 6.2: Physical cues compared to user-centric virtual cues for MAV nagivation. Left) Occlusion cues for a planned waypoint of a MAV. A critical position is highlighted in red. Right) User-centric virtual cues visualize the distance between the user and the waypoint. A virtual scale texture shows the distance in meters.

6.2.2 Virtual Cues for Floating Objects

Another application that benefits from integrating user centric-virtual cues, is AR flight management for Micro Aerial Vehicles. As mentioned in Section 2.4, the flight management of aerial vehicles still requires a user in the loop that supervises flight sessions. Far field navigation of an aerial vehicle can strongly benefit from additional graphical aids that an AR visualization can provide. Currently, users that supervise a flight session face a set of problems while monitoring a flying *Micro Aerial Vehicle (MAV)* and simultaneously making use of a 2D map interface for flight path inspection. These problems include the high mental effort required for mapping 2D map positions to the current physical environment and the complicated estimation of the distance and height of the aerial vehicle. AR visualization can provide flight-relevant information overlaid to the current environment. Such an overlay helps to address problems in understanding the spatial relationship of waypoints, current positions and already captured positions within the physical world. Physical occlusion cues can support the user in understanding if planned positions are located behind or inside obstacles. For instance, Figure 6.2 (Left) demonstrated a highlighted visualization of a waypoint inside a building.

However, for the managing flight session often it is required to understand the exact depth between the MAV and the user. This information is not provided by the occlusion highlighting. In order to allow for a relative depth estimation in relationship to the user's position, we propose a set of additional graphical hints that help to improve the depth perception of floating objects such as waypoints. For this purpose, in addition to a virtual representation of a waypoint, we render a rectangular geometry connecting the user's ground position with the waypoint. The rectangular object is thereby starting at the user's ground position, connecting with a point on the groundplane, which is created when projecting the waypoint on the ground plane, and finishing in the waypoint itself. By texturing the additional geometry with a metric scale, the depth estimation can be further supported (Figure 6.2, Right).

Figure 6.3: Interactive planning and surveying with mobile AR. Left) Creating and surveying a new cable with the setup. Middle) Manipulating the cable. Right) Comprehensible visualization showing an excavation along the cable.

6.3 Data-centric Virtual Cues

In contrast to user-centric virtual cues, data-centric virtual cues provide additional information about the relationship between virtual data and the physical world. For instance, an virtual excavation along a pipe allows one to understand the subsurface location of this pipe in reference to a street (Figure 6.3, Right). Such a virtual excavation supports the user not only in understanding that the pipe is located under the ground, but also transfers depth measurements (relative to the world reference system). Instead of requiring the position of the user as the user-centric cues, these cues can be directly derived from large data bases (e.g., GIS databases) since they only depend on the data geometry and a reference to the ground (which is usually provided by a terrain model of the environment). From this information, virtual cues can be derived automatically. Similar to the physical cues from Section 5.1.1, they are merely different representations of the GIS data. It is important to note, that method that automatically create such additional visual representations have a special challenge when it comes to data that is interactively modifiable such as GIS data in surveying or planning applications. The challenge is to maintain data consistency. If the user is manipulating a selected object, its depth cues should be updated automatically.

In the following, we will describe a set of additional depth cues, such as virtual cutaways providing hints about the distance to the ground (Figure 6.4, Bottom Left), reference shadows projecting the pipe outlines to the surface, or connection lines visualizing the connection between the object and the ground (Figure 6.4, Bottom Right). The main goal for all these additional geometries is to establish the relationship between virtual data and the physical ground plane.

Object-aligned cutaways Object-aligned cutaways are cutaways that create the impression of cutting out parts of the occluding object to make the occluded objects visible (Figure 6.4, Bottom Left). These cutouts are visualized as excavations and provide the user with depth cues about the position of the underground object. Additionally, they are supplemented with depth indicators, such as soil textures with metric scales. We can increase the impression of a subsurface location by combining the virtual cutaway geometries with a transparency rendering. For this purpose, we render those parts of the excavation

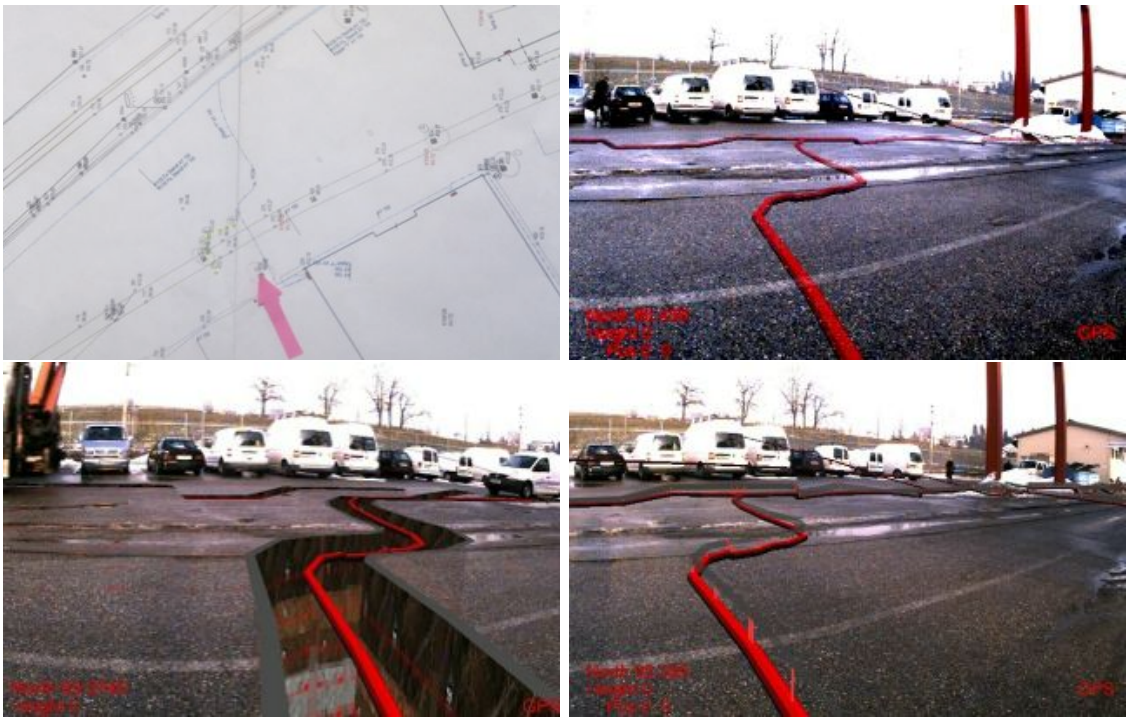


Figure 6.4: Different visualizations of an electricity line feature. Top Left) Paper plan showing the pipe as 2D line. Top Right) A red cylindrical extrusion. Bottom Left) Showing an excavation along the electricity line. Bottom Right) Showing virtual shadows cast on the ground plane.

that would be naturally visible when looking inside an excavation fully opaque. Parts that would be not naturally visible, such as the side walls of the excavation from the outside, we render with an transparent overlay (Compare the opaque visualization of the interior with the transparent rendering for the outsides of the excavation in Figure 6.4, Bottom Left).

Reference shadows For some tasks, it is important to understand the corresponding location on the ground plane. For instance, for digging tasks the user has to find the right position on the ground to start the digging. Reference shadows support these tasks by showing the outlines of the subsurface object on the ground plane. They give the user a cue, where objects are located in relation to the ground. For the digging tasks, the reference shadows in grey, show the corresponding position on the ground (Figure 6.4, Right).

Such reference shadows can not only be used for subsurface visualization, but also for improving the depth perception for floating objects such as waypoints for aerial vehicles. By projecting the waypoint on the ground plane, we can indicate its position in relationship to the surface. This graphical hint appears as a virtual shadow of the waypoint (Figure 6.5, Left).

Figure 6.5: Additional graphical hints for supporting the understanding of the position of the MAV (Asctec Falcon 8 in black). Left) Sphere representation of the waypoint and its shadow projection on the ground. Right) Waypoint visualized with connection line and virtual shadows.

Connection lines As discussed in Chapter 2, the natural depth cue *height in visual field* only provides absolute depth information for objects that are connected to a flat ground plane [23]. Thus, one of the main issues for visualizing subsurface or floating objects is the missing connection between physical ground and the object. To restore the cue's ability of providing absolute measurements, we will include virtual connections between the virtual objects and the ground. For virtual subsurface objects connection lines provide hints about the depth of the occluded objects and their corresponding location on the surface. These lines allow one to interpret the depth of the subsurface object since a direct connection between the ground level and the subsurface object is visualized. For a better visualization effect, we can highlight the connection point between the connection line and the ground. In Figure 6.4 this is demonstrated by combining the connection lines with reference shadows.

Likewise, connection lines can be used to provide absolute depth information for floating objects. The connection lines directly represent the height of floating objects in relationship to the ground. The knowledge about the height supports the user in judging the depth. Additionally, the connection lines can be textured with a scale, to further support the depth perception (Figure 6.5, Right).

Junctions Abstract data often has the same problem as subsurface or floating objects as is not clear how it is connected to the physical world. Even if an abstract data point is located directly on the ground plane, the user can often not interpret this relationship due to missing background information. An example for such a problem is the visualization of survey marks (geospatial bench marks). Such a visualization is for instance useful for accuracy tests or for understanding registration errors. When using a sphere or a cylindrical representation for visualizing this kind of abstract data, it is often not clear if they are located on the ground plane. Consequently, users can often not tell to which location on the ground plane they belong. Junctions can indicate their connection to the

Figure 6.6: Virtual junctions. Left) Using a junction to support the surveying of a reference point. Right) Junction highlighting the reference point on the curbstone corner.

ground plane and support the visualization of this abstract data.

We use such junction visualizations for analyzing the accuracy of our tracking system based on survey marks. In Figure 6.6 (Right) we visualize an existing survey mark from the GIS, and compared it to the physical survey mark on the ground. The junction point highlights the position of the virtual mark on the ground. A printed circular grid with a metric scale allows to compute the difference between survey point from the GIS and the physical survey point (Figure 6.6, Left).

6.4 Implementation

The implementation of these virtual cues depends on the data source and the kind of cue we want to provide. In the following, we will discuss how we can create 1) User-centric depth cues for subsurface information, 2) user-centric depth cues for aerial vision, 3) data-centric depth cues for aerial vision and 4) data-centric cues for subsurface GIS data. While the first three techniques create the virtual cue geometries by directly using the data points of interest and predefined geometries, the last technique requires a more sophisticated data management to maintain data consistency between the data and the virtual cues during manipulations.

6.4.1 Implementing User-centric Cues for Subsurface Information

In order to create user-centric depth cues, we integrate a predefined geometry representing a magic lens into the rendering. For positioning, we use the pose of the user in reference to the world and move the magic lens by subtracting the height of the setup in relation to ground level. The magic lens's geometry consists of three parts, a box textured with a realistic soil texture and a metric scale, a cover geometry that provides information about the top opening, and a frame geometry that provides information where the magic lens is connected to the ground (Figure 6.7). The rendering is done as follows:

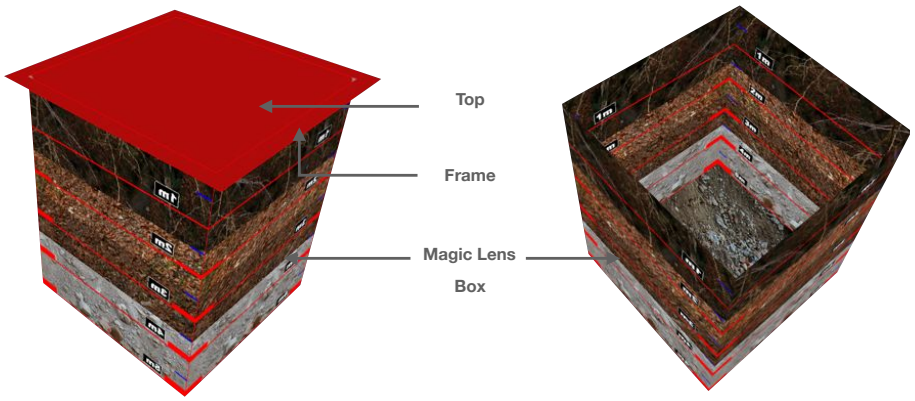


Figure 6.7: Creation of magic lens. Left) Lens box geometry with covered with a top and a frame. Right) Box geometry.

Listing 6.1: Rendering of magic lens.

```
Render video texture
Render magic lens box
Render the top of the magic lens box into the stencil buffer
Render virtual data
Render transparent video texture with enabled stencil test
Render frame geometry
```

This rendering code results in a composition that displays the virtual magic lens geometry containing the virtual data (red pipes in Figure 6.1, Right), and an area outside the magic lens where the virtual pipes are displayed transparent. Additionally, the red frame area around the magic lens occludes the video image to indicate the connection to the ground.

6.4.2 Implementing User-centric Depth Cues for Aerial Vision

Our user-centric virtual depth cues consist of a connecting line from the user position on the ground p_{ug}, the projection of the waypoint on the ground plane p_{wg} and the waypoint location itself p_w. For displaying this connection, we use again a rectangular graphical representation (Figure 6.2, Right). The waypoints are available as georeferenced data in WGS84[1] format. Due to the single precision ability of OpenGL, we have to convert the georeferenced information into a local coordinate system before we can display them. For this purpose, we define a reference point in WGS84 that is used as center point of the local coordinate system. After the conversion all information is available as single precision coordinates and can be used for rendering.

Listing 6.2: Rendering of user-centric cues for aerial vehicles

```
Map georeferenced information to local 3D information
Render a rectangular object from p_ug to p_wg
Render a rectangular object from p_wg to p_w
```

[1] World Geodetic System 1984

6.4.3 Implementing Data-centric Depth Cues for Aerial Vision

In addition to the user-centric depth cues, we introduced depth cues that support the spatial relationship between the information that is relevant for aerial vehicle navigation, such as waypoints, and the physical world. These cues comprise virtual lines connecting the waypoints with the physcial ground and virtual shadows. In order to create connecting lines, we add rectangular objects starting at the location of the waypoint and ending on the ground. For this purpose, we need access to the waypoint information and either information about the ground plane or the height above mean sea level at the location of the waypoint. We can further support the depth estimation by supplying additional height indicators such as a metric scale to the connection lines. These relative measurement are integrated by using a texture with a scale mapped to the connection line geometry.

Another depth cue that we discussed to be helpful for the depth perception of floating objects in AR are shadows. Whereas Wither et al. included an artificial shadow plane to visualize shadows of floating objects [131], we are using the ground plane of the physical environment as shadow plane. For this purpose, we create a flat rectangular object that is located at the position of the waypoint, but at the height of the ground plane. This graphical hint appears as a virtual shadow of the waypoint (Figure 6.5, Left).

Listing 6.3: Rendering of data-centric cues for aerial vehicles

```
Map georeferenced information to local 3D information
If renderconnectionlines
Render a rectangular object from p_wg to p_w
Else if rendershadow
Render a rectangular object at p_wg
```

6.4.4 Implementing Data-centric Depth Cues for GIS Data

The implementation of method that creates data-centric depth cues from GIS databases is more challenging than the methods that create cues for single data points. Data-centric cues require that various visual representation of the same GIS data can be created and managed. The big challenge is that several GIS applications, such as surveying, require that the data can be interactively modified during usage. Thus, it is highly important to maintain consistency with the GIS database for all additional virtual cue geometries during interactive modifications.

In order to achieve this consistency, we separate the data sources from the visual representations by introducing two different types of data levels in our visualization architecture:

- GIS-database level
- Comprehensible 3D geometry level

The GIS-database level consists of a set features, each describing one physical world object with a 2D geometry and a set of attributes. Attributes are stored as key-value pairs and provide a description of various properties of the feature, such as type, owner or status. In contrast, the second level, the comprehensible 3D geometry level, consists of

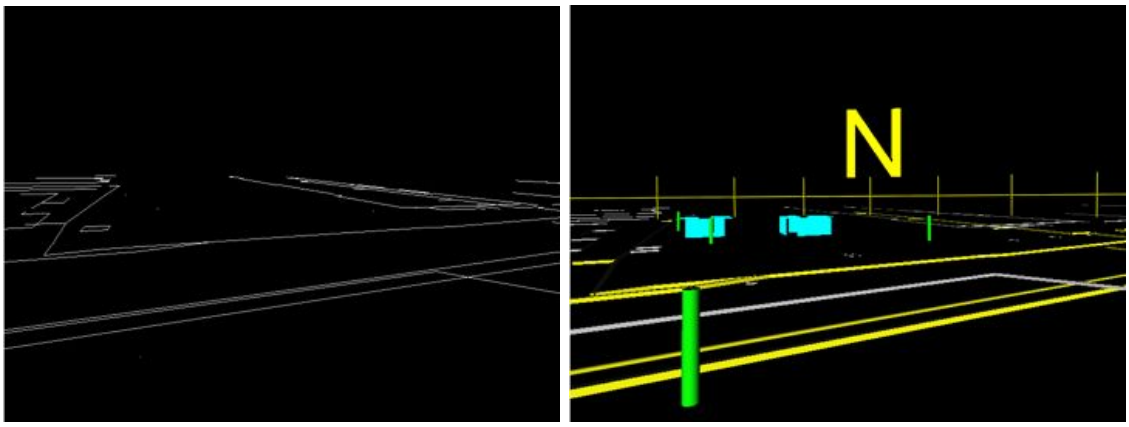


Figure 6.8: GIS data model vs. transcoded geometries. Left) GIS data model, all data is represented by lines and points, which makes interpretations difficult. Right) Transcoded geometries, GIS data is transcoded to a comprehensible representation showing cylindrical objects representing trees and cylindrical objects representing pipes. Color coding enables the user to interpret semantics.

a set of 3D geometric representations of physical world objects, such as extruded circles, rectangles, polygons and arbitrary 3D models; visualizing pipes, excavations, walls or lamps respectively.

So far both levels exists separately and can not interchange data. We add a new data layer that serves as transmission layer to support the consistency between both data levels. We call this layer *transcoding layer* as it supports the bi-directional conversion of data between the comprehensible 3D geometry level and GIS database level. Each feature of the GIS database is stored as scene graph object with a set of attributes. Interactive modification in our system are conducted at this level and automatically propagated to the two other levels. Applying manipulations at the transcoding layer allows manipulating feature data directly and avoids that manipulations are only applied to specific 3D geometries. For instance, the exclusive manipulation of an excavation geometry of a pipe makes no sense without modifying the line feature representing the pipe. Furthermore, the transcoding layer has still access to the semantic information of a feature, which is important since interaction methods can depend on the type of object.

We implemented a bi-directional transcoding pipeline that creates the transcoding layer and the comprehensible 3D geometry level automatically from the geospatial data and updates the database with manipulations applied in the AR view. The pipeline is working as follows: (1) The conversion of GIS data into the transcoding layer and into specific comprehensible 3D geometries. (2) If the user manipulates the data in the AR system, the data connections between the three data layers guarantee data coherency while interacting with the data. To avoid administration overhead, modifications are recorded through a change tracing mechanism and only changed features will be written back to the GIS database.

Real-time AR visualization and manipulation of geospatial objects requires a different data model than traditional geospatial data models. Abstract line and point features need to be processed to create 3D geometry representing more the actual shape than the

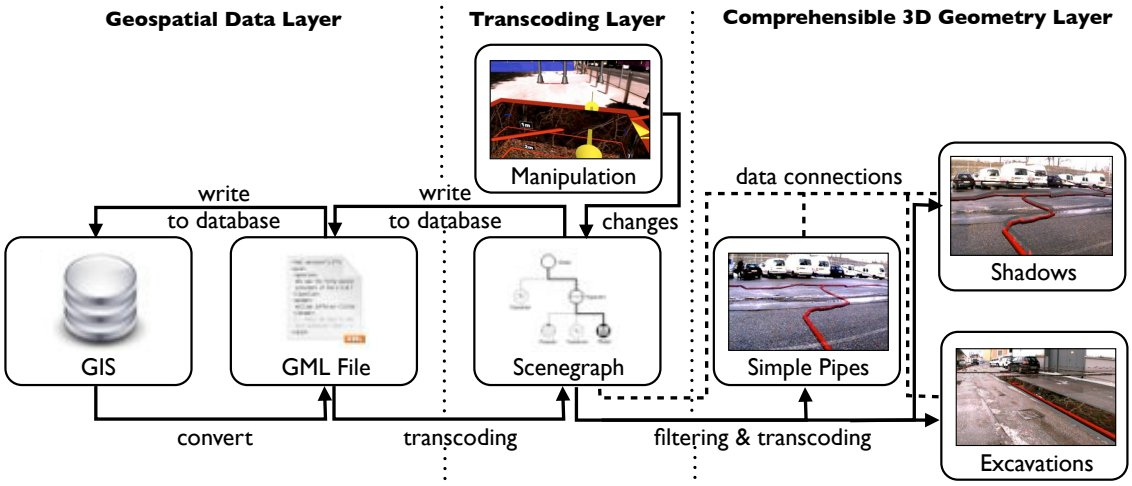


Figure 6.9: Overview of the bi-directional transcoding pipeline. Data from the geospatial database is converted to a simple GML exchange format. The GML file is imported to the application and transcoded into the transcoding layer representation. Filtering and transcoding operations map the transcoding layer data to comprehensible 3D geometry. Data connections between the transcoding layer scene graph and the comprehensible geometry scene graph keep the visualization up-to-date. User interaction is applied directly to the transcoding layer.

surveyed line of points (compare Figure 6.8, (a) and (b)). In particular, it is important to create additional virtual cue geometries that improve the comprehension of the presented information in the AR overlay.

All of these additional geometries need to be interactive and modifiable, so that interactive manipulation allows for updating the features. To support these operations we developed a bi-directional transcoding pipeline that realizes the conversion from GIS features to comprehensible 3D data and back (Figure 6.9). A transcoding operation using a commercial tool (FME) extracts and translates geospatial data into a simpler common format for the mobile AR client, listing a set of features and their geometries (for a detailed description refer to Section 3.2). The AR system further filters the features for specific properties and applies transformations to generate 3D data from them. The 3D data structures are functionally derived from the geospatial data and stay up-to-date when the data changes. Interactions operate directly on the feature data, synchronizing the 3D visualization and features.

From Geospatial Data to the Transcoding Layer As explained in Section 3.2, we use FME for extracting the features from the geo-database and to create a GML file containing all features in a selected area of interest. This area of interest is selected interactively by the user in the back-end GIS. Each feature in the GML file describes either an as-planned or an as-built object in the physical world.

The GML file is converted into a scene graph format representing the data in the transcoding layer.The transcoding layer finally consists of a set of features, each representing semantic attributes and geometric properties.

From Transcoding Layer Data to Comprehensible Geometries So far we have the data available as abstract 3D representations. The next step is the creation of 3D geometries that support the comprehension of the users. The way how the geospatial data will be visualized strongly depends on the application, application domain and the preferences of the user (e.g., color, geometry, or geometry complexity). For instance, a pipe could be represented in several ways, such as a normal pipe using an extrusion with a circular ground plane or as an extruded rectangle to show an excavation around the pipe (Figure 6.10). We call the conversion from the transcoding layer data representation to comprehensible geometries *geometry transcoding*. Different types of transcoding operations are called *transcoders* and each transcoder can be configured offline or during runtime to create different geometries from the same geospatial data. Each comprehensible 3D geometry is independent from other 3D representations of the same corresponding feature but connected to the feature data in the transcoding layer (Figure 6.9).

The implementation of different visualization styles for different feature types is supported by a *filtering-transcoding* concept. The filtering step searches for a specific object type from attributes stored in the transcoding layer and the transcoding step transforms the data into specific geometric objects, which can later be displayed by the rendering system. The separation of the two steps allows for a flexible system that can support many professional applications that use data from GIS databases.

The *filtering* step searches for specific tags in the semantic attributes of features in the transcoding layer and extracts the corresponding features. For instance, features can be filtered by a unique id, a class name, class alias, or type. The matching is implemented through regular expressions testing against the string values of the attributes. The features extracted by filtering can then be processed by an assigned transcoder. Filter rules and transcoding operations can be configured by the application designer using a script or during runtime. The mapping of filters and transcoding operations has to be implemented in the application and allows not only for a configuration of the visualization methods for specific data, but also a filtering of the presented information.

Each *transcoding operation* depends on the type of transcoding and the transcoding parameters. The transcoding type assigns the underlying geometric operation for deriving the 3D geometry, for instance converting a line feature into a rectangular extrusion representing a excavation along the pipe. The transcoding types comprise:

- PipeTranscoder: Converting polygonal chains into circular extrusions representing pipe geometries.
- ExtrusionTranscoder: extruding polygonal lines to extruded rectangular objects.
- PointTranscoder: Converting point features into upright cylinders or specified 3D models being placed at the point features location.
- DepthGeometryTranscoder: Converting polygonal lines into excavation geometries.

The transcoding parameters configure visualization specifications such as color, width, radius, height, textures and 3D models of the objects.

Multiple transcoders can be used to create different visualizations of the same data. The user selects the appropriate representation during runtime and the geometry created

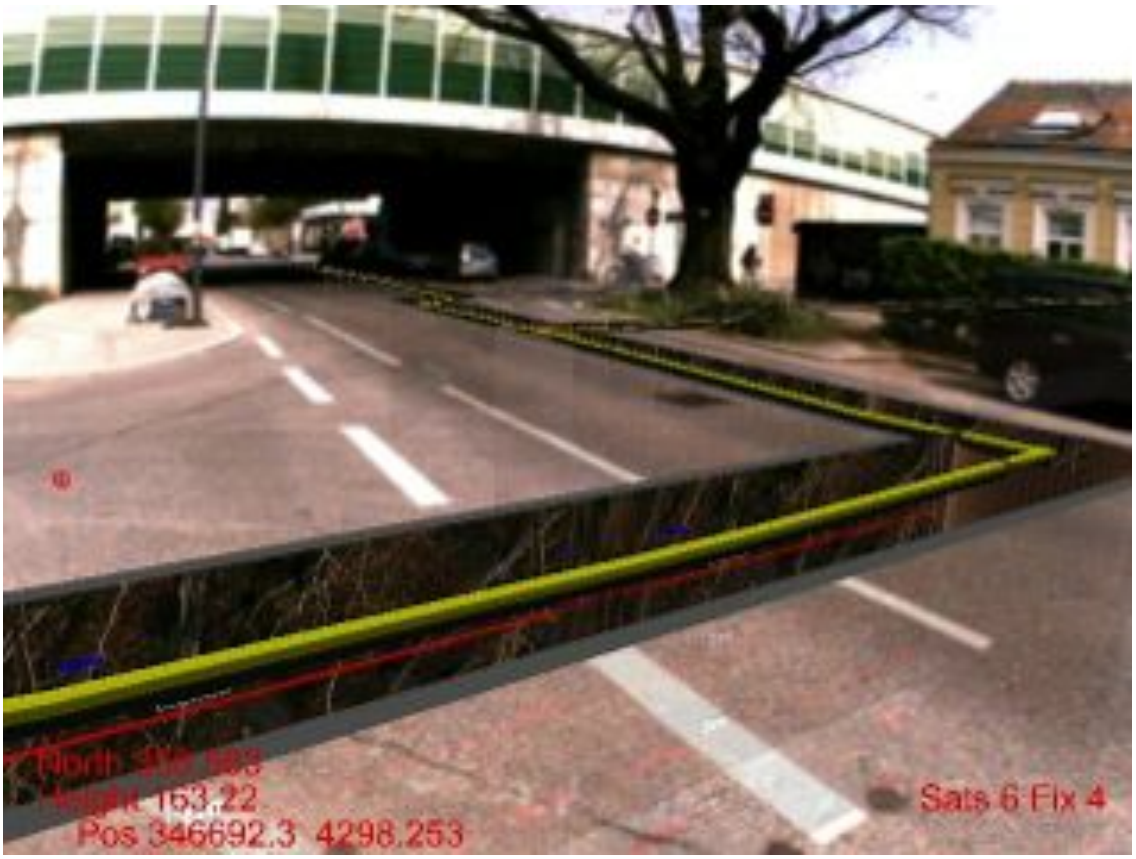


Figure 6.10: Excavation along a yellow pipe. Excavation and pipe are only rendered opaque at pixels that are covered by the top of the excavation. Outside this mask virtual content is rendered transparent.

```
DEF WATER_FILTER FeatureFilter {
    attribute_id ""
    attribute_name "water"
    attribute_alias "WA pipe"
    geometry_type "GmlLineString"
}
```

```
DEF WATER_TRANSCODER PipeTranscoder{
    radius 0.5
    material SoMaterial{
       diffuseColor 0 0 1
       }
}
```

Figure 6.11: Example for a filter and a corresponding transcoder.

by the filtering and transcoding is finally rendered by the AR application. The transcoding is usually performed during start-up time of the application and takes several seconds for thousands of features.

Implementation of Object-aligned Cutaways The object-aligned cutaways show an excavation along a subsurface object. The cutout area is given by the projection of the subsurface object on the ground plane. This projection is computed by using the intersection between a ray starting at each underground object and being orthogonal to

the ground plane given by the *Digital Terrain Model (DTM)*. From the cutout area we compute vertical and horizontal rectangles to form an excavation-like geometry object. We can use these geometries to apply a similar rendering as for the user-centric magic lens. As shown in listing 6.4, we start by rendering a mask of the top of the excavation into the stencil buffer. After, we render the camera image and overlay all virtual geometries (pipes and cutaway geometry). For all pixels outside the stencil mask the camera image is rendered transparently on top. This creates the effect shown in Figure 6.10, where the excavation and pipes are only opaque at pixels that are covered by the top of the excavation. Outside this mask, the virtual content is rendered transparent.

Listing 6.4: Rendering of object-aligned cutaways.

```
Render the top of all cutaway geometries into the stencil buffer
Render video texture
Render all cutaways without top geometries
Render virtual data with enabled depth testing
Render transparent video texture using the stencil buffer
Render frame geometry for all cutaways
```

Implementation of Reference Shadows Reference shadows show the outlines of the subsurface object in relationship to the ground plane. To create the reference shadows, the transcoder calculates the intersection of the projection of an underground object and the DTM. The intersection points are used to create a semi-transparent flat shape that is located on the ground level. The rendering combines video image, virtual shadow geometries and the virtual data.

Listing 6.5: Rendering of reference shadows.

```
Render video texture
Render virtual shadows geometries with selected transparency
Render virtual data
```

Implementation of Connection lines Connection lines visualize virtual connections between the virtual objects and the ground plane. In order to create them, we define a transcoder that computes the point on the ground plane that results from projecting the 3D outline points p_v of virtual object to the ground plane. Adequate outline points are for instance start and endpoints of a pipe segment. The transcoder calculates the intersection point p_i between a ray starting at the virtual object contour point and having a direction orthogonal to the ground plane. Finally the transcoder creates line geometry connecting p_v and p_i (Figure 6.4, Bottom Right). The connection line geometries are rendered for each object that are defined to use this transcoder.

Listing 6.6: Rendering of connection lines.

```
Render video texture
Render virtual connection lines
Render virtual data
```

Junctions Junctions are virtual geometries indicating the connection of virtual objects to the ground plane. For instance, two flat rectangular objects can be used to represent a cross on the ground. For this purpose, we define another transcoder that allows creating junction geometries. The transcoder calculates the projection p_i of the virtual geometries on the ground plane. Finally, the transcoder places two flat rectangular shapes in such a way that they have their middle point at p_i and a normal orthogonal to the ground plane. These geometries are displayed for each object that was defined by the filtering-transcoding concept as being represented by a junction geometry.

Listing 6.7: Rendering of junctions.

```
Render video texture
Render junction geometry
Render virtual data
```

6.4.5 Results

We integrated our approach into the mobile AR platform described in Section 3.3. This platform is designed for high-accuracy outdoor usage and allowed to test the virtual cues created in this chapter. Additionally, we gained first feedback with the setup in workshops with expert users from civil engineering companies.

Transcoding Times Before we started to test the approach in the field, we analyzed the time that is needed to transcode a set of selected GIS features into our AR visualization. Based on these timings we wanted to analyze if the transcoding times are feasible for an outdoor application.

Table 6.1: Transcoding Times.

	Salzburg (1397 Features)	Sandgasse (4003 Features)
Desktop	4.98s	18.02s
Tablet PC	5.47s	18.39s

Thereby we compared two datasets describing two different test sites provided by different civil engineering companies. We compared the transcoding time on a desktop PC (Intel Core 2Quad CPU Q9400, 2.6GHZ) and a tablet PC (Motion J3400 with 1.6GHz Pentium CPU). As shown in Table 6.1, the transcoding time is not real-time, but can be achieved in a reasonable time when only doing once during the start-up process of the application. The timing further shows that a limitation to a smaller working area can reduce the transcoding times notably.

Expert Interviews The developed visualization techniques demonstrate virtual pictorial cues that mainly focus on professional application scenarios. Thus, it is difficult to evaluate them with standard users having no background in these professional applications. To gain insights about the usefulness of these virtual cues, we performed an explorative study with professional users from the civil engineering industries within an industrial surveying scenario.

Figure 6.12: Field trials with expert users.

For this purpose, we used data from conventional GIS provided by two civil engineering companies to provide a realistic working environments to the experts. We worked on two different urban test sites, one in Vienna and another one in Salzburg (Austria), where we performed field-trials using the AR system for on-site surveying with 16 expert participants (12m/4f). The idea of the explorative study was to gain feedback of the expert users within an industrial task. For this purpose, we decided to set the study in a surveying application. As described in Section 2.4.2, in surveying tasks, the user has to measure points in his physical proximity. By using an AR system, the user directly receives a visual feedback about the dimensions of the surveyed object. By overlaying the surveyed object, the user can compare the actual physical object with the surveyed virtual object. For the surveying task in the study, we used a part of a plastic pipe that we placed on the ground.

We started with explaining the AR system and the surveying task. After the introduction, each participant was asked to survey an existing plastic pipe using our system. After finishing the surveying task, we asked the participants to complete a short questionnaire where we asked questions about the general AR system and the perceived registration quality as well as asked for the visualization techniques. 69% of the participants stated that they have practical experience with surveying. For the ratings, we used a Likert Scale ranging from 1=Very Bad to 7=Very Good. Results of the questionnaire (Figure 6.13) show that users rated the AR system in total as being useful (M=5.9, std=1.1) and the registration being rather good (M=5.4, std=1.1). This shows that the expert users see the potential of using AR for a surveying task and found the registration acceptable. But it also shows that there is still room for improving the registration. Furthermore, we asked questions about the visualization itself. The results show that the user generally rated the virtual cues represented by the excavation and the virtual shadows above average. They rated the virtual shadows with a mean value being approximately good (M=6.2, std=0.7). The virtual shadows received a rating between rather good and good (M=5.6, std=0.8). This indicated that the expert user see a general benefit in using these virtual cues for improving the perception in surveying with AR.

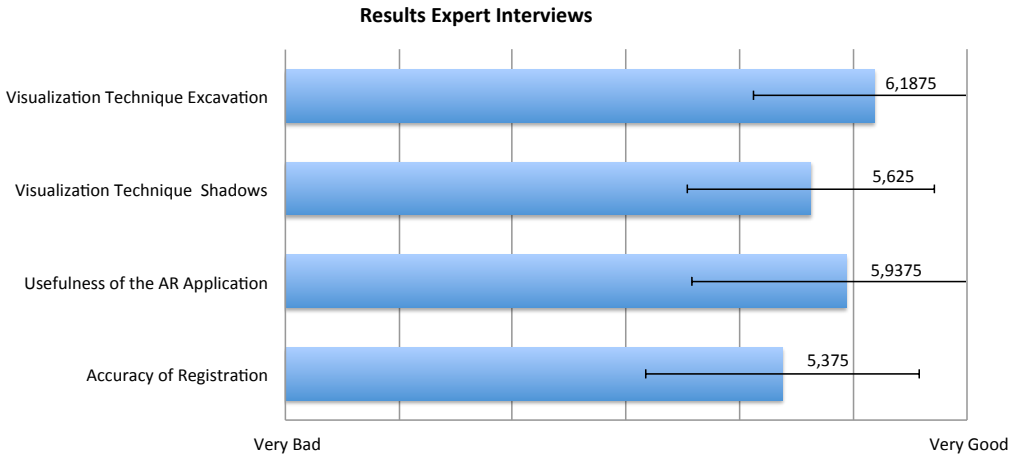


Figure 6.13: Results of the interviews with expert users.

6.5 Summary

In this chapter, we described how virtual pictorial cues can be automatically created from professional data sources, such as GIS databases or MAV waypoints. These virtual cues support the depth perception and in particular support the user in deriving absolute depth information. This is important, because the depth perception for subsurface objects and floating objects is difficult since several depth cues are not working or only working partially. While, user-centric cues can be easily created once the user's pose is known, for data-centric cues a more sophisticated cue creation pipeline is required to maintain consistency between the data and the geometric representation during interactive manipulations. For this purpose, we implemented a data roundtrip which allows a comprehensible AR visualization and is still flexible for modifications. We demonstrated these interactive functionalities by integrating a set of interaction tools for manipulating and surveying features. Additionally, we investigated how planning and surveying of infrastructure elements is supported by testing these techniques with professional users from civil engineering.

Chapter 7

Information Filtering and Abstraction

Contents

7.1 Introduction

In the last chapters we focused on how to achieve a convincing scene integration of virtual content and how to support users in depth perception. Nevertheless, the presented methods do not address comprehension problems that occur when visualizing complex data. This is the main goal of this chapter. As described in chapter 1 the visualization of complex information in AR poses several challenges:

1. The overlay of an already complex physical outdoor environment with complex information is likely to be subject to information clutter.

2. Complex data may occlude important information of the physical environment.

3. Complex virtual information may be subject to self-occlusion.

In contrast to other visualization environments (e.g VR applications), in AR the user cannot simply change the viewpoint to explorer complex data from different viewpoints to

Figure 7.1: Simple blending of complex data. The current state of the construction site is overlaid with 3D data of one previous point in time. Left) Transparent overlay. Right) Overlaying a grey 3D mesh for highlighting the difference between virtual data and physical environment.

get more insights. The view of the user is fixed to his own viewpoint, since the visualization is registered to the physical world.

To approach these challenges in AR, we investigated the abilities and limitations of interactive Focus&Context techniques in combination with information filtering. Based on these findings, we extended these methods with techniques for information abstraction to support the comprehension of complex data in outdoor *Augmented Reality (AR)*. These methods allow for a comprehensible view on the complex data while still preserving the spatial relationship between physical world and complex data.

As an application scenario for complex data visualization in AR, we choose the visualization of progress of a construction site. The main goal for construction site monitoring is to visualize changes of construction sites over time (Section 2.4). The visualization of this kind of information in an AR view allows preserving the spatial relationship of the progress information in the context of the physical world. This enables construction site staff to reproduce and understand progress, and link possible mistakes directly to the physical entities on the construction site. As described in Section 3.2, we capture progress information by using aerial vision. For this purpose, a *Micro Aerial Vehicle (MAV)* flies over a construction site and captures a set of aerial views on a regular basis. Based on this data, the construction site can be reconstructed at different points in time. This finally results in a set of 3D data sets representing different states of the progress can be represented. Due to the additional time dimension, we call this kind of data 4D information.

Presenting 4D data in a simple AR view will directly lead to the above mentioned problems of visualizing complex data in AR. As shown in Figure 7.1 already the simple overlay of merely one data set (representing one point in time) is subject to information clutter. Neither a transparent nor a color coded overlay supports the comprehension. Both visualization methods make it complicated to understand the previous state of the construction site. Moreover, a lot of the physical world context is occluded by virtual data. Due to this occlusion, it is complicated to even understand the current state of actual physical environment. To sum up, using an simple overlay of complex 3D data, the user does neither understand the previous state nor the current state.

The problem becomes even more challenging for the visualization of multiple 3D data sets as given by a 4D representation. Instead of having one data point for each point in

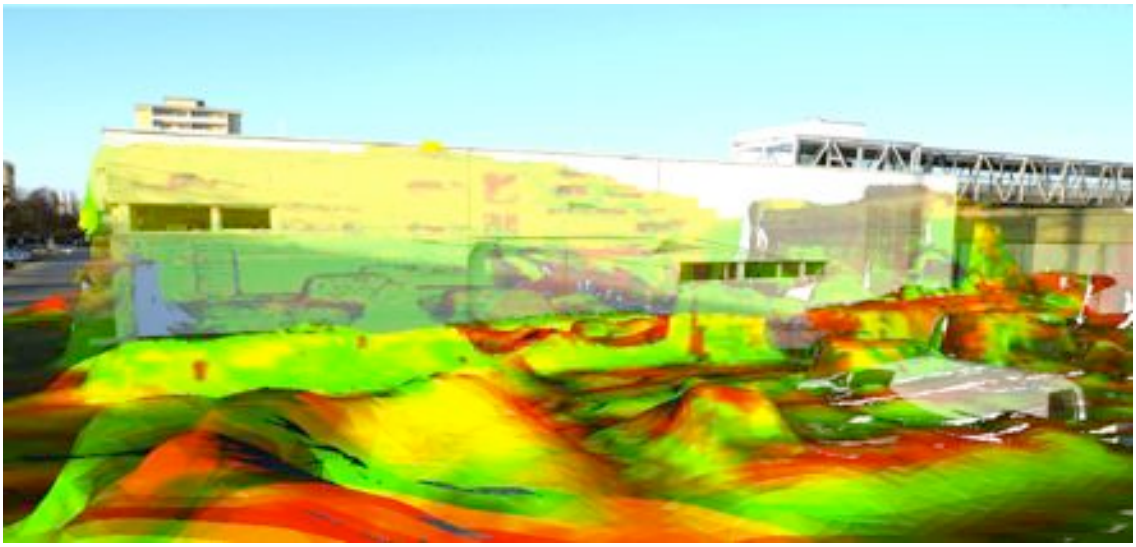


Figure 7.2: Visualizing two different points in time in one view with different shadings leads to image clutter as well.

space, in this case, we want to visualize multiple data entries representing different points in time for each 3D point in space (Figure 7.2).

The main research question of this chapter is how to effectively visualize this kind of complex data in an AR overlay. In the next section, we will start by discussing traditional methods for the visualization of complex data in AR. We will show that they can be used to visualize one point in time and even a small number of points in time. For this purpose, we apply a spatial information filtering controlled by Focus&Context techniques. These techniques allows one to compare a limited number of previous points in time to the current state. However, we will discuss the drawbacks of these methods that appear when visualizing more datasets in one view. In the main section of this chapter, we will introduce methods that use information abstraction to reduce the complexity of the visualization of 4D datasets.

7.2 Background: Traditional Information Filtering

Spatial information filtering combined with a set of interactive Focus&Context tools can be used to explorer a limited number of points in time. For this purpose, the user interactively selects a focus area to define where information about a previous point in time is displayed. The information filtering helps to only displays virtual information in the dedicated area. Vice versa, the current status of the construction site, which is represented by the camera image, is only shown in the context region.

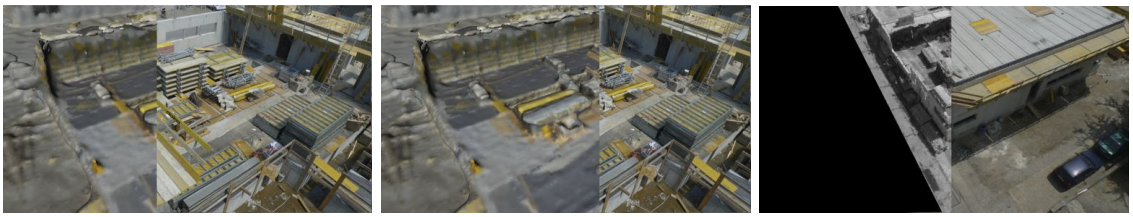


Figure 7.3: Side-by-side visualization using 2D sliders. The mouse input defines the border between video image and virtual content. Left and Middle) The virtual content is overlaid on the right side of the border. Right) On the right side of the border only the virtual data representing a previous point in time is displayed.

7.2.1 2D Focus&Context Tools

2D Focus&Context tools address the problems of information clutter by allowing the user to define a focus region in image space. Within this 2D focus region, the virtual information presenting a previous point in time is displayed. The remaining part of the view that does not contain the focus area is called context region. In the context region, the current status of the construction site is displayed using the current camera image of the physical environment. The interactive tools that we will discuss in the following comprise 2D sliders, 2D magic lenses and a selection of 2D regions. For a fast prototyping, we use shaders that allow for a fast adaption of the spatial information filtering based on different focus areas.

2D Slider Tools that are popular for interactive side-by-side visualizations, in particular, for before and after comparison of urban or historical scenes, are 2D sliders. For instance, 2D sliders combined with aerial photography helped to visualize the amount of damage during the 2011 Tohoku earthquake and tsunami in Japan[1]. Another application example is the documentation of urban development. Here, buildings are often displayed in their current state and in the past[2]. In order to provide such a visualization using photographs, it is important that both images (past and presence) are captured from the same view. The process of taking the picture from the same position is called re-photography. The photographer of the newer image has to take the same position as the one of the older photography. In order to find a similar position, photographers search for specific features in the scene that are easily to recognize such as walls or roofs. Recently, researcher even developed automatic methods to guide the photographer to the right pose by using SIFT features [87] and pose estimation [9].

The drawback of re-photography is that it limits the observer to views that were captured in the past. If the previous status is available as a 3D reconstruction, it is also possible to choose different viewpoints. By overlaying the 3D information to the camera image of the current environment, the technique moves from re-photograph to AR (Figure 7.3). Similar to the methods for re-photography, the the user can interactively move a slider in image space to control the border between video image and the virtual overlay

[1] `http://www.nytimes.com/interactive/2011/03/13/world/asia/satellite-photos-japan-before-and-after-tsunami.html?_r=0`

[2] `http://zeitsprung.animaux.de`

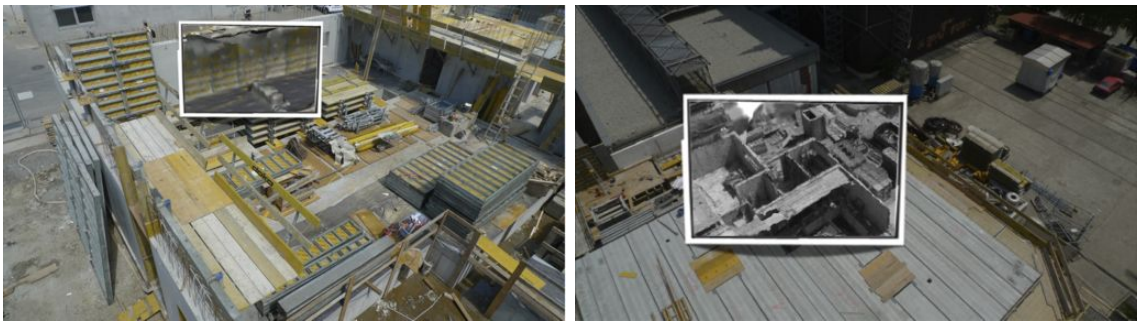


Figure 7.4: 2D Magic Lens. Inside the focus region (highlighted in white) the virtual content is displayed. Left) Overlay of virtual data inside the context region. Right) Grey shading inside the focus area supports the differentiation between virtual information and physical world.

containing the 3D information. In our application, the 3D information contains a previous status of the construction site, but the technique could also be used for different urban scenarios as long as the 3D data is available. When the user clicks at an arbitrary location in the 2D view, the x-coordinate of the mouse pointer is used to define the border. The fragment shader then displays the video image for all fragments with x-coordinates larger than the border's x-coordinate. For other fragments the virtual data is either overlaid to the video image (Figure 7.3, Left) or rendered exclusively (Figure 7.3, Right).

2D Magic Lens Another technique that artist and photographer sometimes use to create before and after effects is embedding a cutout of old photographs into new ones. This technique is similar to magic lens interfaces from Information Visualization [14]. In contrast to the 2D slider, a 2D magic lens allows the user to move a dedicated focus area to the regions of interests. Mouse coordinates define the center of the focus region. The focus region can have different shapes, such as a rectangular or circular shape. Based on the shape definition, the fragment shader tests if a fragment is inside the focus area and renders the virtual content in this case. Thereby the virtual content is again either displayed using an overlay or exclusive rendering. In addition, the virtual content can also be rendered using different shadings, such as Toon shading or a grey scale shading to highlight the difference to the actual physical environment (Figure 7.4).

2D Image Regions A method that has more influence on the regions showing the previous status, is the selection of image region. Some artists, such as Larenkow, used such a technique to define image regions of interest that show historic content[3]. Larenkow manually selected parts of an image that was replaced with an historic view. With this method he combined current views of buildings in Berlin with photographs captured during WorldWarII.

We can apply a similar technique in AR overlays. The idea is to allow the user to interactively select regions of interest in the camera image that serve as focus regions. Only in these focus regions the complex virtual data is visualized. For this purpose, we

[3] `http://sergey-larenkov.livejournal.com/pics/catalog`

Figure 7.5: Information Filtering based on 2D image region selection. The user interactively selects image regions in the AR view. For the selected regions the virtual information is displayed (highlighted in red).

provide the user with an interactive tool that allows one to select an image region in the AR view. The selected image region is given by an oversegmentation of the camera image and the mouse coordinates of the selection. This region-based selection allows to preserve the context of regions with the same visual characteristics.

The quality of this visualization technique depends on the quality of the segmentation. The more the segmentation preserves the visual coherency, the better. An ideal solution would use methods that segment the image in semantic regions. Since these segmentation methods are still computationally too expensive, we use an over-segmentation based on superpixel. By clicking on the regions of interests, the selected regions are added to the focus area and used to visualize the virtual content (Figure 7.5 Right).

Importance Driven Filtering An image-based technique that preserves important image structures is importance driven filtering. It varies the visibility depending on the image structure's importance, similar to the image-based ghostings presented in Chapter 4. The user can change the amount of preserved image structures by modifying the threshold of importance. With this approach the user can exclude less important image structures depending on his preferences.

For this purpose, we compute importance measurements of the camera image and the reconstructed point cloud data to decide which part of the scene has to be preserved. Importance measurements include: saliency of the camera image, the amount of changes in specific image parts, edges in the point cloud and the image (Figure 7.6).

Discussion The advantage of the 2D Focus&Context tools is that the visualization requires no additional information or post processing of the reconstructed data. However, they have the disadvantage of not addressing self-occlusion, since they simply define a 2D region in the user's view. This means we can filter information in image space but not in depth. Furthermore, it is difficult to present more than one point in time.

Figure 7.6: Importance driven filtering. The visibility of 3D varies based on the importance of regions in the camera image. For less important image parts, we display 3D data. In regions with important image structures, we display the camera image. This allows the user to see both, important image structures and the 3D, at the same time.

7.2.2 3D Focus&Context tools

In order to address the problem of self-occlusion, we investigated tools that allow one to define focus and context regions in 3D. This allows to spatially filter the information in all three dimensions. Such a 3D filtering is in particular interesting for the visualization of 3D data that was reconstructed with aerial vision, since the users may want to inspect one specific element that is occluded by other previous structures in their view. By defining a focus area in 3D, it is possible to exclusively visualize information for this selected region.

Another interesting aspect of 3D tools is their ability to convey information about depth. This supports the understanding of spatial relationships between objects at the present time and objects from the past.

3D Slider Similar to the 2D Slider, the 3D slider allows to separate the visualization in a focus area showing the 3D information and a context area showing the camera image. The difference between both techniques is that the 3D does the separation in 3D. The 3D slider is defined as a large wall that the user can move in the scene. To provide a convincing visualization, the wall has to be aligned in relationship to a plane in the 3D world. We provide a 3D manipulation tool that allows the user to interactively manipulate

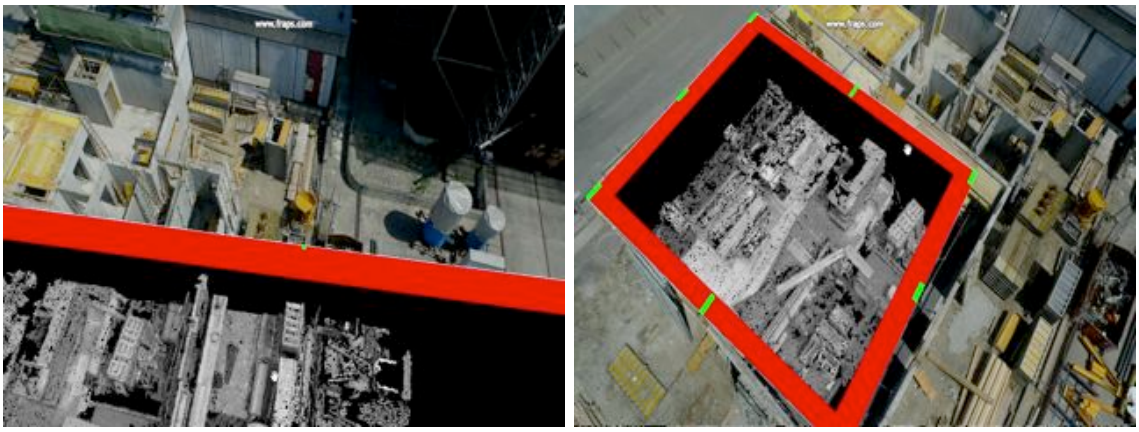


Figure 7.7: 3D tools for information filtering. Left) A 3D slider allows the user to divide the view in an area where a previous status of the construction site is shown and a area where the current status is shown. Right) The 3D magic lens magic lens defines a three dimensional region where the virtual content is displayed.

the alignment and the dimension of the 3D slider. The intersection plane between virtual geometry and sliding wall provides information about the depth and height of the virtual content.

3D Magic Lens In order to allow the selection of a smaller focus region, we provide a 3D magic lens. The 3D magic lens allows the user to define a box-shaped focus area in the AR view. The box can be interactively moved and scaled to fit the requirements of the user. Inside the box the virtual content is displayed. For all elements that are outside the focus area video image information is shown. Similar to the 3D slider, the alignment has to be done in relation to 3D planes in the scene to achieve a convincing visualization. An extension of this magic lens would align itself to the planes in the scene, for instance by detecting planes in the 3D point cloud.

3D Color Coded Magic Lens So far, we only discussed visualization tools supporting the overlay of one 3D data set. This allows only to visualize one previous point time at once. Several professional applications, such as construction site monitoring, require to compare multiple points in time to inspect progress. With the existing tools, this is only possible by presenting different points in time one after another. Unfortunately, such an approach is often a subject to change blindness [117]. To address this problem, we further investigated methods that allow for the presentation of multiple complex datasets in one view. For this purpose, we visualized different points in time using different visualization techniques. This allows a differentiation between multiple datasets. For instance, we use a color coded rendering to display different 3D datasets. Thereby the different color codings are implemented by enabling the corresponding color buffers (Figure 7.8). This technique again makes only sense in combination with Focus&Context techniques, because otherwise the view is too cluttered to understand the changes (Figure 7.2). Combining the color coding with multiple magic lenses allows for the comparison of multiple points in time.

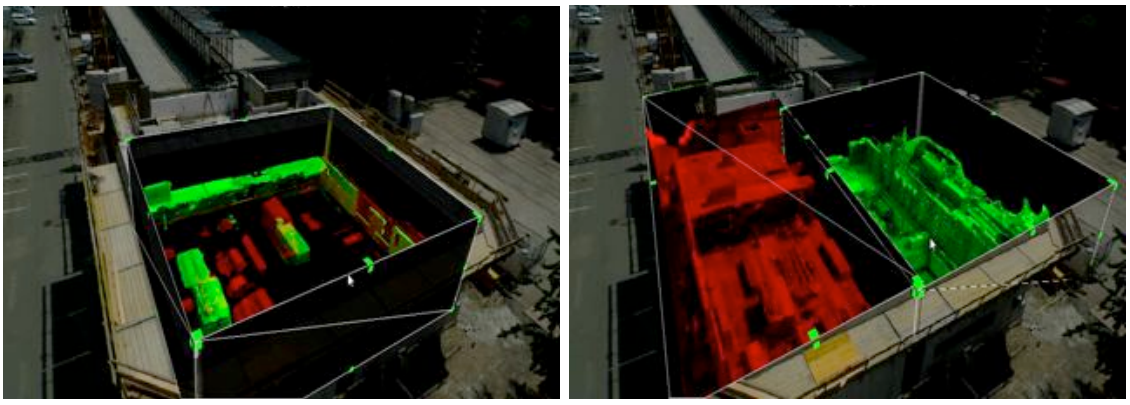


Figure 7.8: 3D Focus&Context tools using different color codings. Thereby green represents an earlier stage of the construction site as red. Left) Two color codings are used in the same 3D magic lens. Right) Two 3D magic lenses display different points in time.

7.2.3 Limitations of Information Filtering

The information filtering methods presented in this section require a high amount of user interaction, such as moving a slider or a magic lens to compare the current status with the previous one. Since for most of the methods, previous and the current status are never displayed at the same time in the same region, it often requires a lot of mental effort to understand what exactly had changed. Usually, users try to solve this problem by switching several times between both views. But if there is to much change between two views the change is often not perceived [117]. The biggest problem of these techniques is that they are limited in the number of data sets that can be displayed at the same time. In particular, for the visualization of time-oriented data, with these information filtering techniques we face the limitation that with a higher number of points in time information clutter will increase.

7.3 Visualization Concept for Multiple Datasets

The last section and previous work showed that AR can be used for comparing two different 3D data sets, such as as-planned with as-built data [97, 115] or visualizing a single previous snap-shot of a building [70]. However, there is still a gap in research for visualizing more than two datasets in one view. Moreover, in AR there is no research on comparing multiple datasets representing different points in time, as it would be required for inspecting progress on a construction site. This means, in AR it is still complicated to follow changes over a time period.

In this section, we present an interactive AR visualization technique that displays data in time and 3D space using a combination of abstraction and Overview&Detail techniques. The presentation of 4D information at different levels of abstraction and detail allows the user to thoroughly analyze dynamical scenes. One the one hand, overview techniques allow one to inspect multiple objects and multiple points in time. On the other hand a detail view presents information of a selected object at a selected point in time. Since the 4D

Figure 7.9: Cluttered visualization of time-oriented data. By simply blending a single or multiple points in time into the real world view the visualization either becomes difficult to understand or changes between different points in time become difficult to track. The visualization shows two consecutive points in time in separated views. Clutter and complex changes make it difficult to comprehend the changes. Left) Overlay of a 3D mesh representing the beginning of the construction process. Right) Overlay of 3D reconstruction when scaffolds were already set up.

visualization is registered to the physical world, the user can inspect changes in relation to the physical world context.

The visualization of time-oriented data is traditionally a topic of information visualization. A large body of visualization techniques have been developed over the last decades, which all aim to allow efficient and effective analysis of the time-oriented data. Existing techniques range from very simple line plots [50] and bar charts [50], up to more complex visualization techniques such as Braided Graphs, which combine data from multiple points in time into one view [59], or the Time-Space cube, which combines time-oriented data with spatial information [48]. For an extensive review of visualization techniques for time-oriented data refer to [2]. However, existing visualization techniques for time-oriented data commonly have been designed to study the data in front of a PC. The main goal of our technique is to allow for a inspection of time-oriented data on-site using AR to provide the user with contextual information about the physical world.

In order to visualize changes in a comprehensible way, perceptional issues like change blindness have to be addressed [117]. Change blindness describes the problem of not noticing changes. This problem often appears when a lot of changes occurs in the view of a person. To avoid change blindness, it is important that the user can keep track of selected changes for instance by providing visual additional hints [94]. This is particularly important for outdoor AR visualizations, since the environment is changing all the time and important changes in the data may be concealed. Traditional information filtering tools did not addressed change blindness (Section 7.2). Often there is too much change between multiple data sets, which makes it tedious to identify important changes. In Figure 7.9, we show an example where it is complicated to understand the changes even when using time-separation visualization tools, such as sliders.

To address all information clutter and change blindness, our technique follows the *Visual Information Seeking Mantra* of Shneiderman [116]. The mantra defines visualization guidelines as follows: 1) There should be an overview first, 2) then zooming and filtering to concentrate on important information elements and 3) finally get details on demand. In following, we will show how the mantra can be applied for visualizing multiple complex 3D datasets in AR. We start with defining three visualization levels varying in their level

Figure 7.10: 4D visualization concept. Multiple level of detail allow to explore time-oriented data in an AR visualization. The highest level of abstraction provides just enough information to provide an overview of the data in relation to the objects in the AR environment. The second level of abstraction presents traditional time-oriented visualizations registered in AR to enable an effective analysis of time-oriented data. Its registration in AR additionally provides information about its relation to real world structure. The third level provides structural detail of the object of interest for a selected point in time and 3D space. To allow to first study the data in a higher level of abstraction before analyzing it in more detail we interactively combine all level using Overview&Detail and Focus&Context techniques.

of detail.

To clarify the description of these visualization levels, we group the virtual content presented in the visualization in the following way: (1) The *Scene* describes all virtual objects of interest in the visualization. In Figure 7.10 the scene is the set of walls of the white building. (2) The *Object* describes one object of interest in general. For instance, in Figure 7.10 an object is one of the walls of the white building. The object itself can be represented by rendering an abstract visualization of a certain characteristic of this object, an overview of multiple points in time or by rendering the object in detail for a selected point in time. (3) The *Object Detail* refers to the actual geometry and appearance of one object at a selected point in time (Figure 7.10 Bottom).

The visualization levels are based on this content grouping. According to the information seeking mantra, each level display the virtual content on a different scale:

- L0: Scene overview level
- L1: Object time overview level
- L2: Object detail level

Each visualization level is a detailed view of the higher visualization level. Transition techniques such as Overview&Detail and Focus&Context allow for moving between the

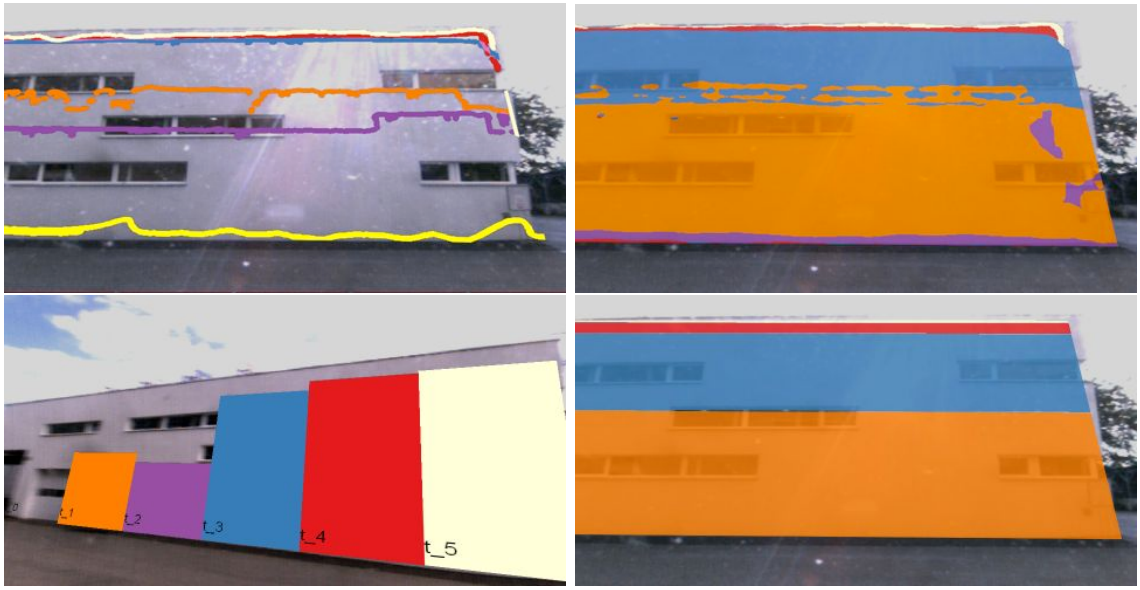


Figure 7.11: Object time overview visualization: providing information about multiple points in time for a single object. The different points in time are thereby color coded. Top Left) Height Lines showing different heights of the wall at different points in time. Top Right) Geometric Completion shows the completion in geometric reference to the wall. This provides a summary which parts of the wall were added at which time. Bottom Left) Block diagram showing the average completion of the wall for multiple points in time. Bottom Right) Completion diagram showing the average completion for multiple points in time.

different visualization level and relating them to each other. We first discuss each level of abstraction in detail, before we describe how we move from one to the other.

7.4 4D Visualization Level

The first visualization level L0 represents abstract information for individual objects through aggregating time-oriented attributes per object. These attributes may include information such as completion or progress and will be represented with a per-object visualization such as a color coding. While the first level allows for global overview in the scene according to *overview first*, the second level L1 presents more detailed information about one object according to the second item of the mantra. This level information displays the variation in time of one attribute of a selected object, such as a contour diagram visualizing the heights for multiple points in time. The third visualization level L2 provides the *details on the demand* by presenting a complete 3D rendering of the one object at a selected point in time. In the following paragraphs, we will describe these levels more in detail.

L0: Scene overview level Since 4D data is usually rather complex, it is hard to explore and visualize such data. Particularly, in on-site scenarios, users should find interesting objects quickly despite being limited to a small screen size. Therefore it is important that

a visualization technique provides a good overview of the scene and guides the user to conspicuous objects. An abstract visualization that shows summary information without any detail supports users in finding these objects.

Golparvar-Fard et al. [41] use color codings on a per object-basis to visualize different measurements for a complete construction site such as completion and progress, criticality or cost in a Mixed Reality visualization. In their work, they overlay registered color coded BIM models over a previous captured camera image of a construction site. Such a visualization is adequate for distant views, such as birds eye views or overviews, where multiple objects are visible.

We use such a per-object-abstraction for overview visualization on the *scene overview level*. Combined with filtering techniques, we use this visualization level as a starting point for further inspection on a per object-basis. As shown in Figure 7.10 (Top) physical objects, such as walls, are visualized using a color coding representing their current status. Additionally, measurements can be as textual annotations on the object. It is important to note, that this visualization level can only represent one single value to summarize the data in one view.

L1: Object time overview level The visualization of L0 provides a quick overview over the entire scene, but restricts inspecting a selected object in more detail. To understand the current status of an object in relation to earlier points in time and its geometry, we need a graphical representation showing information of multiple steps at once in reference to the object's geometry. To get an overview of the object's change over time, we visualize abstract information that summarizes the object's status for each point in time. This abstraction already conveys the evolution of the object's shape over time using outlines, contours or average diagrams, while retaining enough abstraction to show the whole time series. Different visual abstraction techniques enable for effective information presentation at this level. For example,

- Height Lines: representing different heights of an object at different times (compare Figure 7.11, Top Left).
- Geometric Completion: representing the completion at multiple points in time in reference to the object geometry using a color coding. For instance, a diagram showing the color coded completion (Figure 7.11, Top Right) or the color coded average completion (Figure 7.11, Bottom Right).
- Block diagram for average completion: block diagram showing the average completion in relation the geometry of the object of interest at multiple points in time (compare Figure 7.11, Bottom Left).

L2: Detail Level The detail visualization level allows for inspecting the detailed geometry and appearance of an earlier status by rendering the a 3D representation of the earlier status itself. In the overlay of the 3D data, the user can analyze the progress on the object in a detailed way. We can present the data at this level as pure 3D, as a an orthogonal snapshot of the 3D for reducing rendering times or by using the abstract blockworld representation described in Section 3.2.4 (Figure 7.12, Left). Furthermore, color-coded

Figure 7.12: Different representation of a detailed view on a selected point in time. Left) Block model textured using projective texture mapping. Right) Color-coded shadings of a wall geometry at different points in time.

shading of can support the visual discrimination between the status of objects at different points in time (Figure 7.12, Right).

7.5 Transitions between Visualization Levels

After introducing the visualization levels, the second challenge is the interactive transition between them. In this section, we introduce a set of interaction methods that allow the user to navigate between the visualization levels. The general idea of our visualization concept is that a user starts in the overview level and selects objects of interest for further inspection in the other levels. For example, the overview level displays a set of objects and their progress values. By filtering the objects based on their progress, only objects with a certain amount of progress are shown. Afterwards, the user can select or physically move to an object of interest (e.g., with a certain amount progress). We connect all levels by interactive view manipulation techniques, such as Focus&Context providing visual continuity when drilling down for more details.

In order to use these techniques, we have to define context and detail information regarding our visualization levels. The definition depends on the particular transition between visualization levels, since the lower level corresponds to lower level of detail. For the transition between L0 and L1, L0 contains overview information and L1 contains detail information. In contrast, for the transition between L1 and L2, L1 describes the context and L0 is in detail.

Managing contextual and focused information has a long tradition in information visualization. We can adapt these traditional methods for our needs to move between the 4D visualization levels. As described by Cockburn et al. [20], Overview&Detail techniques separate context and detail information temporally or in space by applying, for instance, zooming techniques or by presenting the data in separated views. Focus&Context techniques integrate this information in one single view and present context and focus information together.

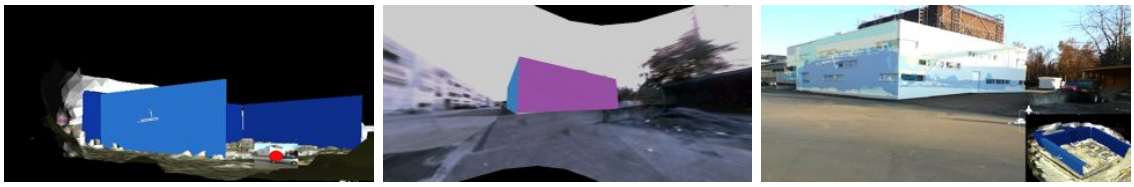


Figure 7.13: Overview&Detail techniques. Left) Zooming out to a virtual overview with abstract visualization for multiple objects. Middle) Using an extended field of view for providing an scene overview. Right) Birds eye view in WIM with abstract visualization for multiple objects.

7.5.1 Overview&Detail

The first level of abstraction is used to present an overview of the environment. Due to the highly abstract character of this level, less information is presented and thus it is possible to inspect more objects at once. In contrast, the second level of abstraction provides more detail that may easily clutter an overview visualization. The falloff in detail of the visualization levels makes the transition between the first and the second level a perfect candidate for Overview&Detail techniques. However, providing techniques for switching between overview and detail visualization is crucial for the acceptance of any tool for visual analysis. Thus, we provide two common techniques for transitions between overview and detail visualizations. A zooming interface and a WIM presentation allow for a comprehensive transition between the first and the second 4D visualization level.

Zooming The zooming interface allows the user to zoom out of his current view. For this purpose, he either leaves the AR view and zooms out to a virtual overview (Figure 7.13, Left) or she zooms out of her current position using an extend view visualization [56]. The extended view visualization still provides the AR overlay, but based on a simulated wider angle of view, similar to the egocentric panoramic view of Mulloni et al. [93]. The wide angle of view allows the user to inspect objects that are not in her current field of view. Based on the amount of zooming this provides an overview of the scene where the user can inspect multiple objects at once. If the distance exceeds a defined threshold, we assume that the interest on detailed information vanishes. Accordingly, we switch to the overview level L0 and present the more abstract representations of the scene objects (Figure 7.13, Left). Thus, the user get no only a spatial overview but also a overview in terms of provided information.

World in Miniature In the WIM visualization, a birds eye view allows the user to get an overview of the scene and the objects of interest, while on the same time showing the information of the second abstraction level in the main view. By selecting a point of interest the user can determine the look-at vector of the camera and the camera is translated by using the up vector of the scene (Figure 7.13, Right).

7.5.2 Focus&Context

Since a full scale of both L1 and L2 visualizations is critical for their interpretation, spatial Overview&Detail techniques are not suitable to switch between these levels. Also,

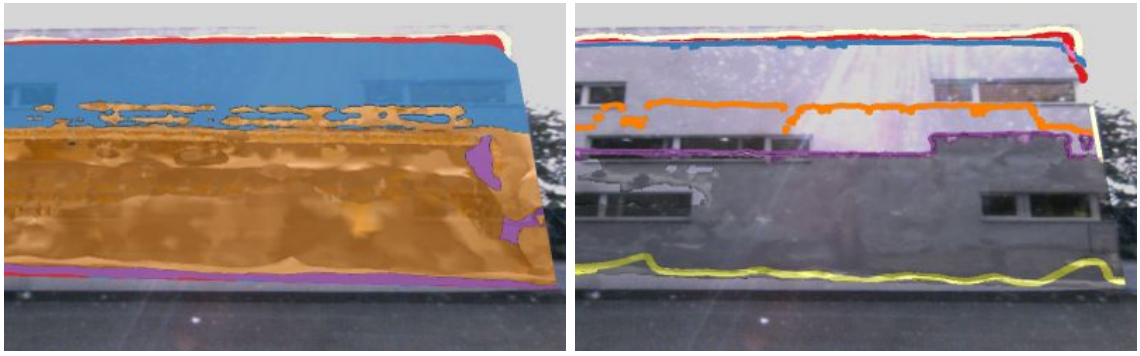


Figure 7.14: Transitions between visualization levels with overlay. Abstract representations such as Left) geometric completion and Right) height-lines are combined with the object's detailed view to support the transition between both.

temporal Overview&Detail are not suitable, since they demand a rather high amount of workload due to change blindness [117]. Therefore, we use Focus&Context techniques to combine the visualizations in a single view and within correct scale.

Overlay A simple technique to present both, abstract information A and concrete information T in one view is an overlay using simple blending (Figure 7.14).

Slider In contrast, 2D slider allow to separate the focus and the context region, similar to the techniques in Section 7.2. In this case the 2D slider operates in object space, instead of working in image space as used for the information filtering 2D slider. The slider allows the visualization of the time overview on one side and the detail object information on the other side (Figure 7.15, Left).

Magic lens A magic lens is an interface technique which allows to present the information inside the lens in a different style than the information outside [20]. We adapt this concept in such a way that the area inside the magic lens displays virtual representation from a former point in time, while its context area shows the abstract information about multiple points in time (Figure 7.15).

Distorted View In the research field of information visualization, view distortion techniques are typically used to enlarge a focus area and reduce the context area of a visualization. The *Bifocal-Display* visualization is for instance used to provide detailed information of a focus area and less information of a context area by distorting it [119]. We adapted this technique to embed detailed object information of one point in time while showing a compressed overview of the relative changes in the same view (Figure 7.16). This provides a detailed view of a selected point in time, while still visualizing relative changes in one view and allowing to select different points in time.

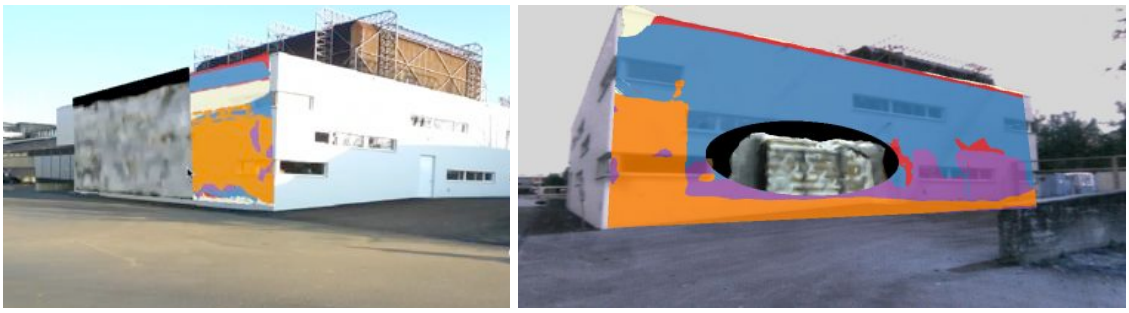


Figure 7.15: Transitions between visualization levels with a 2D Slider and a Magic Lens. Left) Side-by-side visualization of time overview information and object detail. Right) The magic lens area provides a detailed view on a scaffold used to built the wall.

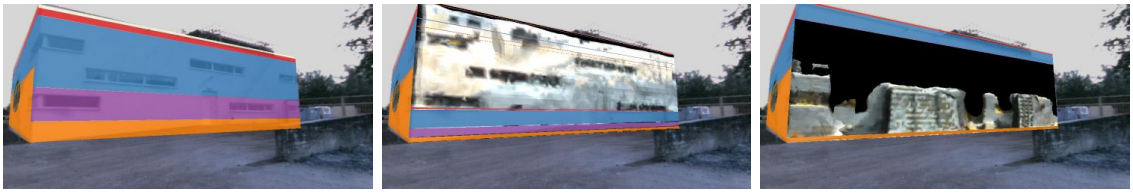


Figure 7.16: Transitions between visualization levels with distorted view. Left) Average Completion Diagram. Middle and Right) By selecting a point in time within the completion diagram the user gets a detailed visualization of the wall at the point point in time he selected. To provide still the abstract information, the completion diagram is distorted and presented in the same view.

7.6 Implementation

We face two challenges in order to implement 4D overview and detail visualizations in AR. Firstly, registered 3D data of multiple points in time should be available. There are different sources that can provide 4D data of certain scenes such as laser scans, image-based reconstructions from terrestrial or aerial views, or manually created 3D data for different states of a scene. Secondly, a segmentation of the scene into regions or objects of interest. Since complex 3D scenes can change heavily over time, our visualizations are focused on selected regions of interests. Such data come from *Building Information Modeling (BIM)* systems, CAD plans or manually defined regions of interest and can be represented by 3D boxes. Based on this input data the 3D representation for each point in time and for each region of interest can be computed. Finally, the 4D overview and detail visualization for changing scenes can be computed and overlaid in an AR view.

7.6.1 Extracting Time-oriented Data

We compute abstract time-oriented data for each object in the scene. For this purpose, we extract the corresponding mesh data for each object by (1) determining all triangles inside the region of interest and (2) by projecting all triangles inside the region of interest onto its visible planes (Figure 7.17, Top and Middle). These *time-oriented snapshots* of each object are stored together with the object itself and used for later processing to generate abstract representation of the object (Figure 7.17, Bottom).

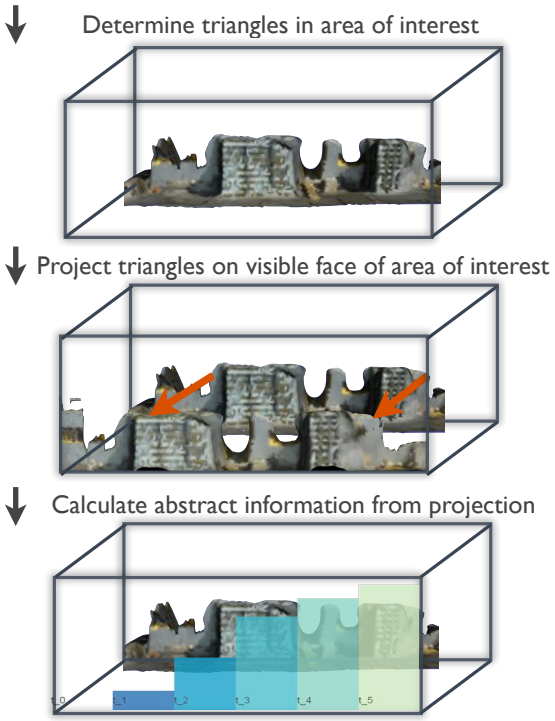


Figure 7.17: Extract time oriented data for areas of interest. First, triangles are tested, if they are in the region of interest. Second, projecting the determined triangles onto visible face of region of interest. Third calculate abstract representations from the projection.

To determine all triangles inside the region of interest, we perform an intersection test between all vertices of each face and the 3D box representing the region of interests. If at least one intersection test is positive, the triangle is considered to be a part of the region of interest. This test has to be performed for all triangles of the mesh of each point in time for each region of interest. To compute abstract representations such as the height of a region of interest at certain time, we will project all triangles inside the area onto its visible faces using an orthographic projection. This simplifies the calculation of abstract parameters by using a 2.5D representation of the object. We capture the result of the rendering in an off-screenbuffer and save it with the region of interest for each point in time.

7.6.2 Implementation of Visualization Levels

After extracting time-oriented data, we can visualize the data using different representations. These representations vary in the level of abstraction as described in Section 7.3.

Implementation of Scene Overview The scene overview visualization level L0 is used to present overview data of multiple objects. To avoid clutter due to the presented infor-

mation of each object is abstracted to a single measurement. This abstract information can come from external sources, such as cost or schedule information from construction site documentation or can be computed from the 4D data itself. For instance, we can compute the completion index of an object or its progress. If the region of interest describes as-planned data, completion is defined as the area of the region of interest that is already filled. Progress can be computed as the amount of completion over time.

In order to compute the degree of completion, we binarize the time-snapshots from Section 7.6.1 to determine the occupied pixel on the surface of the region of interest. The completion is then computed as the area of occupied pixels *Occupied* divided by the projection surface of the region of interest $InterestArea$

$$Completion(t) = Occupied(t)/InterestArea. \tag{7.1}$$

After, we can compute the progress as

$$Progress(t) = Completion(t) - Completion(t-1). \tag{7.2}$$

These measurements are presented by color coding an object or by textual annotations. Instead of using our measurements for completion and progress, we can integrate other values such as criticality, cost or time deviation as well, if they are provided by external sources.

Implementation of Object Time Overview The main purpose of the time overview level is to present multiple points in time. In order to be able to visualized multiple points in time, we have to extract abstract information. By using the extracted time-oriented data, we can compute abstract time-oriented representations such as height lines, geometric completion and block diagrams showing the average height.

To compute height lines, we binarize a time-shot image and apply an edge detector. For each x-value we determine the edge with the highest y-value and use this information to show the current height of the region of interest.

For computing geometric completion diagrams G, we determine the difference area of *Occupied* between two points in time t and $t-1$. The difference area shows the amount of change and is visualized by using the color code of the corresponding point in time t

$$G = ColorCode(Occupied(t) - Occupied(t-1)). \tag{7.3}$$

The geometric average completion diagram for multiple points in time is computed by using Equation 7.1. To visualize this value on the surface of the region of interest, we color all its pixels that are below $Completion(t)$ with the corresponding color value of the point in time t. The block diagram is computed in a similar way, average completions $Completion(t)$ are drawn color coded while dividing the x-axis of the object into the number of time steps.

For the 4D AR visualization, we superimpose the physical world scene with all the computed 4D information as shown in Figure 7.11.

Table 7.1: Overview of interactive transition modes with their corresponding interaction methods.

		Interaction Methods					
		Change Camera View	Filtering	Select Time-Visualization Method	Select Objects	Select Focus Points on Object	Select Points in Time
Overview &Detail	WIM	GUI Slider	GUI Spinbox, GUI Combobox	-	Mouse Input on Scene	-	GUI Slider
	Zooming	GUI Slider	GUI Spinbox , GUI Combobox	-	Mouse Input on Scene	-	GUI Slider
Focus &Context	Overlay	-	-	GUI Combobox	Mouse Input on Scene	Mouse Input on Object	Mouse Input on Object Overview, Slider
	Magic Lens	-	-	GUI Combobox	Mouse Input on Scene	Mouse Input on Object	Mouse Input on Object Overview, Slider
	Distorted View	-	-	GUI Combobox	Mouse Input on Scene	Mouse Input on Object	Mouse Input on Object Overview, Slider

Implementation of Object Detail The last visualization level allows the user to retrieve detailed information about the object geometry and appearance at a single point in time. The implementation of the object detail level is straight forward, since it shows the reconstructed data, the blockworld model, or the time-oriented snapshot of the region of interest for a selected point in time.

7.6.3 Implementation of Interaction Techniques

Our 4D visualization approach enables the user to interactively navigate between the different visualization levels. More precisely, one can navigate between the scene overview presenting abstract summary information, such as average completion, over to the object time overview visualization of the evolution of a parameter to finally a detailed view presenting the actual object's appearance at a point in time. As interaction input we use a combination of mouse pointer input in the AR view and a 2D GUI. Table 7.1 gives an overview of the interaction methods.

Transition from L0 Scene Overview to L1 Object Time Overview Overview&Detail techniques provide the user with an overview of the complete scene.

For this purpose, the user selects either the zooming or WIM mode in the 2D GUI. For controlling the zooming itself, the user manipulates a slider in the 2D GUI (Table 7.1).

In the WIM mode the scene overview level L0 is displayed in the same view as the AR visualization. For this purpose, we apply an offscreen rendering to compute a virtual birds eye view of the scene (Figure 7.13). For the zooming mode, we provide two different methods that provide an overview of the scene:

1. An VR zooming mode, where the user translates the virtual camera along the current look-at vector by using the slider.

2. An AR zooming mode where the slider is used to control the angle of view of the camera rendering virtual and physical scene. By using panoramic images of the physical environment, this method allows the user to increase his field of view of the virtual scene while preserving the context to physical world.

For both zooming methods, abstract scene information such as average completion is displayed, if the distance of the camera to the actual AR camera or the field of view is above a defined threshold. In this transition mode the user can configure the visualization by mostly using the 2D GUI, for instance the type of attribute that is displayed as abstract scene information can be selected or a threshold for filtering the displayed objects.

Transition from L1 Object Time Overview to L2 Object Detail For the transition between level L1 that shows multiple points in time for one object and level 2 that provides a detail view of an object, we apply Focus&Context techniques (Section 7.6.2). Focus&Context is used to combine abstract time-oriented values such as completion or progress with the concrete appearance of regions at certain points in time. We implemented a magic lens, 2D sliders, view distortions and overlays for this purpose.

If the user interactively selects points of interest in the time overview representation L1, the detail information of the select point in time is shown as Focus&Context visualization. In order to implement the selection mechanism for the points on the surface of the object of interest, we have to convert the mouse coordinates from the camera image space into the image plane on the surface of the object of interest. For this purpose, we transform the mouse coordinates into 3D world points, and by using the object of interest's transformation matrix, we map the mouse coordinates into the local coordinate system of image plane of the object of interest. These coordinates are then used for interacting with the object time overview level to get a more detailed view on the data.

For instance for the magic lens rendering, the user selects a point on the object of interest by using the mouse or a pen. After mapping this point into the image plane of the abstract information, a rectangular or circular area is drawn into the blending mask M. We use this mask then to combine the abstract information A with the detail information of the selected point in time $T[t]$ in a blending.

To compute the distorted view visualization, the object time overview information is scaled down and translated in such a way that it is only occluding a minimal part of the detailed object information (Figure 7.18, Top and Middle). The scale direction is computed from the direction of the time vector in the visualization. After scaling, the scaled abstract information is distributed on the visualization area to gain space for

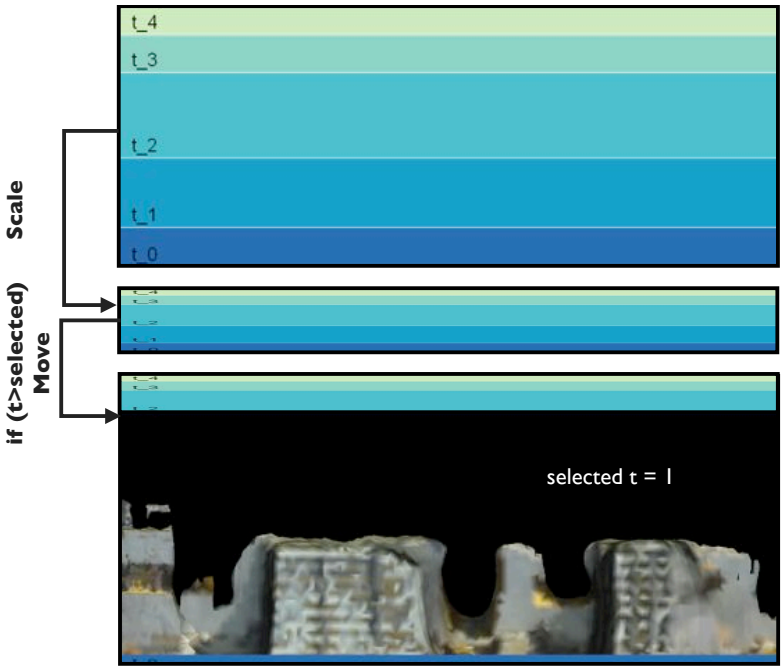


Figure 7.18: Computation of distorted view. The abstract information (average completion) is scaled down in such a way that it does not occlude to much of the detailed information. Afterwards, all abstract information representing a time larger than the selected point in time are moved to the beginning of the diagram.

the detail information. For this purpose, abstract information for times larger than the selected point in time is moved to the beginning of the diagram (Figure 7.18, Bottom). For instance applied for the color coded completion, the user can select a color of interest and automatically, the color coded areas are reduced in size and transformed in such a way that the detail information of the selected point in time can be rendered on the geometrically correct location (Figure 7.18 and Figure 7.16).

7.7 Application: Construction Site Monitoring

A 4D representation usually consist of 3D reconstructions accurately generated at relevant points in time. These timestamps can be defined milestones within the executed project management, but also points that are taken from a regular basis. Having 3D models at different points in time provides the possibility to extensively observe the construction site from arbitrary, very interactive, viewing positions. Additionally, the 4D scene information serves as an important referencing platform which can be augmented with any kind of data like individual laser scans or single shots taken by a supervisor.

In order to visualize the 4D data in context to physical world, we apply the 4D visualization concept. This allows for the visualization of geometric changes of construction sites directly on-site. The main requirements for the visualization is (1) to preserve context of

Figure 7.19: Construction site inspection. Left) Color coding for different points in time. Middle Left) The time step block diagram shows that for $t = 2$ the completion was regressive. Middle Right and Right) Accordingly, the user may want to inspect this point in time further by using the interactive tool to combine abstract information with reconstruction data.

changing environment and (2) see how the construction changed over time. It is important to note, that not every geometric change is relevant to the documentation process. For instance, moving machines cause massive geometric changes, but these changes are not semantically meaningful ones. (1) is achieved by using an on-site AR visualization, (2) by implementing a visualization based on the overview and detail visualization for 4D data. For a construction site manager it is much easier to reproduce and understand progress, and possible mistakes if relevant information is directly linked to the physical construction site. Instead of overlaying only overview information as done in the work of Golparvar-Fard et al. [40], our approach allows the detailed inspection of previous points in time as well.

Data We first need as-built 4D models from digital imagery and second the preparation of the synthetic models, extracted from CAD data or BIMs representing a the as-planned status. In Section 3.2.3, we described how to obtain 3D reconstruction information by using aerial vision. Flying over a construction site with a MAV and capturing a set of meaningful camera images on a regular basis creates 4D datasets. The as-built data sets are given by CAD or BIM data and converted into a 3D representation as described in Section 3.2.2.

On-site Visualization For the on-site 4D visualization of the progress on a construction site, we use the mobile setup described in Section 3.3. By presenting progress and completion information about the site, construction site staff can inspect the progress of the site right in place. This allows them to search for problematic points or bottlenecks. Figure 7.19 shows one of the scenarios where a 4D visualization can be helpful to support a supervisor of a construction site to inspect conspicuous points in time. The time step block diagram on the left shows that for $t = 2$ the completion was regressive. By inspecting the 3D reconstruction information in the detail view, the user noticed that a scaffolding was replaced by a concrete wall. Thus, the reason for the regression was not a mistake but as-planned in the building process.

7.8 Summary

The aim of this chapter was to contribute to the research of visualization techniques for complex data and in particular time-oriented data in AR. The fast advancement of 3D reconstruction techniques provides tools for creating more and more 3D data. This also raises new questions, such as how to visualize this kind of data. In this chapter, we showed the possibilities given by traditional information filtering techniques for avoiding information clutter. We discussed how a combination of information filtering techniques and 2D and 3D Focus&Context tools allows the user to specify regions interest that display complex 3D data. These traditional tools provide methods for avoiding information clutter and giving users means for comparing a limited number of different 3D data sets.

However, traditional information filtering methods have some limitations. Firstly, they are not designed to display more than two data sets. Secondly, these kinds of visualization are often subject to self occlusions, where a part of the virtual information occludes important parts. In order to approach these problems, in Section 7.3, we introduced a visualization concept that uses information abstraction to display a higher number of 3D datasets in one AR view.

By introducing three visualization levels that differ in amount of detail and abstraction, we enable the user to browse time-oriented data in a comprehensible way.

These levels implement the *Information Seeking Mantra* of Shneiderman [116] and provide an overview of multiple objects, over multiple points in time, while still providing detailed information about an object of interest for a select point in time. The visualization levels are supported by interactive transitions among each other. These transition techniques provide the important geometric reference between abstract overview visualization levels and physical world object of interest.

We applied the concept for a complex dynamic scenario: the documentation of construction site progress. Discussions with partner companies showed that there is high interest in the visualization of 4D for construction site progress documentation. For the visualization of construction site progress, we used construction-site related measurements such as completion or progress in order to combine abstract information with concrete information such as the objects appearance and geometry.

To evaluate the applicability of the visualization of 4D data in place in such a complex scenario, we plan to conduct a user study with expert users from construction companies. Another research question that is open for future research refers to the shape of objects of interest. In the moment, our technique focuses on the visualization of planar objects. For construction sites this limitation is reasonable, since most of the construction elements are usually planar or nearly planar objects such as walls, roofs and scaffolds. But if we want to apply the visualization technique for other use cases as well, we have to extend this approach for other geometric shapes. Furthermore, the visualization strongly depends on the quality of the data which usually differs. This is in particular interesting when it comes to varying providers of data. Thus it is also worth to spend more research into how to work with different levels of data quality.

Chapter 8

Conclusion

Contents

8.1 Summary of Results

The main goal of this thesis was to identify and address comprehension problems that appear when embedding virtual information to the user's view in outdoor environments. In particular, we focused on problems that occur when visualizing professional data in professional *Augmented Reality (AR)* applications for *Architecture, Construction and Engineering (ACE)* industries.

In this context, we faced the following challenges that refrain users from understanding the embedded information:

- Insufficient scene integration
- Insufficient depth cues
- Information clutter

In order to gain more insight into this topic, we discussed how other researchers approached these problems in different application fields. For this purpose, we developed a taxonomy that allows classifying the techniques. The classification helped us to find differences and similarities within the existing techniques and to find gaps that were not addressed so far.

Based on this classification, we developed an adapted version of the traditional *Information Visualization* pipeline. The main requirements for the adapted visualization pipeline is that it 1) reflects the characteristics of AR visualization in general, such as the need for context and registration information, and 2) is adjustable to the different existing visualization techniques in AR. By providing different versions of this pipeline based on

the classification of visualization techniques in AR, we allow for the fast identification of required data sources and data flows according to a specific comprehension problem or a specific visualization technique. For instance, if an application designer knows that she wants to address the problem of insufficient scene integration, she can use the classification to identify which data source she needs and what result can be achieved using the corresponding techniques. Moreover, the classification helped us to identify open issues in outdoor AR visualization that can not be addressed with the existing techniques.

We first learned that the scene integration of virtual content in outdoor AR environments has only been addressed in a limited scale. In these outdoor environments, the extent of world knowledge is limited. Often nothing more but a camera image and a semantic depth order of virtual and physical objects is known. An application example for this problem is the visualization of subsurface infrastructure in urban scenes. In this case, only the information that all virtual objects are located under the ground is given.

Nevertheless, for subsurface visualization, we need more information about the physical environment, since a normal occlusion culling would hide all virtual objects of interest. In contrast, if the virtual subsurface objects are naïvly overlaid onto the camera image, they will be perceived as floating over the ground, since important depth cues are missing. We end up with an incomplete scene integration, which is a big problem for comprehensions in AR environments.

In order to convincingly reveal virtual subsurface structure, it is important to provide the user with a set of meaningful depth cues. Previous work used edges as meaningful depth cues in outdoor environments. However, we made the experience that this information is not always sufficient, in particular when working in urban scenarios. In street scenes, we often experience prominent visual elements such as street markings or colored elements that were not considered by the existing approaches. Edges seem to be not able to provide enough cues for providing a correct scene integration in these scenes.

To address this problem, we developed a method that extracts a set of importance measurements from camera imagery and uses them as occlusion cues. These measurements include salient regions, edges and texture details. Additionally, in order to maintain perceptual grouping, we analyze these characteristics on a per-superpixel basis. In image regions that were identified to contain not enough important information, the method adds artificial occlusion cues, such as stipples and hatches. With a user study, we confirmed our initial hypothesis **H1** that a seamless integration of virtual and physical information can be achieved by extracting and preserving existing visual cues from the environment. As assumed, the study showed that for these urban scenes, edges are not sufficient for providing enough visual cues to achieve a convincing scene integration of subsurface objects. In contrast, our method is able to provide enough occlusion cues so that users are able to understand the depth order of virtual objects in the physical environment.

Needless to say, not for all applications the depth order is known. For instance, when visualizing as-planned infrastructure objects, such as new lamps or projected buildings, the depth order between the virtual objects and the physical objects in the environment is often unknown. Advantageously, this information is sparsely available from public and private GIS databases. These databases provide already a huge amount of spatial information about infrastructure objects in urban environments. Their drawback is that the data are often too sparse to provide correct occlusion cues or shadows. This drawback so far

prevented from widely using GIS data as input for visualization techniques in outdoor AR.

To address this problem, we developed a method that combines sparse depth information from GIS databases with accurately registered camera imagery provided by an AR setup. Based on these camera images, we can compute a segmentation that uses the projection of the sparse representations into the image space. The result of the segmentation is a 2D region in the camera image. By extending this 2D information with the depth information from the GIS database, we are able to compute a 2.5D model of the physical world object. Applying a billboard rendering using this 2.5D model, we can correct occlusions or shadows.

These techniques confirm our second hypothesis **H2**, assuming that we can compute visual cues automatically from camera images, GIS data or a combination of both.

At this point of the work, we are able to achieve a convincing integration of virtual content into the physical outdoor environment. However, these techniques often provide only ordinal depth information. This is not enough for applications that require the user to estimate the depth in a more exact way. For instance, for construction workers it is often important to determine the depth of subsurface objects to adjust the digger or to determine the location on the ground were they should start to dig. For these applications, ordinal depth cues are not enough. We need to provide more than just natural depth cues.

For this purpose, we developed a set of additional graphical hints, termed virtual cues. These cues aim to support the user in estimating the depth or the height of virtual objects. They focus on restoring the pictorial depth cue *height in visual field*, as this cue is able to provide absolute depth measurements for objects that are connected to the ground. These additional hints comprise virtual excavations with virtual scale textures, connection lines, artificial shadows and virtual junctions. We applied them for supporting the depth estimation of subsurface objects and floating objects. The biggest challenge for creating these kind of virtual cues is how to access and convert the georeferenced data into such a graphical representation.

The challenge is increased, when working with GIS databases that require flexibility for data modifications. To address this problem, we further proposed an approach that allows for maintaining consistency between a set of graphical representation and data in a geospatial database. We achieved this, by storing the graphical representations and pure data in different data levels. A transmission layer builds the connection between both level and updates them according to changes. We tested this approach with different GIS databases from civil engineering companies. In addition to these tests, we conducted interviews with expert users from these companies. The results showed a positive feedback on the additional graphical hints.

Based on these techniques, we can confirm our third hypothesis **H3**, assuming that we can use georeferenced data to automatically create additional virtual pictorial cues for conveying spatial relationships.

Finally, we addressed the problem of information clutter when visualizing complex data. In the discussion of previous work, we showed that some approaches address information clutter by using Focus&Context tools. Within the scope of this thesis, we applied these methods for visualizing complex time-oriented data representing progress on construction sites. We showed that these methods can be used for visualizing a small number of different data sets (confirming our forth hypothesis **H4**).

Nevertheless, for the visualization of progress on construction sites it is important to display a higher number of different datasets in one view. This allows one to search for sources of constructional flaws or to monitor the progress over a larger time period. In order to visualize multiple 3D datasets in one AR view, we introduced a 4D visualization concept. This approach combines abstraction with Focus&Context techniques and allows inspecting and comparing different points in time in one view while selecting points of interest for further inspections. This confirms our fifth hypothesis **H5**, assuming that it is possible to automatically abstract complex data to reduce information clutter in an AR visualization.

The techniques developed in this thesis have the main goal of advancing the usage of AR in professional outdoor applications from ACE industries. We were able to show that it is possible to create comprehensible and convincing visualizations supporting users in different applications, such as surveying, planning, construction site monitoring and navigating aerial vehicles. However, the potential of these visualization techniques depends on the available technologies, so far they often require high-quality sensors to provide convincing overlays in unknown environments. Our intention was to demonstrate the potential of AR for these applications by providing adequate visualization techniques in combination with accurate registration techniques.

8.2 Lessons Learned

The findings of this thesis provide insights how information can successfully be integrated into an AR overlay. We learned that AR in outdoor environments requires visualization techniques that adapts to different conditions, such as

- The current physical environment,
- The requirements of the user,
- The available data.

If the current physical environment has a large amount of important structures, these structures should not get lost in the final composition. We showed this in Chapter 4, by comparing visualization techniques that preserve a large amount of important image structures and visualization techniques that do not. Our research showed, it is more likely to achieve a convincing scene integration if these structures are preserved.

In Chapter 6, we showed that depending on the task, users often need additional graphical hints, in particular, if a task requires to estimate depth in an relative or absolute way.

We learned that complex data requires specific filtering and abstraction operation to provide comprehensible visualizations (Chapter 7). In particular, for visualizing multiple datasets in one view.

It is important to note, that these findings are similar to the requirements that exists for adaptive interfaces [79] and adaptive visualization techniques in *Information Visualization*. For instance, Grawemeyer described that visualization techniques in Information Visualization can adapt to the user requirements and the available data itself [44]. The

adaption to the physical environment was also discussed by researchers, but except for location-based systems, is rather disregarded [18].

In AR the physical environment has a larger influence, since the visualization itself has a direct relationship to the environment. We discussed the importance of the interplay between environment and visualization in terms of comprehension in AR visualization in an earlier work [66]. In this earlier work, we discussed that in order to create convincing visualizations in AR, the virtual as well as the physical world information has to be analyzed and adapted to fit each other. While back then, we limited the data sources for this analysis to camera images and 3D models, within the work of this thesis we were able to extend this list. We learned that the adaption of the visualization is possible by:

- Analyzing the physical context using geospatial data
- Analyzing the physical context using camera imagery
- Deriving additional graphical hints from the geospatial data
- Reducing information clutter using information filtering and abstraction

It is, in particular, interesting to understand, how much information about our physical environment, is already available in public and commercial database. GIS databases already provide a lot of information that can be used to adapt visualizations in outdoor environments. Furthermore, the wide availability of 360° panoramas in combination with depth information given by laser scans such as captured by Google within Google Streetview [1] provides a lot of additional knowledge about our psychical environment, at least in urban outdoor environments. Within this thesis, we only used a small amount of this available information, but the availability of this data opens new possibilities for adapting visualizations in relationship to the actual environment.

There are some more specific insights from our work that can be helpful for further research in visualization for AR. We found that accurate tracking methods are always a prerequisite for achieving convincing visualizations. If the registration is not able to providing an accurate overlay, the visualization technique has to compensate for the tracking errors [27, 74].

Another experience that we had during our research is that collaborations with industrial partners provide challenging data for visualization. A lot of visualization problems only appear when working with real world data. In fact, there is a lot of effort required to access this kind of data, but the realism they provide helps to show the usefulness of the developed visualization techniques.

8.3 Future Work

We can apply our findings about adaptive visualizations in AR to derive guidelines for more general interfaces that build up on a combination of virtual and physical information. This especially interesting for new generations of display systems focusing more on directly embedding information into our physical surroundings, instead of showing it on a closed

[1] `http://maps.google.de/intl/de/help/maps/streetview/`

display. Examples for such interface are Google Glass[2], head-up display or even more conceptual interfaces like Sixth Sense [92]. For these kind of interfaces, sophisticated technologies that provide intelligent ways of integrating virtual content into our physical environment become more and more important.

Maes et al. summarizes interfaces that integrate virtual information into our physical environment with the term *Fluid Interfaces*[3]. In contrast to AR, they do not necessarily require a spatial registration of virtual information within the physical environment. Examples are augmentations of arbitrary objects in our environment with additional information using projector-based interfaces, *Head-mounted Display (HMD)* devices, head-up display and off-the-shelf mobile devices. Nevertheless, there is a set of problems that arise when using this kind of technology in our daily life, such as:

- Virtual information occludes important physical world information (HMDs, head-up displays, projectors)
- Users focus too much on virtual information and miss important physical world information (HMDs, mobile displays, head-up display)
- Too much information will lead to information clutter

To address these problems, we can use the findings from this thesis by adapting them to this group of more general interfaces. For instance, to avoid that virtual information occludes too much important physical world information, similar methods like the preservation techniques from Chapter 4 could be applied. The main challenge is the same as for the X-Ray visualization: How to find a good compromise between visibility of the physical environment and the virtual information?

Additional graphical hints about important physical world objects can guide the user to pay attention to these objects. For instance, if the user is reading an online newspaper on his mobile phone, a graphical hint can indicate the distances to street crossings helping to avoid dangerous situations with cars.

Furthermore, for the integration of complex information into these kinds of interfaces, methods that avoid information clutter, such as information filtering and abstraction can help to improve comprehensibility.

[2] `http://www.google.com/glass`
[3] `http://fluid.media.mit.edu`

Appendix A

Acronyms

List of Acronyms

ACE	*Architecture, Construction and Engineering*
AR	*Augmented Reality*
BIM	*Building Information Modeling*
DoF	*Degrees of Freedom*
DoG	*Difference of Gaussian*
DTM	*Digital Terrain Model*
FPS	*Frames per Second*
GIS	*Geographic Information Systems*
GML	*Geography Markup Language*
GPS	*Global Positioning System*
HCI	*Human Computer Interaction*
HMD	*Head-mounted Display*
HVS	*Human Visual System*
IMU	*Inertial Measurement Unit*
MAV	*Micro Aerial Vehicle*
NTRIP	*Networked Transport of RTCM via Internet Protocol*
OST	*Optical-See-Through*
RTCM	*Radio Technical Commission for Maritime Services*
RTK	*Real-time Kinematics*
SfM	*Structure from Motion*
UAV	*Unmanned Aerial Vehicle*
VR	*Virtual Reality*
VST	*Video-See-Through*

Appendix B

Survey

Subject Number _____

Instructions

In this user study you will have to observe a set of Augmented Reality scenes on a display. These scenes show virtual objects like red pipes and spheres in a real urban environment (roads, park, walkway). All virtual objects in one scene are either positioned at subsurface or overground level.
Your task is to observe each scene for a short time and

1) Answer if you perceived the virtual objects at subsurface or overground level. Please use the keyboard for your rating.
2) Draw an outline of all virtual objects in the scene. To outline one object, you create a polygon by clicking edge points of the object. To complete one object, press the right mouse button. Wrong polygon points can be deleted by pressing **'r'**, this will remove the last created outline points.

We will repeat this task for 12 scenes and afterwards there is a short break. In the break, we will ask you questions about your experience with the presented scenes with a questionnaire. You will be asked how you would rate the visualization technique used for the 12 presented scenes. After finishing the questionnaire, we will show 12 other scenes with a different visualization technique and you have to perform the same tasks. There are 3 different visualization techniques in total. After finishing all scenes, there is a final questionnaire that asks you to order the visualization techniques according to your preference.

This study will take approximately 30 min.

Thanks for your time!

Please circle the appropriate answer or fill in the spaces provided:

Gender: M / F

Age: _____

Occupation: _____

Vision Problems (*normal or corrected to normal vision*):

How familiar are you with using Augmented Reality Applications?

Not familiar at all	1	2	3	4	5	Very familiar

Visualization Technique "A" – Visualization "Subsurface"

A_A: The subsurface visualization using the X-Ray technique "A" was confusing.

I strongly disagree								I strongly agree

A_B: The subsurface location of virtual objects in the scene was hard to understand using visualization technique "A" .

I strongly disagree								I strongly agree

A_C: The shape of red virtual objects was complicated to understand during using visualization technique "A".

I strongly disagree								I strongly agree

Visualization Technique ”E” – Visualization “Subsurface”

E_A: The subsurface visualization using the X-Ray technique “E” was confusing.

I strongly disagree								I strongly agree

E_B: The subsurface location of virtual objects in the scene was hard to understand using visualization technique “E” .

I strongly disagree								I strongly agree

E_C: The shape of red virtual objects was complicated to understand during using visualization technique “E”.

I strongly disagree								I strongly agree

Visualization Technique "I" – Visualization "Subsurface"

I_A: The subsurface visualization using the X-Ray technique "I" was confusing.

I strongly disagree								I strongly agree

I_B: The subsurface location of virtual objects in the scene was hard to understand using visualization technique "I" .

I strongly disagree								I strongly agree

I_C: The shape of red virtual objects was complicated to understand during using visualization technique "I".

I strongly disagree								I strongly agree

Concluding Questions:

F_1: Please rank the visualization techniques (beginning with 1.=best) according to your preferences in terms of **depth perception**.

A: _____

E: _____

I: _____

F_2: Please rank the techniques according to your preferences in terms of **coherence** (Which technique integrated the virtual content convincingly into the camera image?).

A: _____

E: _____

I: _____

F_3: Please rank the visualization techniques (beginning with 1.=best) according to your preferences in terms of **general comprehension** (Which technique helped you the most to understand the spatial relationship of the presented content, but also the objects themselves?).

A: _____

E: _____

I: _____

Thanks for your participation!

Bibliography

[1] Achanta, R., Hemami, S., Francisco, E., and Suessstrunk, S. (2009). Frequency-tuned Salient Region Detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*.

[2] Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011). *Visualization of Time-Oriented Data*. Springer.

[3] Allen, M., Regenbrecht, H., and Abbott, M. (2011). Smart-phone augmented reality for public participation in urban planning. In *Proceedings of the 23rd Australian Conference on Computer-Human Interaction OzCHI '11*, pages 11–20, Cranberra, Australia. ACM Press.

[4] Arth, C. and Klopschitz, M. (2011). Real-time self-localization from panoramic images on mobile devices. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2011)*, pages 37–46.

[5] Avery, B. (2009). *X-Ray Vision for Mobile Outdoor Augmented Reality*. PhD thesis, University of South Australia.

[6] Avery, B., Sandor, C., and Thomas, B. H. (2009). Improving Spatial Perception for Augmented Reality X-Ray Vision. In *IEEE Virtual Reality Conference (VR 2009)*, pages 79–82. Ieee.

[7] Azuma, R. and Furmanski, C. (2006). Evaluating label placement for augmented reality view management. *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pages 66–75.

[8] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385.

[9] Bae, S., Agarwala, A., and Durand, F. (2010). Computational rephotography. *ACM Transactions on Graphics*, 29(3):1–15.

[10] Bane, R. and Hollerer, T. (2004). Interactive Tools for Virtual X-Ray Vision in Mobile Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2004)*, pages 231–239. IEEE.

[11] Bell, B., Feiner, S., and Höllerer, T. (2001). View management for virtual and augmented reality. In *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01*, page 101, New York, New York, USA. ACM Press.

[12] Bichlmeier, C., Kipot, M., Holdstock, S., Heining, S. M., Euler, E., and Navab, N. (2009). A Practical Approach for Intraoperative Contextual In-Situ Visualization. In *AMIARCS '09*.

[13] Bichlmeier, C., Wimmer, F., Sandro Michael, H., and Nassir, N. (2007). Contextual Anatomic Mimesis: Hybrid In-Situ Visualization Method for Improving Multi-Sensory

Depth Perception in Medical Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2007)*, pages 129–138.

[14] Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. (1993). Toolglass and magic lenses. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques - SIGGRAPH '93*, pages 73–80, New York, New York, USA. ACM Press.

[15] Breen, D. E., Whitaker, R. T., Rose, E., and Tuceryan, M. (1996). Interactive Occlusion and Automatic Object Placement for Augmented Reality. *Computer Graphics Forum*, 15(3):11–22.

[16] Bruckner, S., Grimm, S., Kanitsar, A., and Gröller, M. (2005). Illustrative context-preserving volume rendering. In *Proceedings of EUROVIS*, volume 1.

[17] Bruckner, S., Grimm, S., Kanitsar, A., and Gröller, M. E. (2006). Illustrative Context-Preserving Exploration of Volume Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1559–1569.

[18] Brusilovsky, P. (2001). Adaptive hypermedia. *User modeling and user-adapted interaction*, pages 87–110.

[19] Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698.

[20] Cockburn, A., Karlson, A., and Bederson, B. B. (2009). A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1—-2:31.

[21] Coffin, C. and Höllerer, T. (2006). Interactive perspective cut-away views for general 3D scenes. *3D User Interfaces (3DUI 2006)*, pages 25–28.

[22] Colomina, I., Blázquez, M., Molina, P., Parés, M., and Wis, M. (2008). Towards a new paradigm for high- resolution low-cost photogrammetry and remote sensing. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

[23] Cutting, J. E. (1997). How the eye measures reality and virtual reality. *Behavior Research Methods, Instruments, & Computers*, 29(1):27–36.

[24] Cutting, J. E. and Vishton, P. M. (1995). Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. *Handbook of perception and cognition*, 5:1–37.

[25] Davison, A. J. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *IEEE Ninth IEEE International Conference on Computer Vision*, pages 1403–1410. IEEE Computer Society.

[26] Dick, A. R., Torr, P. H. S., and Cipolla, R. (2004). Modelling and Interpretation of Architecture from Several Images. *Int. J. Comput. Vision*, 60(2):111–134.

[27] DiVerdi, S. and Hollerer, T. (2006). Image-space correction of AR registration errors using graphics hardware. In *IEEE Virtual Reality Conference (VR 2006)*.

[28] Elmqvist, N. and Tsigas, P. (2008). A taxonomy of 3D occlusion management for visualization. *IEEE transactions on visualization and computer graphics*, 14(5):1095–109.

[29] Feiner, S., MacIntyre, B., Hollerer, T., and Webster, A. (1997). A touring machine: prototyping 3D mobile augmented reality systems for exploring the urban environment. In *Digest of Papers. First International Symposium on Wearable Computers*, pages 74–81. IEEE Comput. Soc.

[30] Feiner, S. and Seligmann, D. (1992). Cutaways and ghosting: satisfying visibility constraints in dynamic 3D illustrations. *The Visual Computer*.

[31] Feiner, S. K. and Duncan Seligmann, D. (1992). Cutaways And Ghosting: Satisfying Visibility Constraints In Dynamic 3d Illustrations. *The Visual Computer*, 8:292–302.

[32] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181.

[33] Fischer, J., Bartz, D., and Straβer, W. (2005). Artistic reality. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '05*, page 155, New York, New York, USA. ACM Press.

[34] Fischer, J., Huhle, B., and Schilling, A. (2007). Using time-of-flight range data for occlusion handling in augmented reality. In *Proceedings of the 13th Eurographics conference on Virtual Environments (EGVE'07)*, pages 109–116.

[35] Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., and Pollefeys, M. (2012). Vision-based autonomous mapping and exploration using a quadrotor MAV. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4557–4564. Ieee.

[36] Furmanski, C., Azuma, R., and Daily, M. (2002). Augmented-Reality Visualizations Guided by Cognition: Perceptual Heuristics for Combining Visible and Obscured Information. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2002)*.

[37] Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Towards Internet-scale multi-view stereo. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 1434–1441. IEEE.

[38] Goldstein, E. B. (2001). *Sensation and Perception*. Wadsworth Publishing Company, 6 edition.

[39] Golparvar-Fard, M. and Pena-Mora, F. (2011). Monitoring changes of 3D building elements from unordered photo collections. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on Computer Vision for Remote Sensing of the Environment*, pages 249–256.

[40] Golparvar-Fard, M., Pena-Mora, F., and Savarese, S. (2009). D4AR - a 4 dimensional augmented reality model for automation construction progress monitoring data collection, processing and communication. *Journal of Information Technology in Construction*, 14:129–153.

[41] Golparvar-Fard, M., Sridharan, A., Lee, S. H., and Peña Mora, F. (2007). Visual representation of construction progress monitoring metrics on time-lapse photographs. In *Proc. Construction Management and Economics Conference*.

[42] Grasset, R., Duenser, A., Seichter, H., and Billinghurst, M. (2007). The mixed reality book. In *CHI '07 extended abstracts on Human factors in computing systems - CHI '07*, page 1953, New York, New York, USA. ACM Press.

[43] Grasset, R., Langlotz, T., Kalkofen, D., Tatzgern, M., and Schmalstieg, D. (2012). Image-Driven View Management for Augmented Reality Browsers. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2012)*.

[44] Grawemeyer, B. (2001). User adaptive information visualization. In *5th Human Centred Technology Postgraduate Workshop, University of Sussex, School of Cognitive and Computing Sciences (HCT-2001)*.

[45] Grubert, J., Langlotz, T., and Grasset, R. (2011). Augmented reality browser survey. Technical Report December, Graz University of Technology, Graz.

[46] Grzeszczuk, R., Kosecka, J., Vedantham, R., and Hile, H. (2009). Creating compact architectural models by geo-registering image collections. In *The 2009 IEEE International Workshop on 3D Digital Imaging and Modeling (3DIM 2009)*.

[47] Haber, R. and McNaab, D. (1990). Visualization idioms: a conceptual model for scientific visualization systems. *IEEE Computer Society Press*.

[48] Hägerstrand, T. (1970). What about people in Regional Science? *Papers of the Regional Science Association*, 24(1):6–21.

[49] Haller, M., Drab, S., and Hartmann, W. (2003). A real-time shadow approach for an augmented reality application using shadow volumes. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '03*, page 56, New York, New York, USA. ACM Press.

[50] Harris, R. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press.

[51] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[52] Hodges, E. R. S., editor (2003). *The Guild Handbook of Scientific Illustration*. John Wiley & Sons, Hoboken, NJ, 2\textsupe edition.

[53] Hoiem, D., Efros, A. A., and Hebert, M. (2005). Automatic photo pop-up. *ACM Transactions on Graphics (TOG)*, 24(3):577.

[54] Hoppe, C., Wendel, A., Zollmann, S., Pirker, K., Irschara, A., Bischof, H., and Kluckner, S. (2012). Photogrammetric Camera Network Design for Micro Aerial Vehicles. In *Computer Vision Winter Workshop*, Mala Nedelja, Slovenia.

[55] Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(April):629–642.

[56] Hwang, S., Jo, H., and Ryu, J.-h. (2010). EXMAR : EXpanded view of Mobile Augmented Reality. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2010)*, pages 3–4.

[57] Interrante, V., Fuchs, H., and Pizer, S. M. (1996). Illustrating transparent surfaces with curvature-directed strokes. In *IEEE Visualization*, pages 211–218, San Francisco, CA, United States. ACM.

[58] Irschara, A., Kaufmann, V., Klopschitz, M., Bischof, H., and Leberl, F. (2010). Towards fully automatic photogrammetric reconstruction using digital images taken from UAVs. In *Proc. International Society for Photogrammetry and Remote Sensing Symposium*.

[59] Javed, W., McDonnel, B., and Elmqvist, N. (2010). Graphical perception of multiple time series. *IEEE transactions on visualization and computer graphics*, 16(6):927–34.

[60] Judd, D. B. (1979). *Contributions to Color Science*. United States Govt Print. Off.

[61] Julier, S., Lanzagorta, M., Baillot, Y., Rosenblum, L., Feiner, S., Hollerer, T., and Sestito, S. (2000). Information filtering for mobile augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality ISAR 2000*, pages 3–11. IEEE COMPUTER SOC.

[62] Kalkofen, D. (2009). *Illustrative X-Ray Visualization in Augmented Reality Environments*. PhD thesis, Graz University of Technology.

[63] Kalkofen, D., Mendez, E., and Schmalstieg, D. (2007). Interactive Focus and Context Visualization for Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2007)*, pages 191–200.

[64] Kalkofen, D., Mendez, E., and Schmalstieg, D. (2009a). Comprehensible Visualization for Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):193–204.

[65] Kalkofen, D., Tatzgern, M., and Schmalstieg, D. (2009b). Explosion Diagrams in Augmented Reality. *IEEE Virtual Reality Conference (VR 2009)*, 0:71–78.

[66] Kalkofen, D., Zollman, S., Schall, G., Reitmayr, G., and Schmalstieg, D. (2009c). Adaptive Visualization in Outdoor AR Displays. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*.

[67] Kasahara, S., Niiyama, R., Heun, V., and Ishii, H. (2013). exTouch: spatially-aware embodied manipulation of actuated objects mediated by augmented reality. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*.

[68] Kato, H. and Billinghurst, M. (1999). Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. page 85.

[69] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proc. Symposium on Geometry Processing*, pages 61–70.

[70] Keil, J., Zoellner, M., Becker, M., Wientapper, F., Engelke, T., and Wuest, H. (2011). The House of Olbrich - An Augmented Reality tour through architectural history. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2010) - AMH*, pages 15–18. IEEE.

[71] Kim, H., Reitmayr, G., and Woo, W. (2012). IMAF: in situ indoor modeling and annotation framework on mobile phones. *Personal and Ubiquitous Computing*, 17(3):571–582.

[72] King, G. R., Piekarski, W., and Thomas, B. H. (2005). ARVino - Outdoor Augmented Reality Visualisation of Viticulture GIS Data. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2005)*, pages 52–55.

[73] Kiyokawa, K., Kurata, Y., and Ohno, H. (2000). An optical see-through display for mutual occlusion of real and virtual environments. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, pages 60–67. Ieee.

[74] Klein, G. and Drummond, T. (2004). Sensor Fusion and Occlusion Refinement for Tablet-Based AR. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 38–47. IEEE.

[75] Klein, G. and Murray, D. W. (2010). Simulating Low-Cost Cameras for Augmented Reality Compositing. *Visualization and Computer Graphics, IEEE Transactions on*, 16(3):369–380.

[76] Kluckner, S., Birchbauer, J. A., Windisch, C., Hoppe, C., Irschara, A., Wendel, A., Zollmann, S., Reitmayr, G., and Bischof, H. (2011). Construction Site Monitoring from Highly-Overlapping MAV Images. In *Proceedings of the IEEE International Conference on Advanced Video and Signalbased Surveillance AVSS Industrial Session*.

[77] Knecht, M., Dünser, A., Traxler, C., Wimmer, M., and Grasset, R. (2011). A Framework for Perceptual Studies in Photorealistic Augmented Reality. In *Proceedings of the 3rd IEEE VR 2011 Workshop on Perceptual Illusions in Virtual Environments*.

[78] Kruijff, E., Swan, J. E., and Feiner, S. (2010). Perceptual issues in augmented reality revisited. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2010)*, pages 3–12. IEEE.

[79] Langley, P. (1999). User Modeling in Adaptive Interfaces 1 The Need for Automated User Modeling. In *Courses and lectures-international centre for mechanical sciences*, pages 357–370.

[80] Langlotz, T., Wagner, D., Mulloni, A., and Schmalstieg, D. (2010). Online Creation of Panoramic Augmented Reality Annotations on Mobile Phones. *IEEE Pervasive Computing*, PP(99):1–12.

[81] Lerotic, M., Chung, A. J., Mylonas, G., and Yang, G.-Z. (2007). Pq-Space Based Non-Photorealistic Rendering for Augmented Reality. In *Proc. MICCAI '07*, pages 102–109.

[82] Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. (2004). Lazy snapping. *ACM Transactions on Graphics*, 23(3):303.

[83] Livingston, M. and Ai, Z. (2009). Indoor vs. outdoor depth perception for mobile augmented reality. In *IEEE Virtual Reality Conference (VR 2009)*, pages 55–62.

[84] Livingston, M., Ai, Z., Karsch, K., and Gibson, G. O. (2011). User interface design for military AR applications. In *IEEE Virtual Reality Conference (VR 2011)*, pages 175–184.

[85] Livingston, M. A., II, J. E. S., Gabbard, J. L., Höllerer, T. H., Hix, D., Julier, S. J., Baillot, Y., and Brown, D. (2003). Resolving Multiple Occluded Layers in Augmented Reality. *Symposium on Mixed and Augmented Reality*.

[86] Looser, J., Billinghurst, M., and Cockburn, A. (2004). Through the looking glass: the use of lenses as an interface tool for Augmented Reality interfaces. In *Computer Graphics and Interactive Techniques in Australasia and South East Asia*.

[87] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[88] Marr, D. (1982). Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. *Phenomenology and the Cognitive Sciences*, 8(4):397.

[89] Mendez, E. (2010). *Visualization On the Usage of Context for Augmented Reality*. PhD thesis, Graz University of Technology.

[90] Mendez, E. and Schmalstieg, D. (2009). Importance masks for revealing occluded objects in augmented reality. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '09*, pages 247—-248, New York, New York, USA. ACM Press.

[91] Milgram, P., Takemura, H., Utsumi, A., and Kishino, F. (1994). Augmented Reality: A class of displays on the reality-virtuality continuum. *Proceedings of Telemanipulator and Telepresence Technologies*, 2351:282–292.

[92] Mistry, P. and Maes, P. (2009). SixthSense. In *ACM SIGGRAPH ASIA 2009 Sketches on - SIGGRAPH ASIA '09*, page 1, New York, New York, USA. ACM Press.

[93] Mulloni, A., Dünser, A., and Schmalstieg, D. (2010). Zooming interfaces for augmented reality browsers. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10*, page 161, New York, New York, USA. ACM Press.

[94] Nowell, L., Hetzler, E., and Tanasse, T. (2001). Change blindness in information visualization: A case study. In *Proc. of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, pages 15–22.

[95] Nurminen, A., Kruijff, E., and Veas, E. (2011). HYDROSYS: a mixed reality platform for on-site visualization of environmental data. In *W2GIS'11 Proceedings of the 10th international conference on Web and wireless geographical information systems*, pages 159–175.

[96] Osberger, W., Maeder, J., and Bergmann, N. (1998). A Perceptually Based Quantization Technique for MPEG Encoding. In *Proceedings SPIE 3299 - Human Vision and Electronic Imaging III*, pages 148–159.

[97] Pentenrieder, K., Bade, C., Doil, F., and Meier, P. (2007). Augmented Reality-based factory planning - an application tailored to industrial needs. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2007)*, pages 1–9. Ieee.

[98] Perlin, K. (1985). An image synthesizer. *ACM SIGGRAPH Computer Graphics*, 19(3):287–296.

[99] Praun, E., Hoppe, H., Webb, M., and Finkelstein, A. (2001). Real-time hatching. In *Proceedings of ACM SIGGRAPH*, pages 579–584, New York, NY, USA. ACM.

[100] Reitmayr, G., Langlotz, T., Wagner, D., Mulloni, A., Schall, G., Schmalstieg, D., and Pan, Q. (2010). Simultaneous Localization and Mapping for Augmented Reality. *2010 International Symposium on Ubiquitous Virtual Reality*, pages 5–8.

[101] Ren, C. and Reid, I. (2011). gSLIC: a real-time implementation of SLIC superpixel segmentation. *University of Oxford, Department of Engineering*, pages 1–6.

[102] Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 10–17.

[103] Rosenholtz, R., Li, Y., Mansfield, J., and Jin, Z. (2005). Feature congestion: a measure of display clutter. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 761–770.

[104] Rosten, E., Reitmayr, G., and Drummond, T. (2005). Real-time video annotations for augmented reality. *Advances in Visual Computing*.

[105] Sandor, C., Cunningham, A., Dey, A., and Mattila, V. (2010). An Augmented Reality X-Ray system based on visual saliency. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2010)*, pages 27–36. IEEE.

[106] Sandor, C., Cunningham, A., Eck, U., Urquhart, D., Jarvis, G., Dey, A., Barbier, S., Marner, M. R., and Rhee, S. (2009). Egocentric space-distorting visualizations for rapid environment exploration in mobile mixed reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, pages 211–212. Ieee.

[107] Santner, J., Pock, T., and Bischof, H. (2011). Interactive multi-label segmentation. In *Computer Vision- ACCV 2010*, pages 397–410. Springer.

[108] Schall, G. (2008). The transcoding pipeline: Automatic generation of 3d models from geospatial data sources. In *Proceedings of the 1st International Workshop on Trends in Pervasive and Ubiquitous Geotechnology and Geoinformation (TIPUGG 2008)*.

[109] Schall, G., Mulloni, A., and Reitmayr, G. (2010a). North-centred orientation tracking on mobile phones. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2010)*, pages 267–268. IEEE.

[110] Schall, G., Schmalstieg, D., and Junghanns, S. (2010b). VIDENTE-3D Visualization of Underground Infrastructure using Handheld Augmented Reality. *Geohydroinformatics-Integrating GIS and Water Engineering" CRC Press/Taylor and Francis Publisher: CRC*, 1:1–17.

[111] Schall, G., Wagner, D., Reitmayr, G., Taichmann, E., Wieser, M., Schmalstieg, D., and Hofmann-Wellenhof, B. (2009). Global pose estimation using multi-sensor fusion for outdoor Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, pages 153–162. IEEE.

[112] Schall, G., Zollman, S., and Reitmayr, G. (2011). Bridging the gap between Planning and Surveying with Augmented Reality User Interfaces. In *Mobile HCI 2011 Workshop "Mobile Work Efficiency: Enhancing Workflows with Mobile Devices"*, pages 1–4, Stockholm.

[113] Schall, G., Zollmann, S., and Reitmayr, G. (2012). Smart Vidente: advances in mobile augmented reality for interactive visualization of underground infrastructure. *Personal and Ubiquitous Computing*, pages 1–17.

[114] Schmalstieg, D., Schall, G., Wagner, D., Barakonyi, I., Reitmayr, G., Newman, J., and Ledermann, F. (2007). Managing complex augmented reality models. *IEEE Computer Graphics and Applications*, 27(4):48–57.

[115] Schoenfelder, R. and Schmalstieg, D. (2008). Augmented Reality for Industrial Building Acceptance. In *IEEE Virtual Reality Conference (VR 2008)*, pages 83–90.

[116] Shneiderman, B. (1996). The Eyes Have It: {A} Task by Data Type Taxonomy for Information Visualizations. In *IEEE Visual Languages*, pages 336–343.

[117] Simons, D. J. (2000). Current Approaches to Change Blindness. *Visual Cognition*, 7:1–15.

[118] Sinha, S. N., Steedly, D., Szeliski, R., Agrawala, M., and Pollefeys, M. (2008). Interactive 3D architectural modeling from unordered photo collections. In *SIGGRAPH Asia '08*. ACM.

[119] Spence, R. and Apperley, M. (2011). *Bifocal Display*. The Interaction Design Foundation, Aarhus, Denmark.

[120] Steiner, J., Zollmann, S., and Reitmayr, G. (2011). Incremental Superpixels for Real-Time Video Analysis. In *Computer Vision Winter Workshop*.

[121] Tatzgern, M., Kalkofen, D., and Schmalstieg, D. (2013). Dynamic Compact Visualizations for Augmented Reality. In *IEEE Virtual Reality Conference (VR 2013)*.

[122] Unger, M., Pock, T., Trobin, W., Cremers, D., and Bischof, H. (2008). H.: Tvseg-interactive total variation based image segmentation. In *In: British Machine Vision Conference (BMVC*. Citeseer.

[123] Uratani, K. and Machida, T. (2005). A study of depth visualization techniques for virtual annotations in augmented reality. In *IEEE Virtual Reality Conference (VR 2005)*.

[124] Ventura, J., DiVerdi, S., and Höllerer, T. (2009). A sketch-based interface for photo pop-up. *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling - SBIM '09*, page 21.

[125] Ventura, J. and Hollerer, T. (2012). Wide-area scene mapping for mobile visual tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2012)*, number November, pages 3–12.

[126] Viola, I., Kanitsar, A., and Gröller, M. E. (2005). Importance-driven feature enhancement in volume visualization. *IEEE transactions on visualization and computer graphics*, 11(4):408–18.

[127] Wagner, D. and Mulloni, A. (2010). Real-time panoramic mapping and tracking on mobile phones. In *IEEE Virtual Reality Conference (VR 2010)*.

[128] White, S. (2009). Interaction with the Environment: Sensor Data Visualization in Outdoor Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, pages 5–6.

[129] Wither, J., Coffin, C., Ventura, J., and Hollerer, T. (2008). Fast annotation and modeling with a single-point laser range finder. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2008)*, pages 65–68. Ieee.

[130] Wither, J., DiVerdi, S., and Höllerer, T. (2009). Annotation in outdoor augmented reality. *Computers & Graphics*, 33(6):679–689.

[131] Wither, J. and Hollerer, T. (2005). Pictorial depth cues for outdoor augmented reality. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers (ISWC '05)*.

[132] Woodward, C., Hakkarainen, M., Korkalo, O., Kantonen, T., Rainio, K., and Kähkönen, K. (2010). Mixed reality for mobile construction site visualization and communication. In *10th International Conference on Construction Applications of Virtual Reality (CONVR2010)*, pages 1–10.

[133] Zollmann, S., Kalkofen, D., Hoppe, C., Kluckner, S., Bischof, H., and Reitmayr, G. (2012a). Interactive 4D overview and detail visualization in augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2012)*.

[134] Zollmann, S., Kalkofen, D., Mendez, E., and Reitmayr, G. (2010). Image-based ghostings for single layer occlusions in augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2010)*, pages 19–26. IEEE.

[135] Zollmann, S. and Reitmayr, G. (2012). Dense depth maps from sparse models and image coherence for augmented reality. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, pages 53–60.

[136] Zollmann, S., Schall, G., Junghanns, S., and Reitmayr, G. (2012b). Comprehensible and Interactive Visualizations of GIS Data in Augmented Reality. *Advances in Visual Computing*, pages 675–685.