Graz University of Technology

Institute for Computer Graphics and Vision

Dissertation

Draft Copy: May 27, 2013
Visualization in Outdoor Augmented Reality

Stefanie Zollmann

Graz, Austria, April 2013

Thesis supervisors

Prof. Gerhard Reitmayr

Prof. Tobias Höllerer

To ...

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

Ben Shneiderman

Abstract

Industrial outdoor applications in the context of architecture, construction and engineering often require a fast access to relevant data directly on-site. Augmented Reality (AR) provides not only this data access, but also 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 work environment, the workers are able to focus on the important tasks, while getting the information embedded into their view. Relevant tasks, such as the the on-site planning of new infrastructures, the information query about subsurface or overground infrastructure elements, the surveying of infrastructure as well as the monitoring of construction sites can benefit from this 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. Especially when combining virtual information with information from the physical environment, serious perceptual problems can occur. For instance, missing pictorial cues data can lead to a wrong scene perception. Another problem that often occur due to the combination of virtual and physical information are insufficient depth cues that complicate the depth estimation in these scenes. Furthermore, the visualization of complex information in an AR overlay often leads to information clutter. In this thesis, we address these problems by proposing adequate visualization techniques that take these perceptual difficulties into account. In particular, we developed methods for visual coherence to achieve a seamless integration of virtual content into the physical environment, additional graphical hints for improving the depth estimation and finally methods for 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 werde ich die Herausforderungen für Vor-Ort Visualisierungen von industriellen Daten mittels Augmented Reality diskutieren. Hierbei werde ich insbesondere auf die Themen visuelle Kohärenz, Tiefenwirkung und Informationsüberfluss eingehen. Im Verlauf meiner Arbeit werde ich darüber hinaus Ansätze und Techniken vorstellen, welche es erlauben diese Herausforderungen zu bewältigen.

Keywords. Augmented Reality, Outdoor, Visualization, Perception

Draft Copy: May 27, 2013

Acknowledgments

empty

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 that describe non-existing objects, meta information or hidden information. The visualization of these kinds 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 geo-spatial 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.

Figure 1.1: Visualization of non-existing objects [3], meta information [40] and hidden information in AR.

Meta information supports the user by providing information about real 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.

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

Outdoor applications from the Architecture, Construction and Engineering (ACE) industry can especially 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 [80]. 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" [6].

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

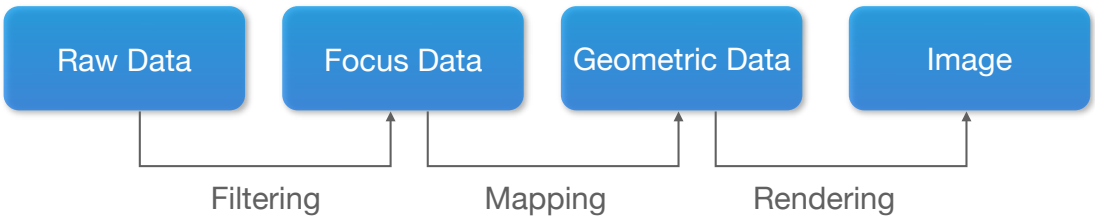


Figure 1.2: Information Visualization Pipeline.

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.

1.2 Visualization

There are various definitions about 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 the term 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." [43]

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 usually described step-by-step in one of the various versions of the *visualization pipeline* [43].

One version is the pipeline with 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. For instance, data points 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 a output device.

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

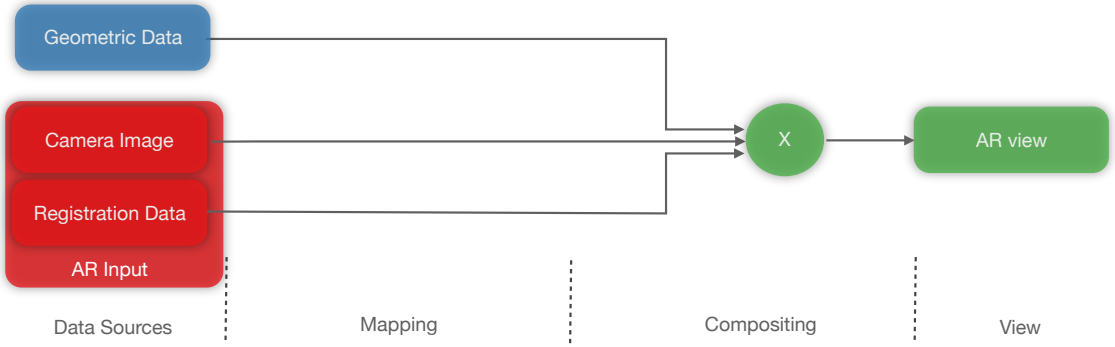


Figure 1.3: AR Visualization Pipeline illustrating a simple overlay using predefined geometric data, a camera image and registration data for creating the composition.

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

To use the 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).

The implementation of these additional steps seems to be straight-forward, if the registration between virtual content and the physical world representation is known (for instance in terms of a camera transformation matrix) and 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 basic 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 simple overlay, is an incomplete scene integration. Incomplete scene integration refers to the situation where important natural perceptual cues that the user needs to understand the spatial relationships are missing. 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 visualizing naturally invisible information with a simple overlay. In Figure 1.4 (Middle) we show several subsurface pipes visualized in an X-Ray

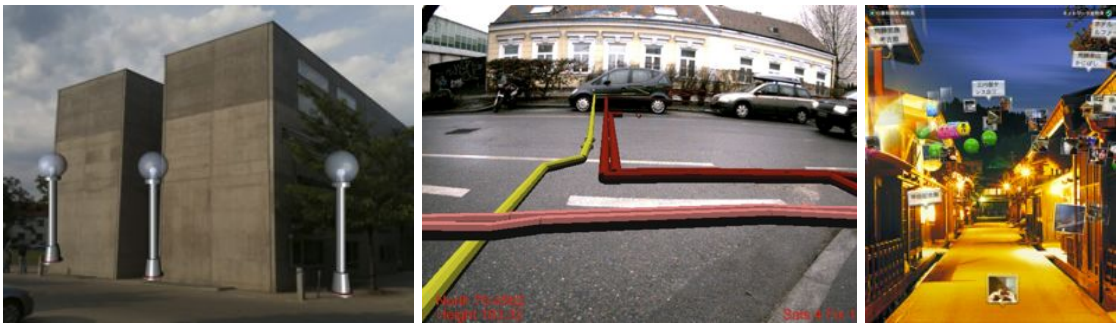


Figure 1.4: Visualization problems caused by naive composition functions. (Left) Virtual planned lamps are perceived to be located in front of the house not next to it. (Middle) Virtual pipes seem to float over the ground. (Right) In this touristic scenario the naive overlay of sight labels produces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

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.

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 [63]. 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."[6]

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* [35] into AR visualizations. Particularly, there is 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 [35]. Pictorial cues that have been sucessfully applied in previous AR visualizations comprise shadows [44], occlusion cues [12], or atmospheric cues [74]. Other researchers integrated additional pictorial cues such as shadow planes, color encoding or top-down views [118], or a virtual tunnel tool [8].

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 [74], 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 [54].

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 [67] and simulating image processing artifacts [65]. Completely in contrast are non-photorealistic renderings that display the physical environment with the same non-realistic visual properties as the virtual objects [30].

All these research directions have the same goal of increasing the convincibility of the 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, 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

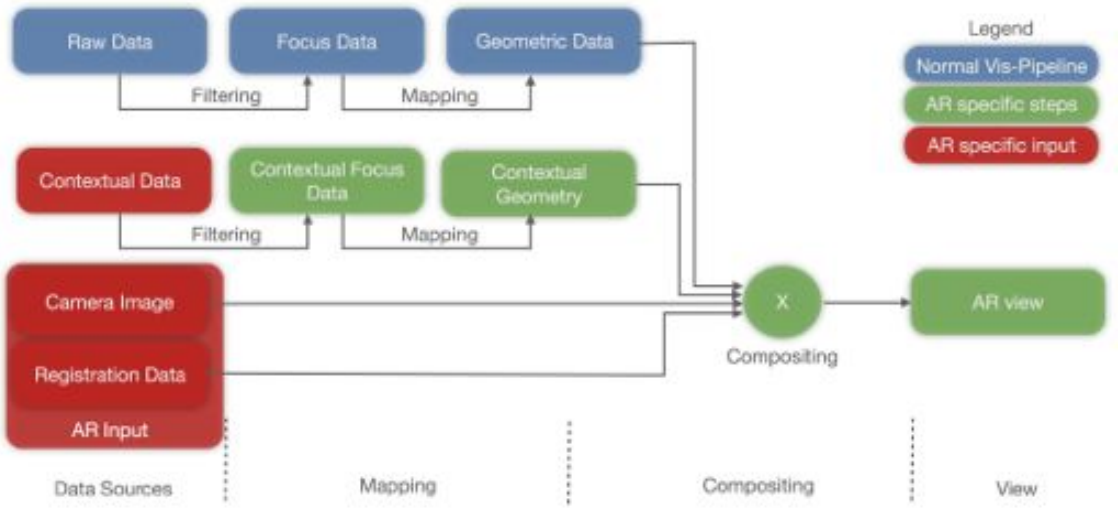


Figure 1.5: Comprehensible AR Visualization Pipeline.

understanding the presented content and for more advanced techniques such as achieving photo-realistic AR renderings.

By having a closer look at the techniques for comprehensible 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 work 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.

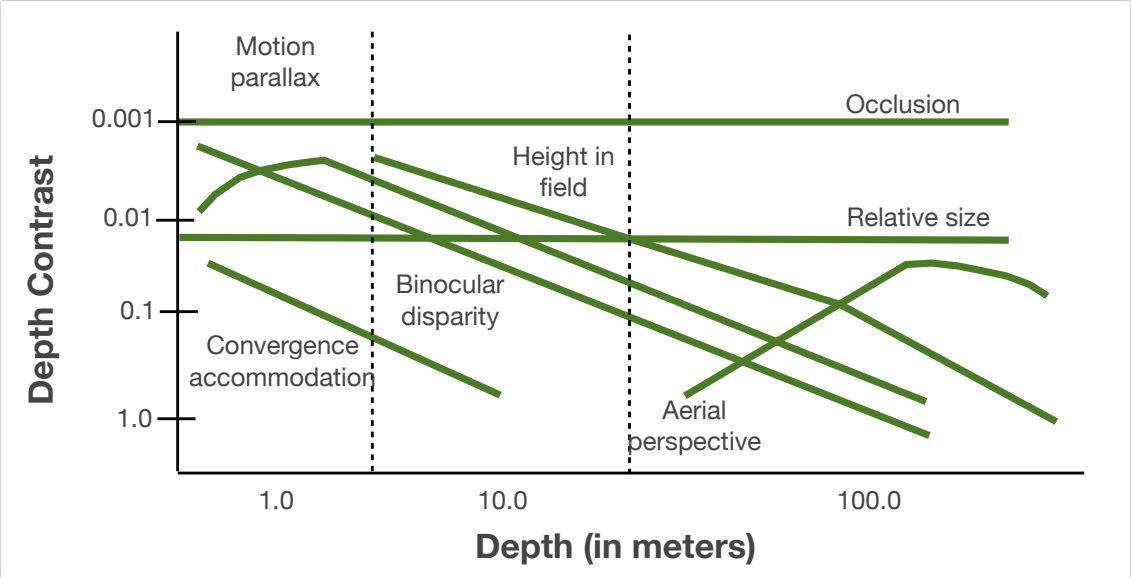


Figure 1.6: Pictorial depth cues. Cutting mapped different cues to their effectiveness to transfer depth in different distances. The depth contrast describes how effective they are. The depth on the x-axis describes in which distances they are working [20].

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 [20]. 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 and height in visual field), 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 [20], 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.

Figure 1.7: Draft: Difference between simple overlay, physical cues, virtual cues.

- 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 must 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 [12]. Fischer et al. applied a Time-of-Flight (TOF) range sensor to determine the depth order of virtual and physical objects [31].

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. [5] and Kalkofen et al. [54], 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 or salient regions, 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 visualizations. This issue emerges especially in AR visualizations of virtual objects that do not follow physical or 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 [20] this visual cue requires

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

and

Figure 1.8: 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.

"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 [20]. 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.

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 1.8 (Right).

The problem of insufficient physical pictorial cues for challenging scene interoperation was addressed by different research groups. For instance, Livingston et al. [74], Feiner et al. [27] and Coffin et al. [18] 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 1.9

Figure 1.9: Cluttered visualization. (Left) Cluttered label visualization. (Right) Cluttered visualization of a 3D mesh.

(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 invisible objects in the scene.

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 1.9 (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 [101]. Julier et al. proposed a system for information filtering in AR [52]. On the other hand Kalkofen et al. used Focus&Context techniques for information filtering in AR. Within their CityLens project, Nokia proposed a information filtering method called Sightline that removes unnecessary information[2].

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.

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

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: Comprehensible visualization requires a seamless integration of virtual and real information that maintains existing visual cues of the physical environment to achieve a convincing composition.
- 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 support conveying the spatial relationship between virtual content and the physical environment can be automatically created from geo-referenced data or interactively.
- H4: Filtering methods reduce self occlusions and information clutter in complex scenes.
- H5: Automatic methods for abstracting complex information in combination with interactive focus and context techniques reduce information clutter and self occlusion.

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 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 geo-referenced 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.

- **Filtering techniques** that allow visualizing complex data in an outdoor AR environment.
- 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.

One of the main challenges in AR visualization is that, in contrast to other visualization applications (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 real world scene
- Sparse knowledge about the real world scene
- Semi-dense representation of the real world scene

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

In cases where sparse knowledge about the real 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 1.10 (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 due to a aerial 3D reconstruction, it is important to extract the important information and preprocess it 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 selected cues extracted from the video image representations of the physical world. This

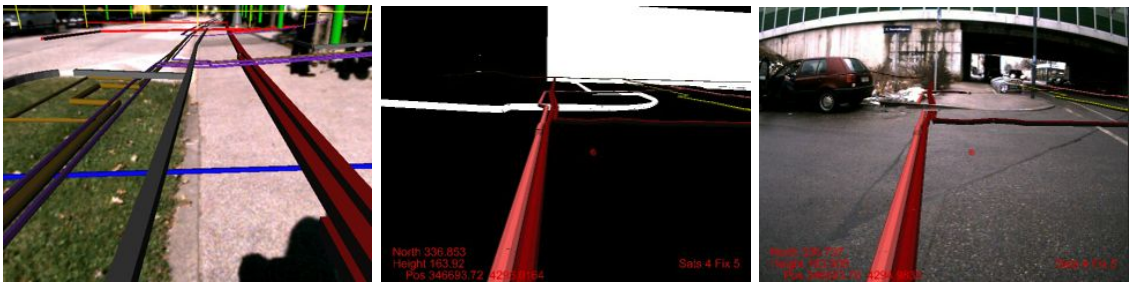


Figure 1.10: Sparse physical cues used for occlusion management. (Left) Physical world objects such as curbstones represented in the GIS database are added to the visualization for providing occlusions cues. (Middle) Sparse representations are used to create a occlusion mask (White). (Right) The same occlusion mask is used for occlusion culling to provide occlusion cues.

Figure 1.11: Seamless integration of virtual content in an X-Ray visualization of a virtual room inside a physical building. The left image shows a simple overlay, where the room is perceived to be located in front of the building. The right image shows the extraction of occlusion cues from the video image. Occlusion cues are extracted from the video image and preserved in the visualization. The virtual room is perceived to be located inside the building.

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 1.11).

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 1.8, 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 1.8, 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 [121]

  This publication contributes mainly to the seamless integration of virtual and physical content that forms the main content of Chapter 4. The main idea of this publication is to analyze the video image 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 [122]

  Both papers describe methods for extracting dense depth data from sparse depth information such as provided by GIS databases. The content of these publications contributes to the description of the used systems in Chapter that 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 work and implemented the most parts of this system as well as the applications and was also responsible for writing the paper. 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 [123]

  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 [120]

  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 these 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 show a set of visualization techniques for supporting the navigation and flight management of aerial vehicle. 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 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 [99]

- **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 [100]

  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 [66]

- **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, [48]

- **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 [108]

  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

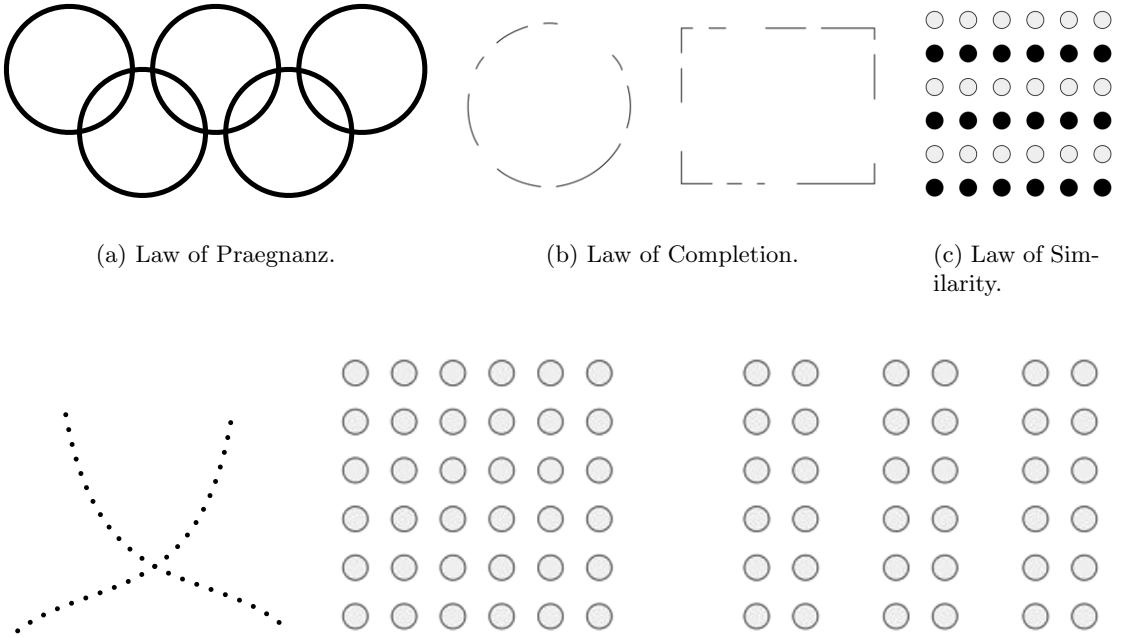
2.1 Perceptual Background

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 [35]. These models contain a set of assumptions that support a fast understanding of objects and relationships. Every time when a new situation and spatial relationship has to be understood, these assumptions are applied. 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 the 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 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 [35]. 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 (Figure 2.1(a)).
- 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.
- Law of Similarity. Objects that are similar in color or shape are likely to be grouped to one bigger entity.
- 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.
- Law of Proximity or Nearness. Objects in close proximity are likely to be perceived as a bigger entity.

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

Figure 2.2: Law of completion in illustrative X-Ray visualization. Left) The external shape of the leg is not rendered, but completed by our perception (Image Courtesy Bruckner et al. [13]). 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 (Image Courtesy Viola et al. [113]).

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 [13, 113]. 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).

On the other hand, by considering these laws for developing visualization techniques, the visual coherence of an AR visualization can be improved. For instance, the law of similarity should be considered when manipulating the appearance of virtual or physical objects. If different manipulation techniques are applied to to similar objects, their grouping can get lost.

2.1.2 From 2D to 3D

Another important aspect is that we have to understand how humans process 2D information and build 3D representations from this. In 1982, Marr proposed a computational approach explaining object perception [77]. His approach is based on a series of processing steps. 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 the discontinuities in orientations and depths. Finally, the 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.

The knowledge about the steps of creating 3D models from 2D information by the HVS,

Assumptions and Scales for Each of Nine Sources of Information About Layout and Depth

Source of Information	Assumptions	Implied Measurement Scale
All	Linearity of light rays (see Burton, 1945; but also Minnaert, 1993, for exceptions). Luminance or textural contrast. In general, the rigidity of objects (rigidity implies object shape invariance).	—
1. Occlusion	Opacity of objects. Helmholtz's rule, or good continuation of the occluding object's contour (Hochberg, 1971; Ratoosh, 1949; but see Chapanis & McCleary, 1955).	Ordinal
2. 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
3. Relative size	Similarly shaped objects have similar physical size (Bingham, 1993) Objects are not too close. Plurality of objects in sight. (*Not* familiarity with the objects, which denotes "familiar size" [e.g., Epstein, 1963]).	Unanchored ratio possible, but probably ordinal
4. 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
5. Aerial perspective	The medium is not completely transparent. The density of the medium is roughly uniform.	Probably ordinal
6. Binocular disparities	The distance between eyes. The current state of vergence. Unambiguous correspondences.	Absolute (Landy et al., 1991), but perhaps only ordinal (van den Berg & Brenner, 1994)
7. Accommodation	Complex spatial frequency distribution (Fisher & Ciuffreda, 1988). The current state.	Ordinal at best
8. Convergence	The distance between eyes. The current state.	Ordinal
9. Motion perspective	A rigid environment. A spatial anchor of zero motion (horizon or a fixated object).	Absolute (Landy et al., 1991), unanchored ratio, but perhaps only ordinal

Figure 2.3: Nine sources of information about spatial relationships. (Courtesy of Cutting [20].)

provide 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. Thus the 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 the depth is estimated during the step between the primal sketch and the 2.5D representation. This research 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

Figure 2.4: Pictorial depth cues in the physical world and in AR. Left) Pictorial Depth cues in an urban scenario. Right) Using a simple overlay of virtual information important depth cues are missing and lead to misinterpretations. **TODO:** mockups replace with right pics

relationships. He compared these cues according to their relative efficiency and described assumptions that have to be fulfilled to make them work (Figure 2.3). He also introduced a measurement of depth contrast describing each cue's efficiency.

As described in his research, **Occlusion** is a cue that provides ordinal information. It occurs when objects are occluded or partially occluded by another one. We can derive from this information that the occluded objects are located behind the occluding object. It is considered to be to most powerful depth cue since it is working at all distances and has a higher depth contrast than other sources (Figure 1.6). This cue is only working if the objects in the scene are opaque. Apparently, for transparent objects there is no occlusion. This means occlusion cues are especially missing in X-Ray visualizations.

The depth cue **Height in visual field** describes that objects that are further away are 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, if a set of assumptions is fulfilled [21]. 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, the information source **Height in visual field** can be used. On the other hand this means that an absolute depth estimation of floating or subsurface objects is more complicated, since in this case the assumptions can not be fulfilled.

Relative size and **Relative density** are two sources that are connected to each other. Objects that are further away appear smaller int 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 which are located further away loose their contrast and adjust more 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 or require stereo 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 2.4, Left). Unfortunately, some of these cues are not available when using a naïve AR overlay (Figure 2.4, Right). Comprehensible visualization techniques in AR address these theoretical models and laws to aviod these 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 and to understand the similarities and difference between different techniques. In contrast to the work of Kruijff et al. that classified perceptual problems in AR [68], our classification focuses on the visualization techniques themselves similar to work of Elmquist et al. [25]. 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 consisting language and classify the techniques. We found the following dimensions to be important:

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

In the following paragraphs we will describe these dimensions more in detail.

Addressed Visualization Problem The common goal of all AR visualization techniques is to improve the comprehension of virtual content being integrated in 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: Achieving a convincing scene integration, Supporting depth estimation, Reducing information clutter

Example Techniques

Achieving a convincing scene integration: Occlusion culling [12],

Supporting depth estimations: Virtual depth cues [74],

Reducing information clutter: Filtering [52].

Virtual Data Visibility On the other hand, the visibility of virtual data 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, the so-called X-Ray visualization, is one main features of AR and is used for various applications such as subsurface visualization or medical applications. As described in the last section, these kind of visualization have special challenges since occlusion cues have to be provided.

Domain: Occluded, Partially occluded, Visible

Examples

Occluded:Subsurface visualization in Vidente [97],

Partially occluded: Occlusion culling [12],

Visible: Tramlines [73].

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. Thereby we define physical cues as cues that try to mimic or rebuild natural 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 a a natural occlusion cue. On the other hand, we define virtual cues as graphical aids that are naturally not available in the physical world such as scales, measurements or other graphical hints.

Domain: none, physical, virtual

Characteristic Techniques: Magic Book [39], AR Ghostings [55], Cutaways [33].

Filtering The amount of filtering is another interesting dimension that allows to group visualization techniques. For the most of the simple AR visualization methods no filtering is applied since a predefined geometric model displayed in a simple overlay. When it comes to the visualization of more complex data, an adequate filtering is required to avoid information clutter. This effect is increased by the already complex physical environment where information is overlaid to. We can divide the dimension into techniques that use raw data and techniques that apply filtering.

Domain: Raw, Filtered

Characteristic Techniques: Shadowplanes [8], Filtering [52]

Abstraction Another way of reducing complexity is to move from a concrete representation to a more abstract representation. Abstractions allow reducing the amount

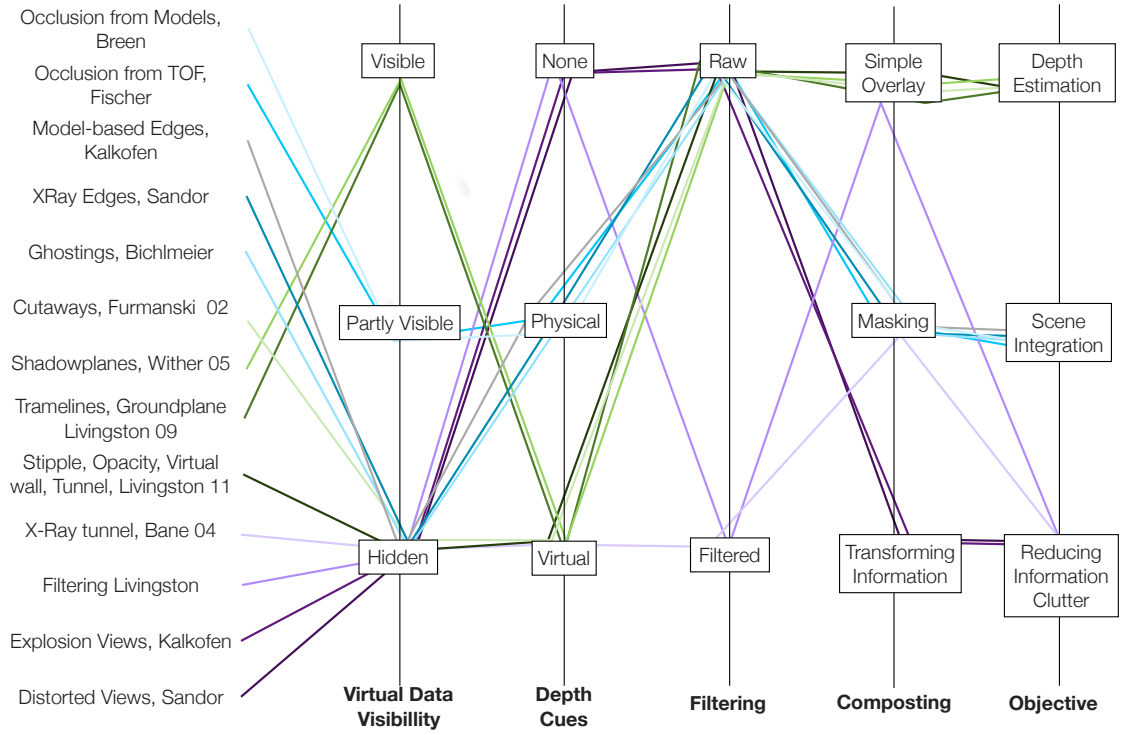


Figure 2.5: AR visualization techniques mapped to the taxonomy.

of information of the data by finding a different representation that preserves only the relevant information.

Domain: Concrete , Abstract

Characteristic Techniques: The most techniques display concrete information.

Composition The way the virtual and the physical information is composed into the final AR visualization 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 added to the physical world representation P. The final composition C is then given by $C = V + P$. 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 is used to rearrange physical or virtual items to avoid cluttered views. These techniques create the final composition by $C = T_V(V) + T_P(P)$ and depends on the transformation of virtual content T_V and of physical content T_P.

Domain: Simple Overlay, Masking, Transforming Information

Characteristic Techniques: Shadow Planes [8], Ghostings [5], Distorted Views [93]

Summary We used these dimension and their domains to classify the available visualization techniques from previous work. Since there is no visualization technique available

that uses explicit abstraction we removed this dimension from the classification. The mapping between techniques and the taxonomy is visualized in Figure 2.5. This overview shows that there is a big amount of AR visualization techniques addressing the problem of visualizing occluded data. It also shows that some dimensions have a 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, there is only a small amount of techniques apply a filtering techniques.

Another important aspect that becomes evident is the relationship between addressed visualization problem and data visibility, depth cues, filtering as well as compositing. Firstly, it seems that the most visualization technique that support depth estimation are using 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. There seems to be no need to address scene integration for visible virtual information. By embedding physical cues with a composition technique that uses masking the hidden virtual content can be integrated convincingly into the physical scenes. Finally, the classification shows that filtering techniques and information transformation techniques are mostly used for reducing information clutter. Depth cues seam to be not used in techniques that focus on reducing information clutter. **TODO:** highlighting gap here?

2.3 AR Visualization Pipelines

Based on the classification in the taxonomy, we can identify the main visualization objectives and the available data. This supports the decision which kind of visualization methods is used. In this section, we use our dimensional space to redefine the classical visualization pipeline for AR. Our AR visualization pipeline provides a simplified of the visualization process. Different implementations of this pipeline allow ordering the different approaches that are available into subgroups.

As mentioned in the introduction, for simple AR scenes that contain no occlusion and no complex data a simple composting can be used that combines a defined virtual geometry with the video image based on the registration data (Figure 1.3. Examples for this kind of visualization are the Magic Book where virtual content is overlayed over a book [39] or the see-through visualization within the Touring Machine [26].

The simple pipeline is not working for more complex situations with partially or completely hidden, or complex information. Therefore, researchers were developing 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

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

2.3.1 Achieving Visual Coherence

To achieve visual coherence and a seamless scene integration in AR, researchers proposed approaches that extract and use natural cues from the physical environment such as occlusion or shadows. We are referring to these natural cues as physical cues, since they can also be found in the physical world.

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. Thus, a lot of ghosting techniques in AR are based on techniques from illustrative renderings and volume rendering.

We can subdivide the methods for physical cues based on the data they are using. This depends on the availability of the contextual information about the physical world. Milgram was referring to the availability of the real world representation as *Extent of World Knowledge* [80]. 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 not available in every scenario, these image-based techniques focus on creating physical cues based on 2D real world data. In Figure 2.6 the process of extracting physical cues from the camera image is depicted in an instance of the AR visualization pipeline. Important elements from the camera are filtered and mapped to an importance map that represents the physical cues (Figure 2.6, (Left)). These cues are then combined with the camera image and virtual geometries to create the final AR visualization.

Such an approach has been first discussed by Kalkofen et al.[55]. In their work, they extracted edges from a camera image and used them as cues to create edge-based ghostings (Figure 2.7, Left). The AR visualization pipeline in Figure 2.6 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 this approach by using a combination of edges and bright pixels as physical depth cues [9]. Avery et al. also applied edges to improve their X-ray vision system [4] (Figure 2.7, Middle). Based on this work, Sandor et al. later on defined the physical cues as being

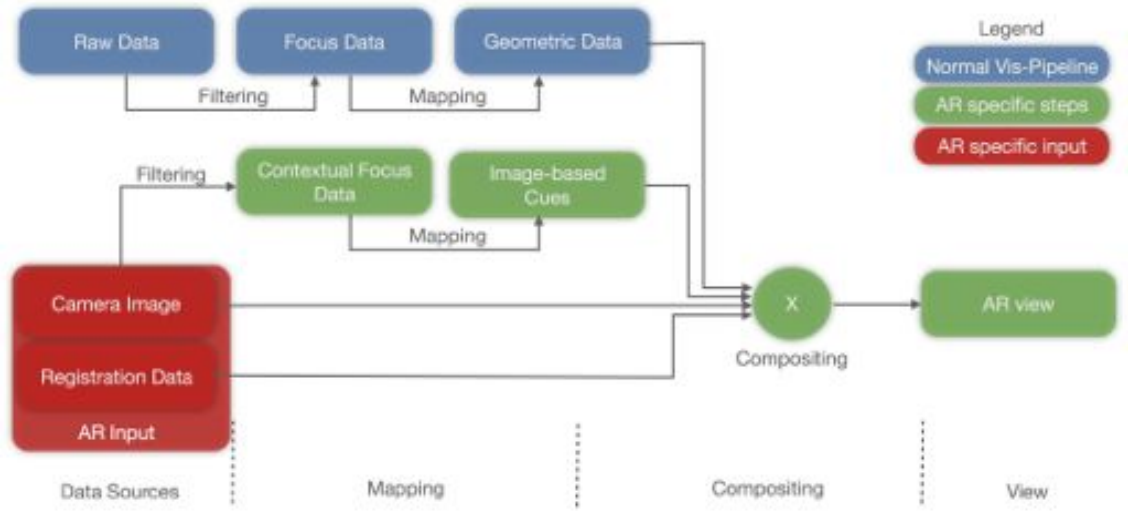


Figure 2.6: AR visualization pipeline for extracting image-based physical cues. (Data sources) Important elements from the camera are extracted and (Mapping) mapped to an importance map that represents the physical cues. (Composition) These cues are then combined with the camera image and virtual geometries to the final AR view.

Figure 2.7: Examples for using image-based physical cues. Left) Using edges for creating a ghosting in a medical application (Courtesy of Kalkofen et al. [55]). Middle) Using edges for visualizing remote views (Courtesy of Avery et al. [4]). Right) Using saliency for a ghosted view (Courtesy of Sandor et al. [92])

saliency information. They computed saliency masks from the camera image and the virtual content to decided which information should be preserved in the final rendering [92] (Figure 2.7, 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 Model-based physical cues are derived from a 2.5D or 3D representation of the environment (Figure 2.8). This contextual data can be used if a accurate 3D registration and accurate models are available. In this case there is no semantic knowledge about the scene required since the correct occlusion can be simply be calculated from the available contextual data. Some of the model-based techniques use

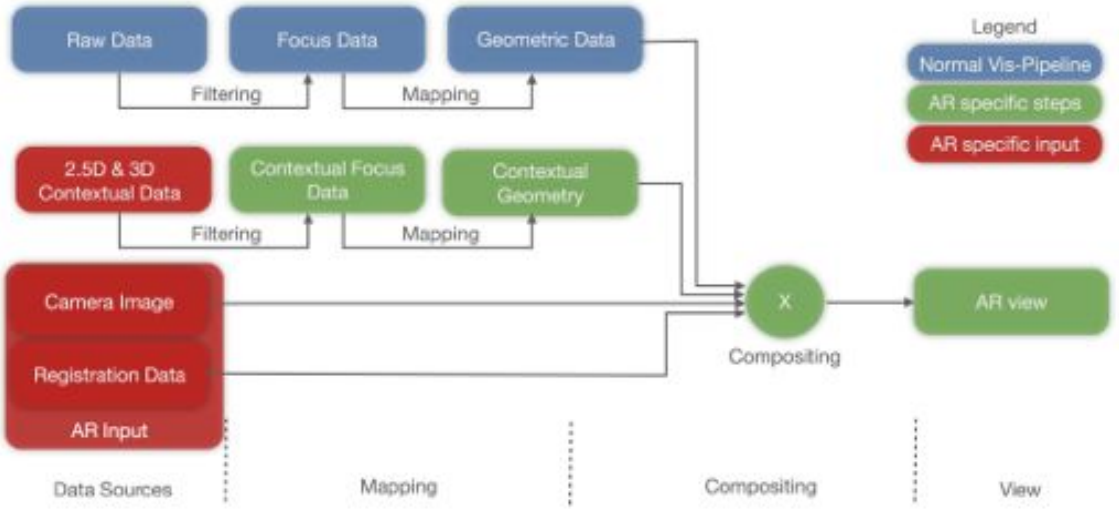


Figure 2.8: Pipeline for creating model-based physical cues. A 2.5D or 3D representation of the physical world is used as additional input for the visualization to create physical cues.

the contextual data for occlusion culling. One of the earliest approach in this field used an interactive method to align models of real world objects and to apply these models then for occlusion culling [12]. In the same paper the authors also proposed to use stereo vision to create a 2.5 depth map. A similar approach was applied by Fischer et al., who used a time-of-flight camera to create a depth map that can be used for occlusion culling (Figure 2.11, Left) [31].

More recent approaches use 3D models of the physical environment for increasing the visual coherency by deriving physical cues from the geometric or visual properties of the model. For instance, Lerotic et al. [71] 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 deficiencies in AR applications in cases where hidden structure is of interest [10]. 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 a registered 3D CAD model in augmented reality [55] ((Figure 2.11, Middle)). The last three model-based approaches only work well, if the models of the occluding object show some interesting features in their geometry. Mendez and Schmalstieg presented an approach that even allows to create comprehensible ghostings for rather simple shaped physical objects [79]. By mapping a predefined importance map on the model of the physical world, selected areas of the real object can be preserved (Figure 2.11, Right).

However, model-based approaches were mostly used for AR visualization indoors, 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 complicated to model these complex environments accurately.

Figure 2.9: Examples for using model-based physical cues. Left) Using time-of-flight camera data for occlusion culling (Courtesy of Fischer et al. [31]). Middle) A ghostings based on edges extracted from a registered 3D CAD model (Courtesy of Kalkofen et al. [55]). Right) Using importance maps applied on a 3D model (Courtesy of Mendez et al. [79])).

2.3.2 Supporting Depth Estimation

The previously described techniques use primarily occlusion cues to achieve a seamless integration between virtual content and physical world. Nevertheless, these cues usually only provide ordinal depth information. In this section we will discuss a group of visualization techniques that focus on supporting the depth estimation. Usually the estimation of depth is complicated if the presented objects are not following physical laws [20], which often occurs in AR visualizations. For instance, this 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 real world. In the literature they were also called *graphical aides* [73]. The AR visualization pipeline has to integrate these cues additionally. We can mainly differ between three ways of adding virtual cues to the AR visualization:

- External geometries
- Mapping from virtual data
- Cutaways

These techniques differ on how the virtual cues are created, but they follow the same goal of allowing the user a depth estimation of the virtual objects.

External aid geometries The group of related work that is discussed in this paragraph includes predefined virtual geometries (such as ground planes or parallel lines) into the composition to support the depth comprehension (Figure 2.10). Usually, these additional cues are already available in a geometric representation. For instance, Livingston et al. included a set of parallel lines (called tram lines) into their visualization of colored makers to improve the depth perception of the users in an indoor and outdoor scenario [73]. Additionally, they were adding grid points to the tram lines. The authors performed a

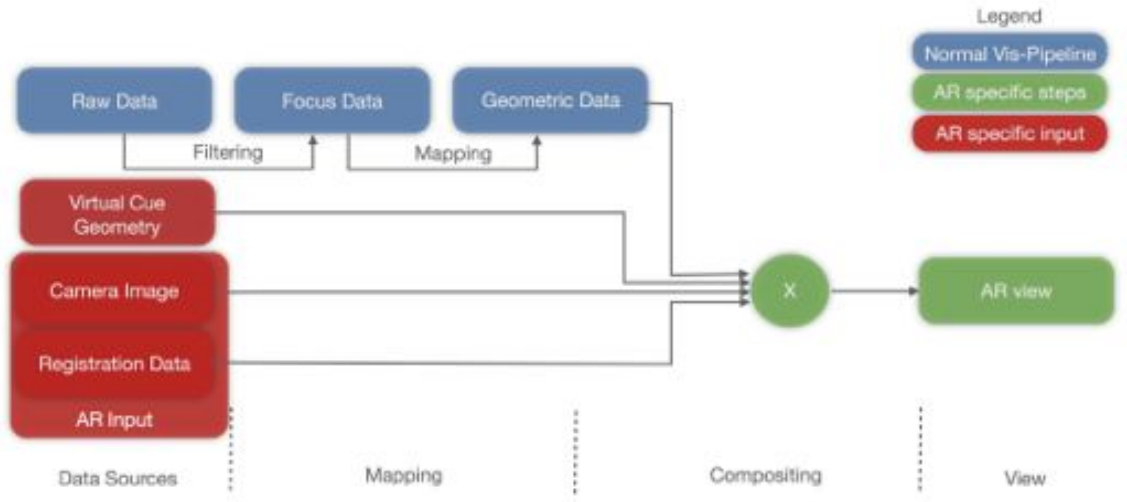


Figure 2.10: Pipeline for creating external virtual cues.

Figure 2.11: Examples for using external virtual cues. Left) Adding a set of parallel lines to improve the depth perception (Courtesy of Livingston et al. [73]). Middle) Using a virtual ground grid to show absolute distances (Courtesy of Livingston et al. [74]). Right) Virtual Shadow Planes for visualizing absolute depths (Courtesy of Wither et al. [118]).

user study with this visualization and could confirm on a positive effect for depth estimation outdoors. It seemed that the users were tending to decrease overestimated depth judgments. For indoor usage adding the tram lines was counterproductive since it again decreased the already underestimated depth.

Livingston et al. also introduced other examples of external geometries to improve the depth perception such as a ground grid, which is a plane on the ground that either shows the distance to the user with concentric circles or with parallel lines [74]. This virtual cue integrates the visual cues of height in visual field, and relative size. The ground plane cue can be extend by ties that show the connection between the virtual object of interest and the ground plane. The last feature is especially interesting for floating or subsurface objects. Wither et al. introduced a similar concept with the *Shadow Planes*, two orthogonal planes with depth measurements are used to project shadows of virtual objects onto it [118]. The shadows in combination with the distance scale on the planes was introduced to support the user in judging distances. Nevertheless, first study results

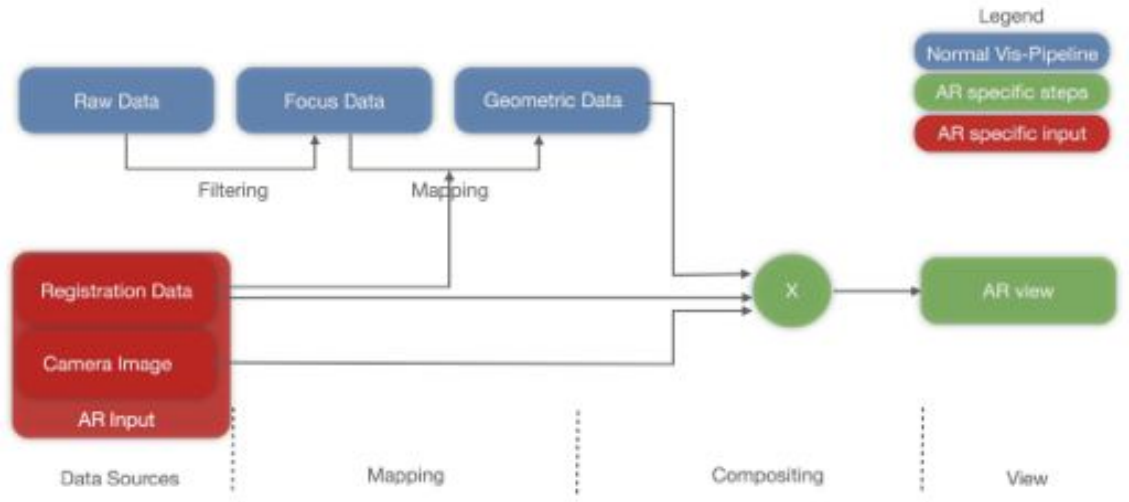


Figure 2.12: Mapping distance to appearance.

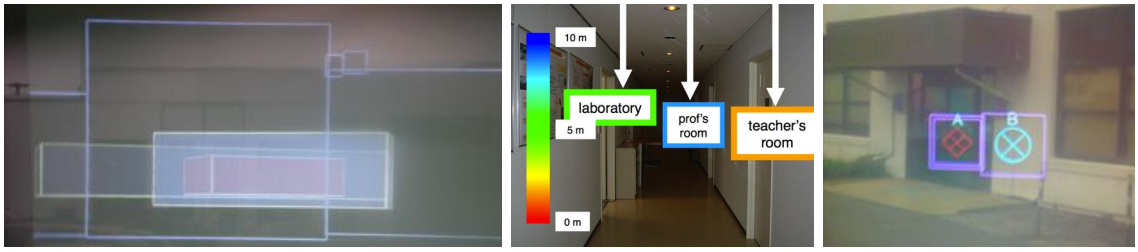


Figure 2.13: Methods that apply a mapping from distance to appearance. Left) Using intensity and ocacity for encoding depth [75] Middle) Using color to encode distance [111] Right) The Tunnel Tool varies the amout of rectangles depending on the distance to simulate a virtual tunnel [74].

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. Thereby, as shown in Figure 2.12, the distance from the user to the virtual object which is given by the transformation matrix is added to the mapping process and included into the visual appearance of the object. Visual characteristics that are used to encode distance are transparency, color, frequency of stipples or density of virtual edges.

This kind of mapping was discussed by Livingston et al. [75]. In their work the authors suggested to change the opacity and the intensity of building renderings based on the distance of the layer. They compared this visual mapping to constant opacity and intensity and found a significant influence of using the mapping to decreasing opacity (Figure 2.13. Left). Uratani et al. discussed how to map monocular depth cues to the appearance by using the distance of labels such as [111]:

- Depth of field by blurring the frame of the label depending on the distance.
- Relative size by changing the size of the 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.

Furthermore they encoded the absolute distance into a color pattern (Figure 2.13, Middle). More recently, Livingston et al. used a set of mapping techniques to encode depth of virtual targets and compared them to each other [74]. 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 2.13, Right).
- Texture gradient by including a texture pattern into the label.

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

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

Cutaways Cutaways are the third big group of related work that creates virtual cues to support the perception of the users and especially with the main goal of supporting depth estimation. Cutaways are often considered as being a part of the group of focus and context techniques, since they allow one to inspect data in the cutaway area more in detail. But actually they can do more than filtering, since they are also able to provide virtual depths cues such as a box around the hidden object that shows measurements or at least perspective cues given by the shape of the cutout geometry or the back faces of the occluding object. In contrast to the last two techniques 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 usually wants to reveal hidden parts of an object to the observer. To support the depth estimation the cutaway does not only just cut out the occluding object but should also add additional information such as back face or a cut out geometry.

To create a convincing cutaway in AR, the cutout geometry is need as well as a model of the occluding object. Since the occluding object is in this case the physical world, contextual data about the physical world is required (Figure 2.14). This data could be a

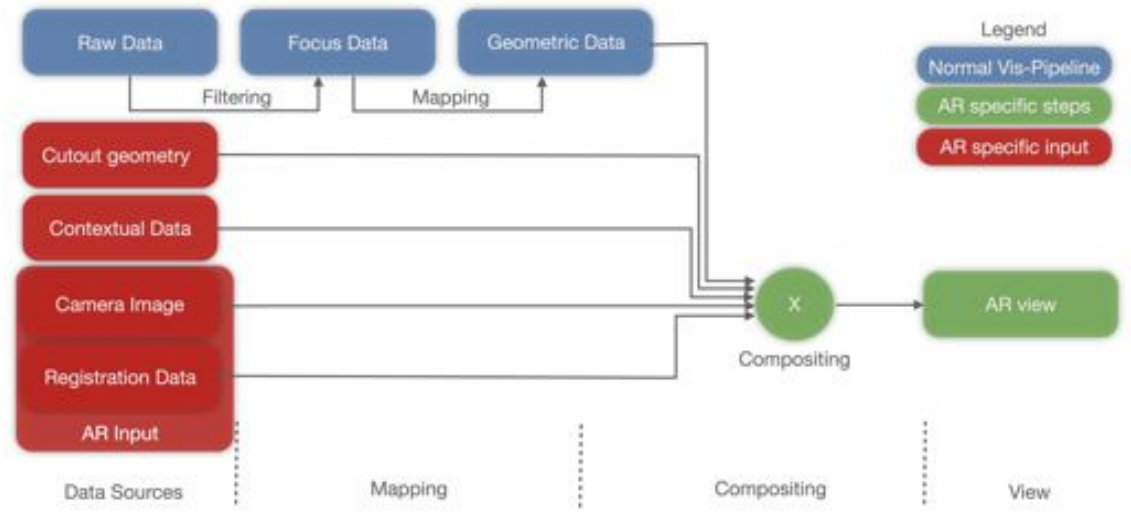


Figure 2.14: Creating additional virtual cues with cutaway geometries.

Figure 2.15: Cutaways as virtual cues in AR. Left) Virtual cutaways rendered on a wall to reveal a hidden green target object. Middle) Virtual cutaways used to visualize the interior of a car [53]. Right) A virtual excavation with a depth scale is used to visualize subsurface infrastructure in an urban civil engineering scenario [78].

rough surface model or a phantom geometry. By combining the cutout geometry and the phantom model the correct cutout is computed 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 [33]. Among different suggestions for visual aids (ground planes grids, distance marker and temporal distance markers), they showed how to render virtual cutaways on walls to reveal hidden information (Figure 2.15, Left). In a user study they compared the visualization of a target inside a wall with and without cutaways. 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 also stated that the findings from their study can be influenced by technical limitations of the AR implementation. This was confirmed by the participants reporting that their perception was influenced by the jitter from the

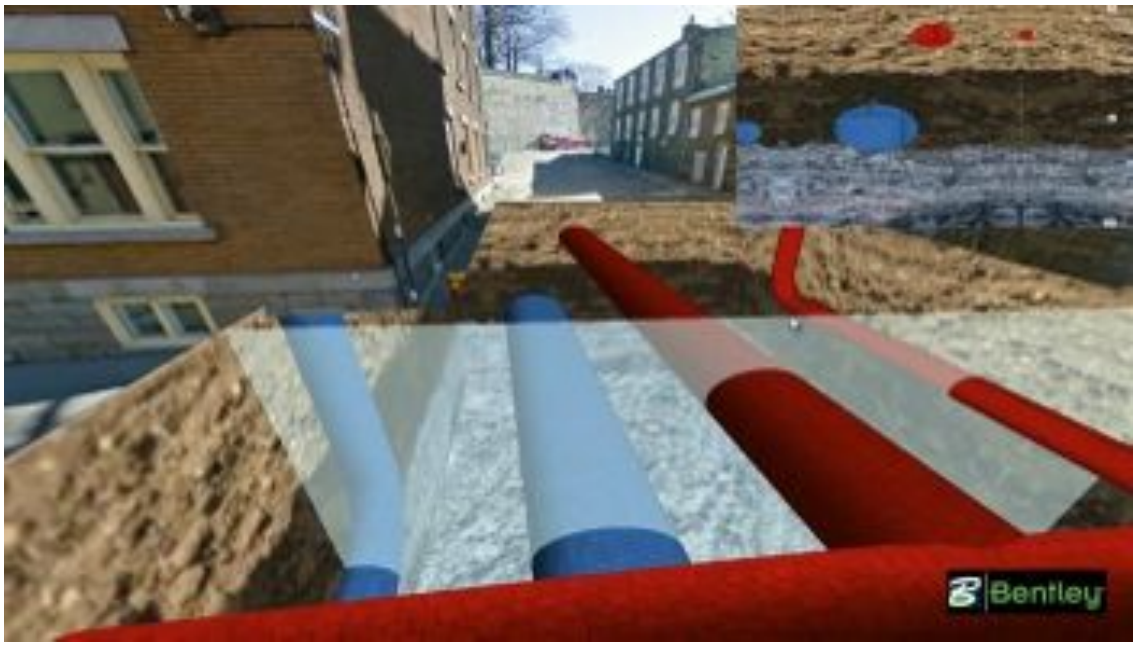


Figure 2.16: Vertical Slicing Tool. User can interactively control the slicing plane. The inset shows a 2D vertical section of the cut to support the comprehension of the spatial relationship between virtual elements (Image courtesy of Stéphane Cote, Bentley)

registration.

Later on Kalkofen used cutaways to visualize the interior of a car (Figure 2.15, Middle). By using the phantom representation of the occluding object (the car) to compute the bending of the contour of the cut-out area, he was able to preserve the shape of the occluding object [53]. Nevertheless, 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, for instance, renders the cutout volume to add depth cues.

In the Vidente[1] project Mendez et al. showed how to include such additional visual hints in a cutaway visualization (Figure 2.15, Right) [78]. 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 improve 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 to inspect the space between subsurface pipes more accurately by allowing the user to move the slicing tool in the virtual excavation and showing a 2D vertical section of the cut in a separate view (Figure 2.16).

[1] `www.vidente.at`
[2] `http://communities.bentley.com/`

Figure 2.17: Focus&Context techniques for information filtering in AR. Left) Magic lens that defines a focus area for displaying defined virtual content [76]. Middle) Interactive X-Ray tunnel [8]. Right) Focus&Context allows to explorer occluded information in an X-Ray view [54].

2.3.3 Reducing Visual Clutter

With the increasing amount of omnipresent information, the presentation of these information is more likely to become a subject of clutter. Due to this fact, researcher 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." [90]

In the research field of Information Visualization several techniques were developed 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 Augmented Reality visualizations, complex data is usually embedded in complex physical environments that are crowded with information by nature. Thus, information clutter is a big issue in AR visualizations as well and several researchers introduced methods that allow to address the problem of information clutter in AR environments. Similar to the methods in 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 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. One of the early research works that investigated filtering in AR is the work of Julier et al. [52]. 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 the importance of virtual buildings was calculated and used to decide whether a virtual building should be culled or not. To avoid that small changes in user's positions extremely change the displayed content a fading function was added for smooth transitions. Later, Livingston et al. used a similar filtering approach based on focus and nimbus areas of objects of interests for supporting military operation with AR [74].

Instead of applying a global filtering, focus and 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 [76]. Users of their system can control the lens with a physical marker. Inside the lens area virtual data is displayed. This allows the user to inspect the virtual data while avoiding a cluttered context area. Furthermore, the filtering criteria of the 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 [8]. These tools allow to define a focus area where virtual data such as heat distribution of a building is displayed. In the focus area the physical world is visualized (Figure 2.17 Middle). Kalkofen et al. used focus&context filters in a scene graph to allow users to explorer occluded information in an X-Ray view [54].

View Management In contrast to the filtering techniques, so-called view-management techniques do not completely remove non-relevant information. View management techniques use for instance distortion methods to reduce the space of non-relevant information in the visualization, but still keep them available for a fast overview [107]. These 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 complete remove occluding areas from a hidden object of interest [56]. Their technique translates occluding parts to a new position to reveal occluded information. Recently, they extended this techniques by using compact explosion views to avoid that the transformed content infers with the environment [109]. On the other hand, Sandor et al. used a distortion of occluding physical world objects to reveal occluded objects [93]. Whereas both work manipulate the appearance of the physical world, Rosten et al. and Grasset et al. applied transformation to virtual annotations for rearranging them based on an analysis of the current environment [40, 91].

2.3.4 Summary

In this section, we discussed different subgroups of related work and described how they create the visualization by using different instance of the AR visualization pipeline. We showed that these works address different perceptual issues in AR. Nevertheless, from the discussion we learned that a lot of problems are still not solved. Especially in the visualization of complex data in nearly unknown outdoor environment, there is still room for improvement.

So far existing methods for seamless scene integration of virtual content often focus on indoor usage. Thus several methods assume that a complete model of the physical world is available, which is complicated to achieve for outdoor environments (in particular if they are dynamic or a large operation range is needed as for inspecting subsurface infrastructure). Other methods assume that the scenes contain a lot of edges that can be used for providing occlusion cues. Since, urban outdoor scenes often contain more important information than edges or bright spots, we need methods that address these

measurements as well. Instead of being limited the 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 create additional virtual cues for supporting depth estimation are needed. In this thesis we will show how we can create such additional cues from geo-referenced data automatically and even allow to modify the data while providing consistent depth cues for the virtual objects.

Finally, there is only little work that investigates the visualization of complex data in AR. The most of these works focus on information that has 3 dimensions at most. If we want to visualize information that has more dimensions, for instance visualizing construction site progress with a 4D representation, these methods often do not allow to understand the relationship between multiple data sets. In this thesis, we will address this issue and propose visualization techniques that allow to visualize complex data even if it has more than 3 dimensions.

2.4 Applications

Several industrial applications, and in particular applications from the Architecture, Construction and Engineering (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 the 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 visualization of GIS data, such as for infrastructure maintenance [95], for agricultural data [62], or scientific data [83, 115].

In this section, we will describe several industrial outdoor applications that can benefit from an AR interface. Nevertheless, professional applications as well as normal applications require a careful visualization design. 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 it is important in many tasks to access information about assets in the field. They need for instance 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 make excavations on their private ground their are supposed to query information about subsurface objects in the proximity to avoid damages on public infrastructure.

Figure 2.18: Traditional Methods for Information Query of Digital Assets. Left) Accessing information with a paper printout. Right) Accessing information with a mobile GIS interface.

Figure 2.19: Traditional surveying methods. Left) User surveys point of interest using the theodolite. Middle) He has to note down information about the surveyed point (such as type). Right) Overview of a set of surveyed points with additional information.

This kind of information is often still presented on paper prints or on stationary computer systems. Mobile GIS systems already extend the way of information presentation and allow one to access information in the field (Figure 2.18). Nevertheless, this way of information presentation does not provide the spatial relationship between the digital asset and the physical world. This has to be built mentally by the workers themselves, which requires a lot of experience and is often a source of mistakes.

To allow less experienced people to understand information about buried assets and to reduce the mental workload for professional workers, a registered AR view can be beneficial, since it can support 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 geo-referenced information about newly built objects on-site. The surveyed information is used for as-built/as-planned comparison and

Figure 2.20: Traditional paper plans vs. an AR interface for planning. Left) Paper-based planning, newly planning lamps are visualized in orange. Right) AR interface for planning lamps.

for documentation purposes. In traditional surveying the field worker is capturing the points with a theodolite (Figure 2.19, Left) and has to move to each surveying point to place the device at its position. Measurements are stored on the device. Additionally, he draws the layout of the surveyed points in relationship to the construction site and between previously surveyed points (Figure 2.19, Middle). 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 also expert knowledge and produces a high mental workload. The need for this post-processing step in the office could be avoided by using AR as an interface for surveying. The post-processing time could be reduced by directly surveying the geometry in relationship to the physical world in combination with interactive tools that allow data input could make the post-processing unnecessary.

2.4.3 Planning Applications

Planning is an essential task for all building projects. Planning allows to inspect and calculate the projected structures and understand their relationships to existing structures. It helps to understand conflicts before the new structures are realized and help to prevent these conflicts. Traditionally, planning is done by using paper plans or desktop applications. The responsible construction site staff 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 allows to inspect planned objects directly in relationship to existing structures and allows to interactively modify the as planned objects to fit the physical occurrence on-site. There is no need to put additional effort to integrate modifications in the office. Furthermore, AR planning also allows to visualize planned structures in a way that external interested parties can be integrated into the decisions process since no special knowledge is required to understand the as planned information

Figure 2.21: Construction site monitoring using camera images. The image sequence represents different steps of the construction progress.

(Figure 2.20, 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 the 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 2.21). 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 usually available. This creates a high mental workload for the supervisor when he has to map the photographs to the actual construction site.

AR in combination with an adequate visualization 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 construciton plans on-site [119]. Furthermore, Golparvar-Fard et al. discussed AR visualization for supervising the progress on construction sites within the scope of the 4DAR project [36, 37]. Instead of presenting the actual data of each point in time, the system computes a single value, such as the current level of completion and display the value by using a color coding of the 3D real world object. While this approach allows 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. Whereas small commodity devices are designed to perform simple movements in the

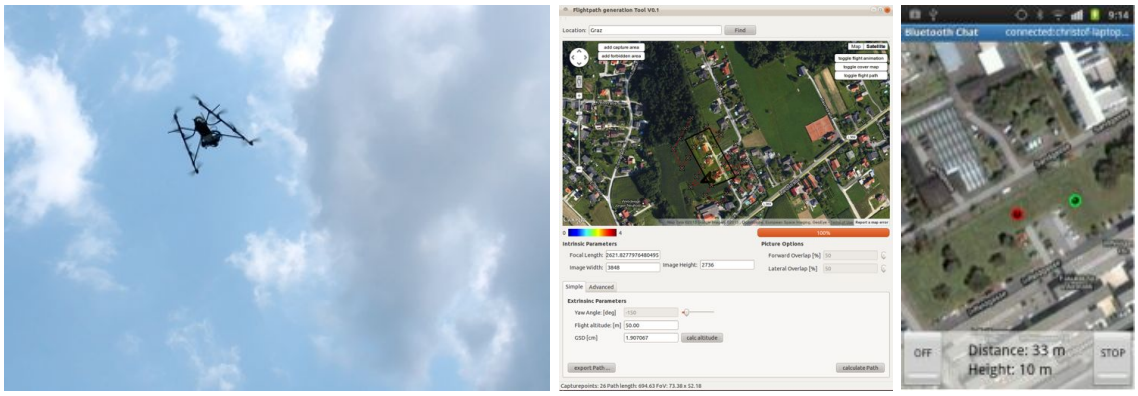


Figure 2.22: Map-based navigation interfaces for MAVs. Left) Desktop application shows flight path in red with white crosses. Right) Mobile applications can be used on-site by the user, but still requires him to map the 2D locations to the physical environment. The current position of the vehicle is marked with green, the next waypoint visualized in red.

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 [48], 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 [32]. Nevertheless, 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 2.22, Middle and Right) Right). However, the user still has to establish the spatial relationship 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 2.22, 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 [58]. 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.

Chapter 3

Methods and Systems

Contents

Before we can start to investigate different visualization techniques in outdoor environments, we need a system that allows displaying registered data and an provides an easy integration of new visualization techniques. This chapter describes how we achieve this by 1) discussing the registration methods we applied, 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 Augmented Reality is the registration technology. The registration makes sure that virtual objects are aligned to the physical world in a coherent way. There are different possibilities to achieve the registration, varying from simple maker-based registration to tracking methods that use natural-features, sensor-fusion and localization-based approaches. All these technologies use different approaches to achieve the same goal, aligning the virtual data in relationship to the real-world for a coherent AR visualization. For testing the visualization techniques in this thesis, we used different techniques to realize the registration. Marker-based and natural-feature-target-based techniques are often used for indoor AR applications. Nevertheless, for the outdoor applications they are usually no option. Thus, we had to integrate more sophisticated sensors into our setup and applied a sensor-fusion approach combining panorama-based tracking with an IMU and a GPS receiver. For application where a 3D model of the environment is available, we used a localization-based approach that is based on a server-client structure in combination with panorama-based or model-based tracking.

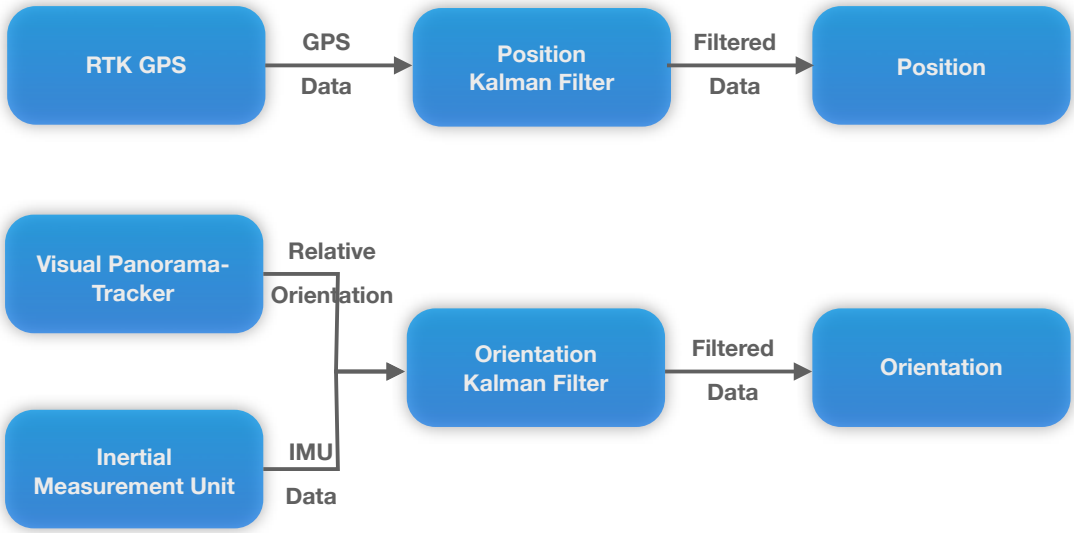


Figure 3.1: Multi-sensor fusion system architecture. The data from the GPS sensor is filtered by the *Position Kalman Filter*. The *Orientation Kalman Filter* fuses data from the *IMU* and the *Visual Panorama Tracker*.

3.1.1 Multi-Sensor Outdoor Registration

For outdoor applications marker-based tracking is usually not the first choice, since it is quite tedious to apply markers at different locations. Furthermore, normal paper-based markers are not very robust against weather influences. In an early stage of this thesis, we tried to use physical outdoor objects such as posters and facades as natural feature tracking targets [114]. Posters still showed an acceptable tracking performance, but for the most facades that we tested the results were unsatisfying due the low amount of features on the facades. This limits the application fields for natural feature target tracking in outdoor environments and we had apply more reliable technologies.

Typically, for touristic and entertaining AR applications in outdoor environments researchers and companies use a combination of the built-in sensors of a mobile phone or a tablet computer. For professional applications the accuracy that can be achieved with these kinds of systems is not sufficient. These sensor show positioning inaccuracies in the range of several meters and for the orientation they show errors in the range of several angles [96]. In order to provide accurate overlays of the virutal data, we implemented 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

Figure 3.2: Panorama generated by panorama tracker.

In a lot of outdoor AR applications Global Positioning System (GPS) is the 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. To address this problem we use a professional GPS receiver in combination with a Kalman filter.

For achieving high GPS localization accuracy, the GPS receiver performs dual frequency measurements and applies RTK for accurate positioning. We use correction data from one of our industrial partners (WienEnergie AG) for differential corrections. With a update rate of 1fps a reference station delivers the correction signal to the device in RTCM 2.3 format. 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 used an Open Source NTRIP (Networked Transport of RTCM via Internet Protocol) application (GNSS Surfer[1]).

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 [98].

For estimating their orientation, AR systems often use **inertial sensors** with gyroscopes, magnetometers and accelerometers that measure orientation with respect to local gravity and the magnetic field. 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 the Inertial Measurement Unit (IMU) with relative measurements provided by a vision-based panorama-tracker [98] to avoid these kind of problems.

The **panorama tracker** is based on feature detection and matching and builds up a panoramic representation of the environment that is stored for further tracking. Since 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 [22], the idea is to 1) determine the pose of a new frame relatively to already mapped data by using feature extraction and matching against the environment map (panoramic image) and 2) adding features from this newly localized frame to the existing map that is then again used for further pose estimation [61, 88].

By combining the measurements of IMU and panorama tracker in a Orientation Kalman filter as shown in Figure 3.1, we are able to achieve robust absolute orientation

[1] http://igs.bkg.bund.de/ntrip/download

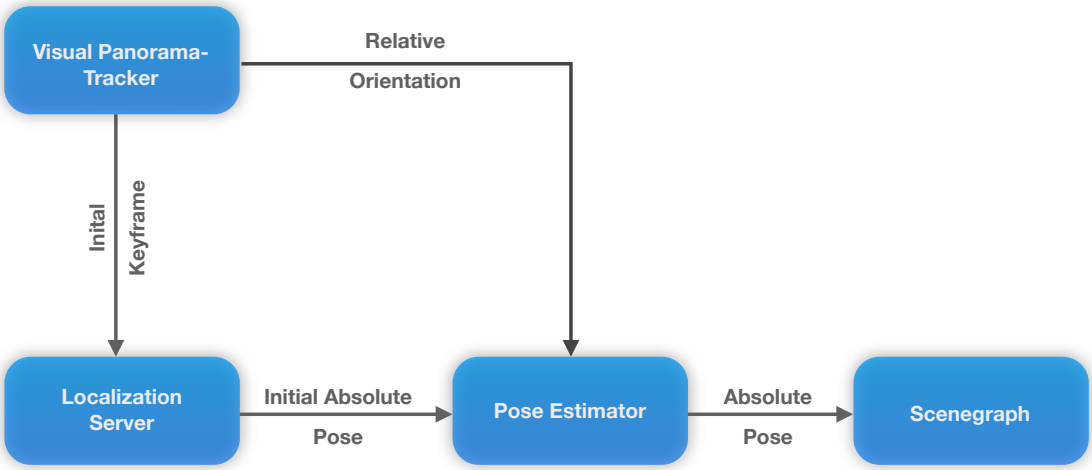


Figure 3.3: Model-based localization and panoramic tracking.

measurements that are mainly drift-free [96].

Other inaccuracies can result from the magnetic deviation, a value that describes how much the absolute orientation measurement differ from the 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.

3.1.2 Model-based localization

If an accurate 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. Accurate 3D models can be created by 3D reconstructions based on multiple camera images of the area that is of interest for localization. This means that the locations where a localization should be performed later on, have to captured in the 3D model. Additionally, by integrating newly localized images into the 3D reconstruction the localization area can be constantly extended.

To integrate a model-based localization into our system, we implemented a server-client structure based on the Robot Operating System (ROS[2]). On the client-site a visual panorama tracker or a model-based 6 degrees of freedom (dof) tracker uses 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 server, an absolute localization pose in relationship to the geo-referenced 3D model can be computed. The localization server sends the localization information back to the visual panorama tracker, that is then using this absolute information to update its relative orientation measurements. The accurate registration in relationship to the geo-referenced model allows rendering accurate AR visualizations (Figure 3.4, Right) as

[2]http://www.ros.org

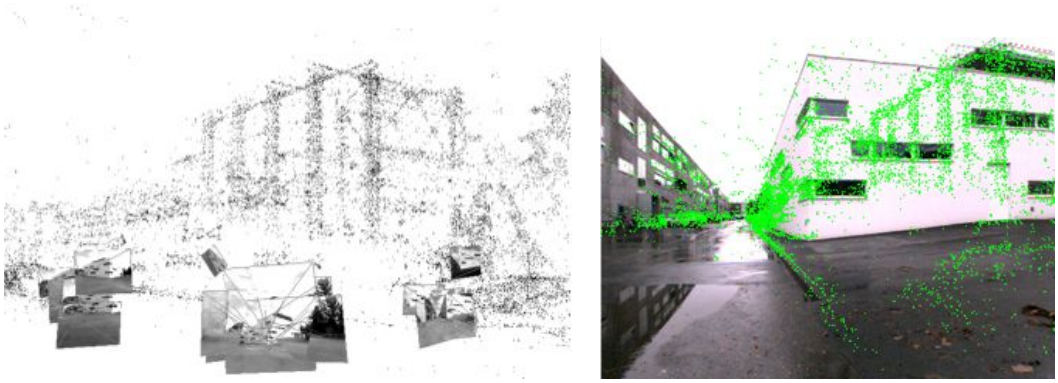


Figure 3.4: Model-based localization. Left) 3D point cloud with localized camera frames. Right) Localized camera frame overlaid with registered point cloud.

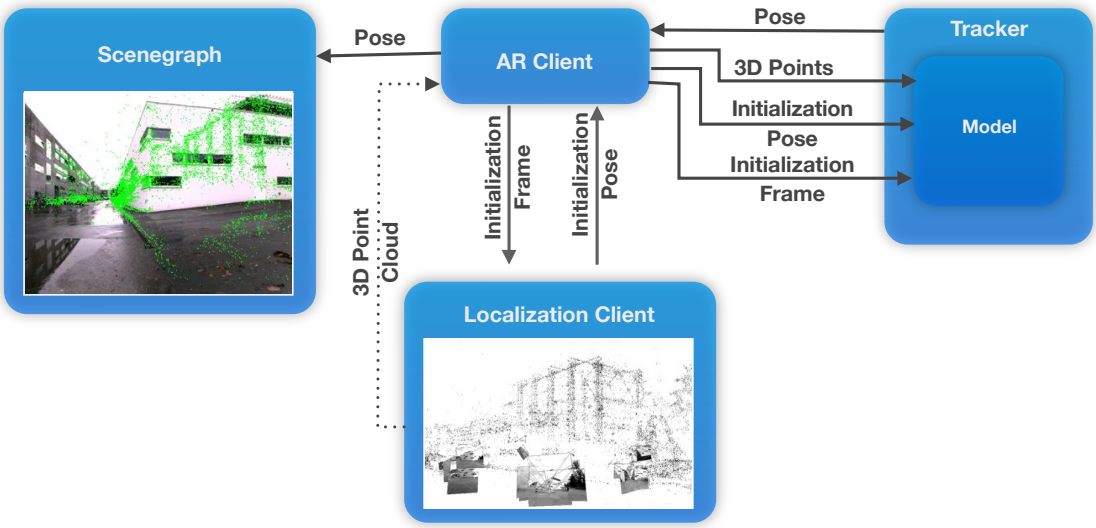


Figure 3.5: Model-based tracking. A localized camera frame and the 3D point cloud data is used to initialize a model for tracking. The model is then used for tracking and allows a accurate registration.

long as the user performs purely orientational movements. If the user performs translational movements, the motion model of the panorama tracker that assumes orientational movements, loses the tracking and starts to create a new panoramic representation of the environment with a new initial keyframe. Therefore, a re-localization has to be performed and the new initial keyframe is send again to the server (Figure 3.3). Sending of the image data as well as receiving the localization information is implemented in an extra thread.

Model-based tracker The model-based tracker is also based on the talker-listener concept of ROS. The AR client publishes an initial camera frame and waits for answers that contain a localization matrix of this frame in relationship to the geo-referenced point cloud. 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 localization 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 if new camera images come in. Therefore, 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 can then be used to compute an absolute transformation matrix. As long as we find enough correspondences between the model and new incoming camera frames, the initialized model can be used. Nevertheless, if the AR client moves to far away from the initial pose, there are not enough correspondences available and we have to compute a new model. The process starts again by publishing the camera frame and waiting for localization answers (Figure 3.5). The accurate registration in relationship to the geo-referenced model is then used to create accurate AR overlays (Figure 3.5, Left),

3.2 Data Sources

The main goal of this thesis is to develop visualization techniques for real application and professional data. Therefore, some effort has been spent in accessing and processing expert data such as data from *Geographic Information Systems* (GIS), Building Information Modeling and 4D data from aerial reconstruction.

3.2.1 Geographic Information Systems Data

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 mandatory. 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[3], 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.

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

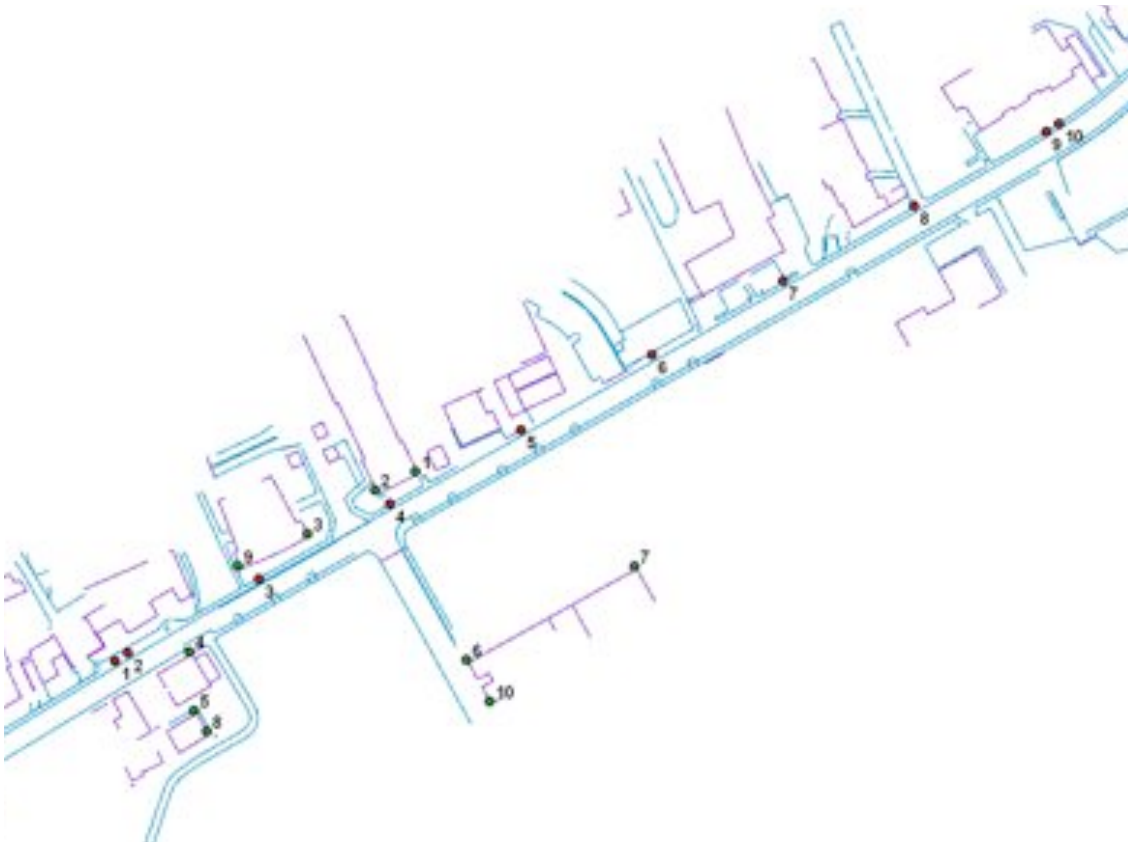


Figure 3.6: GIS information about a street. Purple represents building outlines. Blue shows curbstones.

Recent developments brought GIS tools to mobile devices for on-site inspection (e.g., ARCGIS for Android[4]). However, current visualization techniques implemented in these tools do not show the relation of GIS data to the real world context and still involve the tedious task of referencing assets correctly to the real 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 visual integration of real worlds and existing assets. The visualization of both, real 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 the 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, the data is extracted from the geo-database. For this purpose we use FME[5], an integrated collection of tools for spatial data transformation and data translation. FME is a GIS utility that help users converting data between various data formats as well as process data geometry and attributes. The user interactively selects objects of interest

[4] `http://www.arcgis.com`

[5] 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.

in the back-end GIS (Figure 3.6). These objects can then be exported to an external file by the FME software. For this purpose we decided to use a file format that is based on Geography Markup Language (GML), since GML is already widely used 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 real 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 digital elevation model (DEM) and known laying depths of subsurface objects. From this information we can compute 3D information for each 2D geometry.

This export to a file has to be performed offline before starting the AR system since it requires external software and interactive selection of the export area. To limit the supported data for our test applications, we designed a new GML-based format called 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 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 v3.1.1 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>
```

The generic format of the VidenteGML supports the description of a wide variety of objects without defining new schema for new types. This is a big advantage of the specific file format, since geospatial databases can easily consist of up to 3000 different feature classes from different sources (such as electricity, gas, water, sewer, heating or land register and topography) and it would be cumbersome to describe all those 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). 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[6] to implement the scene graph because it is easily extendable, but the approach can be easily adapted to be used with other scene graphs. 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]
```

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

```
    attribute_id ""
    attribute_name "enclosure"
    attribute_alias "NAT␣Gebaeudewand"
    attribute_group "topo"
    attribute_groupAlias "Basisdaten"
    attribute_mod "none"
    attribute_source "Grazer␣Stadtwerke"

    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 Data from building information modeling

Building Information Modeling (BIM) describe systems that combine different information about the life cycle of a building such as construction plans, but also plans for building management. Information from BIM can provide geometric as well as semantic information about the physical world context similar to information from GIS. Thus they can be a helpful source of information for AR visualizations. BIM data contains geometric as well as semantic information. The concept of BIM aims to provide 3D as well as 4D information for construction sites. This information could be directly used for visualization in AR. Nevertheless, in reality a lot of companies still work with 2D CAD plans. In this case, we have to apply a data conversion step similar as for the GIS data, to create 3D models that are capable for visualization purpose. Often BIM or CAD data is not geo-referenced, thus for outdoor usage we have to register them according to the physical world (For instance by using at least three point correspondences to a geo-referenced model and the Absolute Orientation Algorithm).

Figure 3.8: Aerial 3D reconstruction of a building.

3.2.3 Aerial Vision

In contrast to GIS and BIM that present either as-planned or a surveyed as-built status, aerial vision allows capturing the actual as-built status of a defined area. Nevertheless, due 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 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 a standard consumer digital cameras and capture images automatically from the area of interest [19].

Thereby, one flight session usually results in sets of 200-300 high-resolution and highly overlapping images. The set of overlapping images are used as input to a SfM approach [51]. The SfM approach computes the scene geometry as a sparse point cloud. Since the sparse geometry contains only limited data for visualization, we apply state-of-the-art

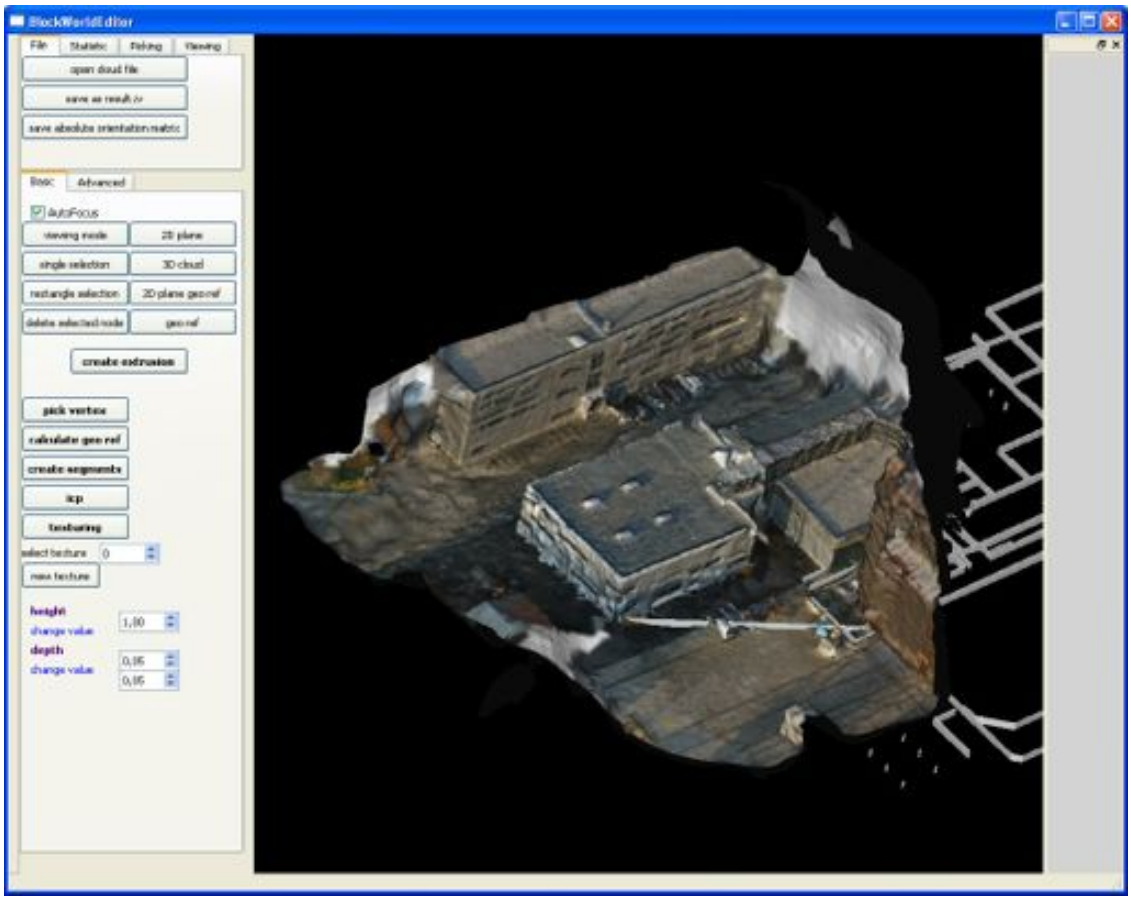


Figure 3.9: Interface for geometry abstraction.

methods for model densification. We apply the approach of Furukawa et al. to calculate a oriented semi-dense point cloud [34] (Figure 3.8). In order to obtain a 3D mesh, we use Poisson surface reconstruction [59] to create a mesh from the point cloud data. Additionally, we include available GPS information from the UAV into the reconstruction workflow to reduce computation time and to obtain a geo-referenced 3D model at metric scale.

Using this workflow we are not only able to create as-built data sets, but also to create 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 the initial geo-reference information enables 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 can often be to complex for understanding a visualization in AR. Methods for interactive geometry abstraction support the comprehensibility of a

visualization by converting complex data (such as 3D meshes) in a more simplified representation. An example are rectangular wall objects that are an abstract representation of a 3D point cloud of a wall.

For a lot of applications it is already possible to access this kind of abstract representation from data sources such as GIS or BIM (Building Information Models). For instance walls of building are often stored in GIS databases such as OpenStreetMap and even time-dependent representations are available for modern construction sites from BIM systems. However, such data is not always available, does not represent the time component or is simply not accurate enough. 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 an abstract representation.

The interactive geometry abstraction editor integrates a set of automatic and semi-automatic tools that support the creation of an abstracted visual representation from point clouds and a 2.5D plan (Figure 3.9). A complete manual modeling of this data would require a high effort, especially when it comes to data varying over time. Our tools allow the fast and intuitive creation of an abstracted representation with a minimal user interaction.

To create these abstract models, different data sources are required. The user can interactively select the data of choice in the editor such as:

- 2.5D as-planned data.
- Geo-referenced point cloud data of the as-built situation.
- Geo-referenced camera images that are registered to the point cloud data.

The 2.5D input objects are transcoded into a 3D representation by using an automatic extrusion transcoder. After the transcoding every polygon is represented as an extruded line with a certain height and depth. If such a 2.5D representation is not available another option is to create this kind of data interactively by outlining the contours of the wall objects with the mouse. Both data sets have to be registered to each other to be able to adapt the abstract models to the actual point cloud. Since the 3D point cloud data is usually geo-referenced, we provide a tool for performing a rough manual registration based on the *Absolute Orientation* method [49]. For this purpose the user has to select at least 3 corresponding points from the abstract transcoded data and the geo-referenced mesh data. Using the selected points a transformation matrix is computed. The transformation matrix is used to transform the as-built data to the as-planned data to allow a visual comparison. After achieving this kind of rough registration, automatic methods can be applied for adapting the 3D extrusions to the exact characteristics of the point cloud and texturing the 3D extrusions.

The editor offers manual as well as semi-manual methods to create an abstracted representation of the data.

Interactive Abstract Modeling Before the user can manipulate the geometric parameters of an object he has to select the object of interest. The selection is implemented by using the 2D mouse pointer coordinates and raypicking. Selected objects are visually highlighted. For the selected objects, the user can then either:

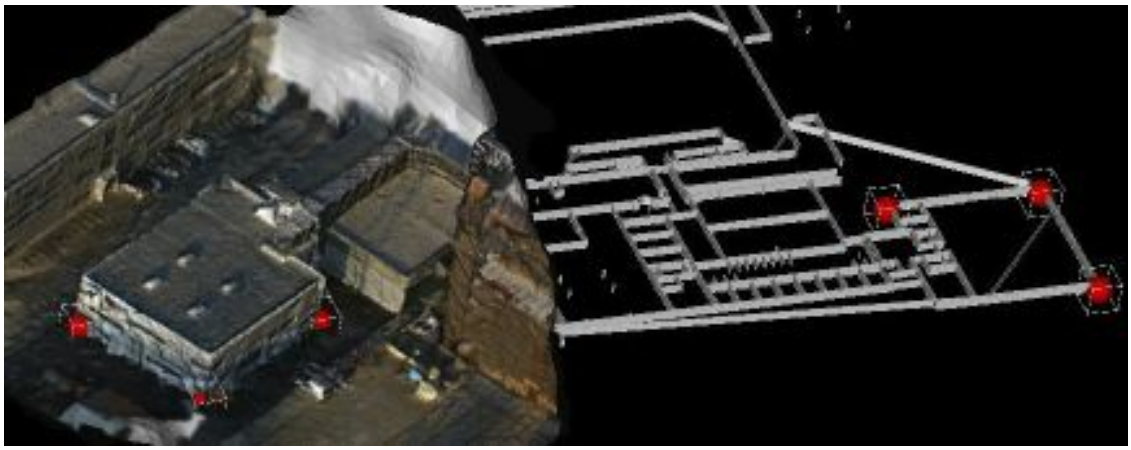


Figure 3.10: Computing the absolute orienation. Left) Selecting 3D points in the 3D mesh (red spheres). Right) Selecting 3D points in the CAD data (red spheres).

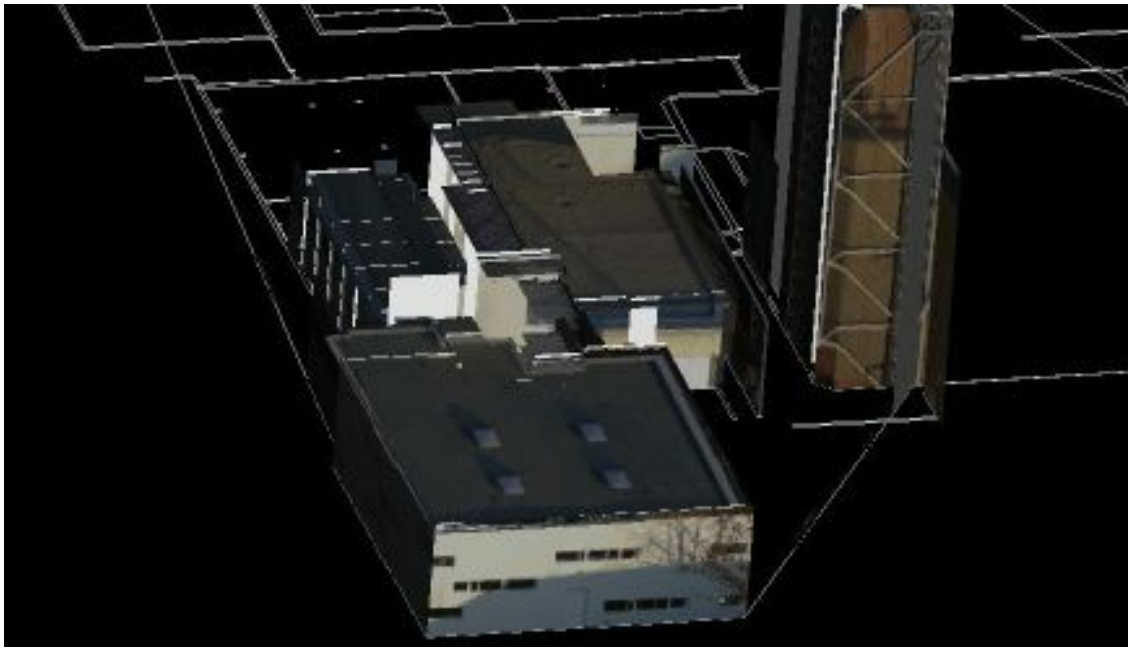


Figure 3.11: Results

- Manipulate the height of the extrusion.
- Manipulate the depth of the extrusion.
- Map selected textures onto the object and select the used camera for texturing.

To manipulate the height or the depth of the selected object the user can input the required value in the user interface (Figure 3.9).

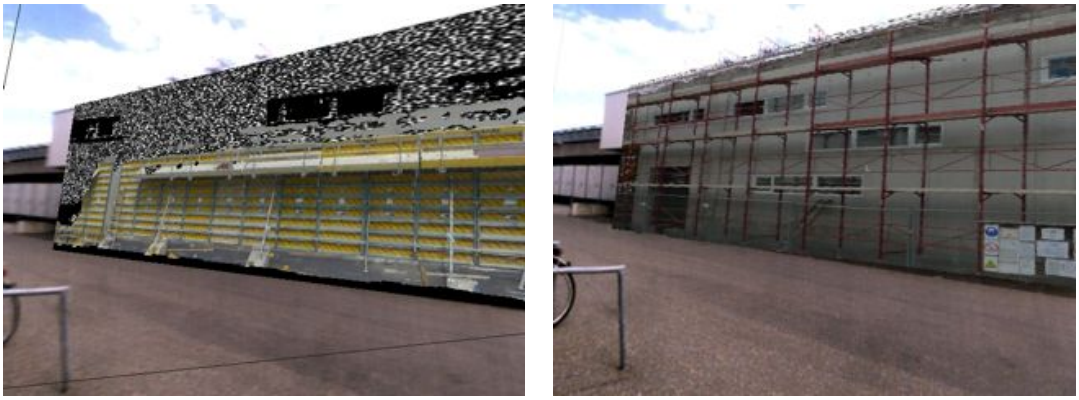


Figure 3.12: Abstract representation of former points in time.

Automatic methods for abstract modeling Additionally to the manual adaption of extrusion, we offer the possibility to adapt the extrusion automatically to the characteristics of the point cloud to reduce the manual effort to create these kinds of models. For this purpose we analyze the data of the point cloud and derive height, depth and width of abstract representations. These adaption methods are performed for all selected objects automatically.

Furthermore we integrated methods that allow an automatic texturing of the abstract models that is based on selecting the most appropriate camera for texturing. Additionally, the editor offers the following interactive methods for manual improvements:

- Extrusions manually textured selecting registered camera images.
- Creating and deleting extrusions.
- Manually increasing and decreasing high of extrusion.
- Exchange textures.

Results After performing manual and automatic adaptions with the interactive editor, the results of adapting a 3D point cloud is a abstract representation that consist of several 3D blocks (Figure 3.11). This kind of data can then be easily used in an Augmented Reality visualization to display former points in time in relationship to the physical world (Figure 3.12).

3.3 Mobile Augmented Reality Setup

For applying the visualization techniques in a real outdoor environment, we developed a system that is able to provide Augmented Reality visualization in an outdoor environment. For this purpose we equipped a ruggedized powerful 1.6GHz tablet PC (Motion J3400)

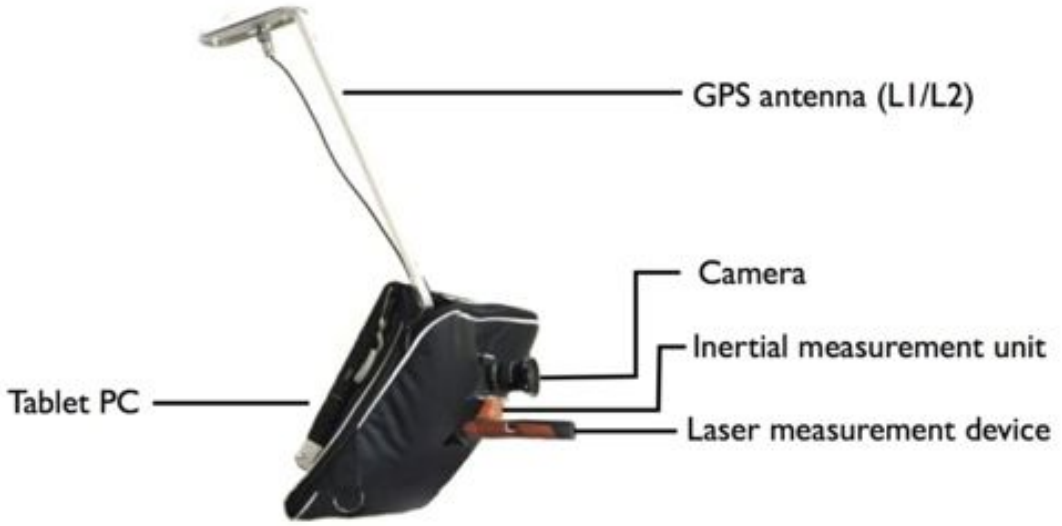


Figure 3.13: Augmented Reality Setup.

with a set of sensors for registration and data capturing. For supporting outdoor usage we used a tablet that provides a screen that specially build to be viewable outdoors even under sunlight conditions. The set of sensors consists of a camera, a inertial measurement unit (IMU), a laser measurement device and a GPS receiver. The camera is using 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 degree of freedom (DOF) orientation measurements. As GPS sensor we integrated a L1/L2 RTK receiver that provides measurements of the device's position within a 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. All sensors are connected via USB. A leather bag covers all cables and and the GPS sensor to prevent them from weather influences (Figure 3.13). 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.

During runtime the sensors are running in separate threads and are used to feed the registration methods from section 3.1 with input data. The output of the registration methods is used to update the transformation matrix of the virtual content. The rendering itself is performed each time a new camera image is arriving. For a simple blending of virtual content and camera image, we use a combination of OpenGL and a Scenegraph API (COIN3D[7]) to render the content.

[7] `www.coin3d.org`

3.4 Summary

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

The visualization techniques further strongly depend on the available data. Convincing visualizations are based on convincing data sources. For this purpose, we described different kind of data sources that can be used as input for visualization for professional outdoor applications. We showed how to access this data and how to convert it to displayable 3D geometries.

Finally, we showed 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 composition with the physical world by detecting and maintaining physical pictorial cues from camera images. If regular pictorial cues, which are usually present to the human observer in the perception of the real world, are not sufficiently presented in an AR visualization, the scene will either look unnatural or produce a wrong perceptions of the order of objects in the scene. Missing occlusion cues may lead for instance to perceiving virtual subsurface objects in an X-Ray view as being floating over the ground.

In chapter 2 we discussed the different cues that are important for for depth perception. Since we 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 we will note that something is wrong in the scene. The most of depth cues are already provided by the general rendering pipeline. But some cues have to be added additionally.

Occlusion cues are the strongest pictorial depth cues and work on all distances [20]. Unfortunately, they are not automatically provided by the rendering. Occlusion cues give an ordinal measurement of objects. Due to the power of occlusion cues, we will focus on

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.

these cues in the following, but we will also discuss some examples where we integrate shadows of virtual objects in the next chapter.

To provide an adequate occlusion management there are two main problems that have to be addressed. Firstly, if there is no accurate 3D representation of the scene available, the order of depth has to be estimated. Secondly, since augmented reality allows not only natural compositions, but also the possibility of having an X-Ray view in physical objects, it has to be decided which information of physical occluding object is preserved in the final rendering. This decision depends on the one hand on the visibility of the occluded objects but also on the minimum number of occluding cues that has to be preserved to achieve an 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 are preserving not only the appearance of the occluding object but also its structure. This is shown in Figure 4.1 where some random occlusion cues are used.

Finally, the compromise between visibility of occluder and occluded object will result in a visualization technique that is called *ghosting* in the literature [28]. The basic idea of ghostings is to preserve certain information from the physical object to be overlaid and render it on top of the virtual objects. This information can be derived in abstract forms such as edges, texture, reflections, curvature and so on. This then raises the questions what information should be preserved and in what amount (opacity). In former works, there have been model-based approaches that use a 3D representation of the occluder to determine this information [54]. The disadvantages of model-based ghostings are often that no occluder model exists or that the registration of the model is not accurate. Even if a perfectly registered 3D model of the occluder exists, the exact texture of the model may be missing, which makes the determination of the adequate amount of preserving impossible. For the case of exact registered data Mendez et al. proposed a method which is based on pre-defining a mask for context preserving [79]. There have also been attempts at using edges extracted from camera images in the case that there is no model of the

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.

occluder available [55].

In this chapter and the following chapter, we will describe different techniques that preserve physical pictorial cues automatically from different available data sources. Depending on the available data source the methods vary in their flexibility.

1. Only video information about the environment is available.
2. A sparse 3D representation about the environment is available.
3. A combination of sparse 3D representation and video images about the environment is available.

The method that we will describe in this chapter extracts occlusion cues solely from video images and is based on the assumption that the depth order between virtual objects and physical world is given. This assumption is valid for scenes where the order of objects is given due to semantic scene knowledge as for example it is the case for subsurface X-Ray visualizations. In chapter 5 we will show methods that use a sparse representation of physical objects in the scene for creating physical pictorial cues. Combined with object semantics these sparse representations allow to preserve real world objects that are defined as being important in the scene structure.

In this chapter we introduce a solution to the problem of deciding which information of the physical environment should be preserved by analyzing camera images and heuristically extracting key information. This technique does not require the presence of a virtual model of the occluder if the depth order of objects is known. Instead, 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 surface of a table), we add synthetic details that maintain physical characteristics [87] for compensation. This approach assumes that 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. Figure 4.2 contrasts a naive 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 section 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 [46] already address the conflict between showing hidden structures and preserving the context of the foreground. Artists identify the important structure of the scene and preserve them in the illustration.

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

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 [14].

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 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 not usually given in X-Ray augmented reality applications. Notably, 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. The determination of the image regions used for the analysis is 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 [84]. High-level factors consider previous knowledge, goals and tasks. An example for high-level importance are users 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 for many scenes. Thus in general, high-level factors based on previous knowledge, cannot not be used. In contrast, low-level or bottom-up factors are stimulus-based for fast information processing. Osberger et al. [84] identified several low-level factors; and used them in MPEG encoding to code visually important regions more accurately. These comprise

1. **Contrast**. Since the human visual system converts luminance into contrast at an early processing stage, contrast is a strong low-level visual attractor and regions with high contrast are likely to have high visual importance. Contrast can be analyzed at a global scale, at a neighborhood scale and at a regional scale.

2. **Shape**. Edge-like shapes have been found to have a high visual attraction. It is more likely that they attract attention compared to regions of the same contrast but with other shapes.

3. **Color**. It has been found 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**. It has been shown that the size of a region effects its visual importance. Large regions are more likely to call attention.

5. **Motion**. Motion has one of the strongest influence in attracting attention.

Other low-level factors include brightness, orientation and line endings. 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 principles describe 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 stereo reconstruction and segmentation. 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. To achieve these groupings we apply a superpixel representation of the image, since superpixels satisfy the demand to perceptual grouping in

images [89]. A useful side effect of using superpixels instead of pixels is that meaningful statistics of these groups can be calculated and the per-superpixel processing is faster than a per-pixel processing. The pixel dependent parameters $p_i(x, y)$ are then extended to be dependent on the corresponding superpixel r. The function $sp(x, y)$ maps a pixel to its corresponding superpixel r:

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

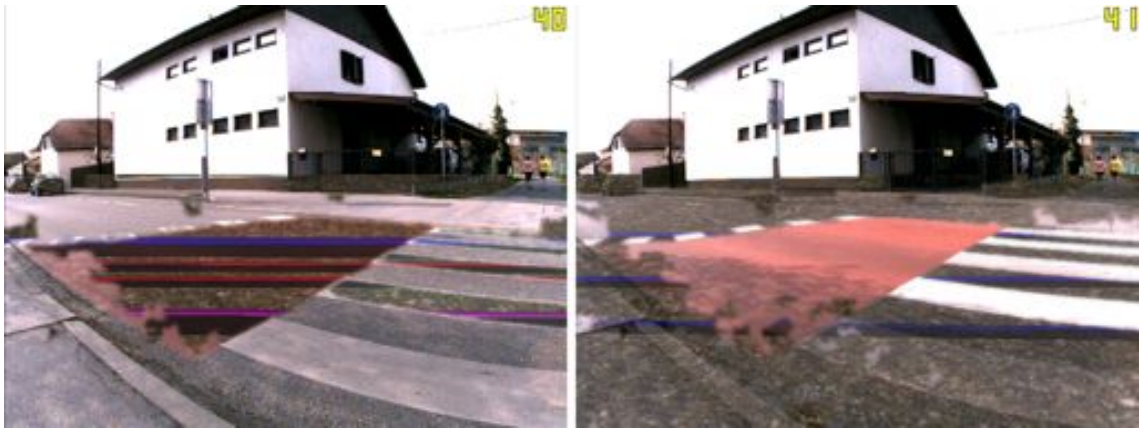


Figure 4.3: Different grades of preservation. Left: The road markings are completely transparent. Right: While the road markings are opaque, texture details of the concrete are extracted and preserved.

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 scene. Thus it is important to allow the user to adapt the visualization interactively according to their preferences. In their context-preserving approach, Bruckner et al. introduced parameters which allow the user to interactively explore the data sets [14]. For example, the user controls the revelation of the interior of the volume data by modifying one of the parameters. We want to apply a similar concept for our augmented reality ghosting. Therefore we introduce a parameter $g_i(r)$ which 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.

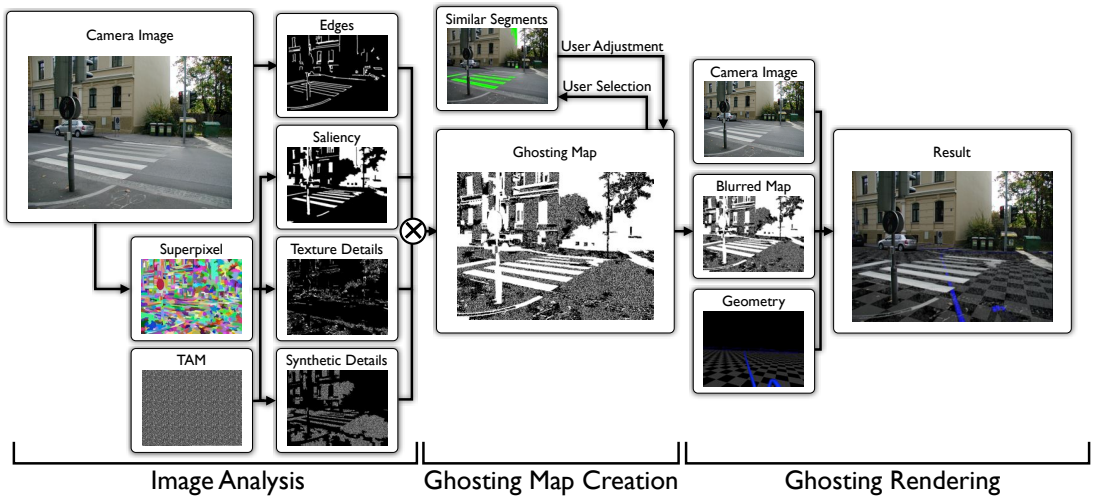


Figure 4.4: Overview of image-based ghostings. The camera image is analyzed for important information that has to be preserved. Important information includes edges, salient regions and texture details. In regions without important information, synthetic structure from tonal art maps (TAMs) will be preserved. 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 Image-based Ghostings

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 camera image is then 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 preserved. The result of the analysis for context preserving is the ghosting map, which is adjustable by the user by selecting reference segments in the image. The 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).

4.3.1 Importance of image regions

Firstly, the image is analyzed to find evidence for the importance of each pixel to the overall scene. In section 4.2.2 several indicators for the importance of image regions were introduced. For our approach we have decided to use contrast, shape and color as importance measurements, because they can easily be computed from one image, without knowledge about image objects. Although we only use a subset of features, the approach is easily extendable to more parameters.

Figure 4.5: X-Ray visualization of a virtual room inside a building. The left image shows the simple overlay, where the room is more perceived to be located in front of the building. The right image shows the usage of 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.

Edge-like features are highly important for scene understanding to the human visual system (section 4.2.2). For this reason we compute edges in the image to identify visual important parts of the image. We apply the Canny edge detector and use edge chaining to extract connected edge segments [16]. 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 by using the average image color of a superpixel $I(r)$:

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

We use the CIELAB color space to represent the color vector, because 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

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).

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:

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

- $E(x, y)$ 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

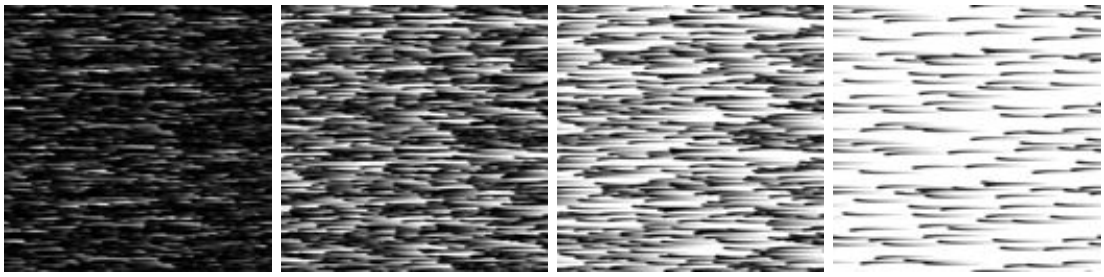


Figure 4.7: Tonal art maps with hatchings. Subsequent textures contain same elements to provide smooth transitions over the intensity ramp.

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

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 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. [50]. 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 [46]. Smooth transitions between regions with different intensity levels are provided by pre-computed *tonal art maps* (TAM) [87]:

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

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 dots 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. To reduce high frequency artifacts, the output of the ghosting transfer function is smoothed by a Gaussian filter.

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

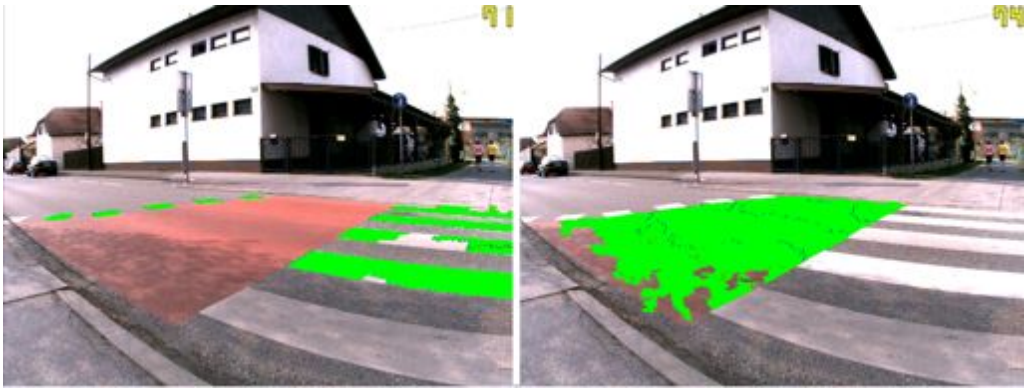


Figure 4.8: Examples of the selection of similar image regions. Left: Selection of white street markings. Right: Selection of red street markings.

equation (4.5). Whereas 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 (7.3) and (4.14) to change the amount of details. We define the range of the parameter to be 0 to 1. Therefore, large values of g_d preserve more texture details, because the threshold for using a pixel as texture detail is getting lower. Similarly, the offset of equation (4.14) is increased and the selected level of the tonal art map is higher. Thus, the density of the synthetic structures is higher and 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.

To change the generic transparency parameter, the user can select one image segment and adjust 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. After selecting a set of regions, the user increases or decreases the parameter g_d to change the amount of video foreground that is preserved.

To find similar regions, we use a descriptor of each region, computed from some of its normalized characteristics. 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. Regions with a descriptor distance to the selected segment descriptor below a certain threshold, will be classified as similar and will be added to the selection (see Figure 4.8).

4.4 Implementation

We tested our method with the mobile outdoor visualization system described in chapter 3. For this application we used the multi-sensor fusion approach (chapter 3). In the following we discuss how we integrated the ghosting method into this system for visualizing underground infrastructure.

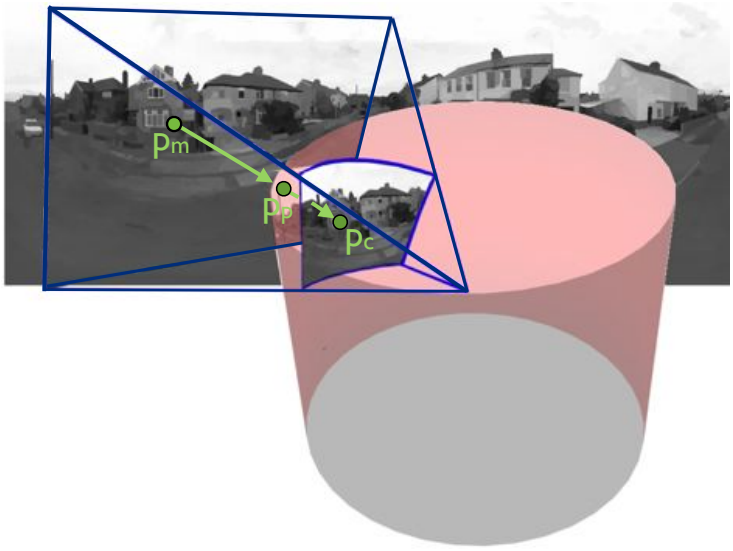


Figure 4.9: Panorama remapping. This image shows the mapping from the panoramic map over the panoramic cylinder into the current camera image.

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 partly on the superpixel representation. We decided to use the EGBIS algorithm of Felzenszwalb et al. [29] 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 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 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 in the video frame is projected into a global 3D ray and intersected with the panorama cylinder (see Figure 4.9). The corresponding pixel in the panorama ghosting map is read and mapped into the video frame. Bilinear filtering is used to avoid aliasing in the resulting map.

The selection of regions to change the transparency parameter uses the same projection. Here, the user selects a reference superpixel by mouse click in camera image space (see section 4.3.3). The mouse coordinates are also projected from image space to panorama

map space given the current camera orientation. The coordinates in the panorama image are then intersected with the list of superpixels and the superpixel at this position is selected.

While the main advantage of the panorama-based method of creating the ghosting is that it allows real-time processing and preserves frame-to-frame coherence in the ghosting, the drawback is 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.

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 into video image
Create ghosting image from video image and ghosting map
Render 3D model
Render ghosting view with blending.
```

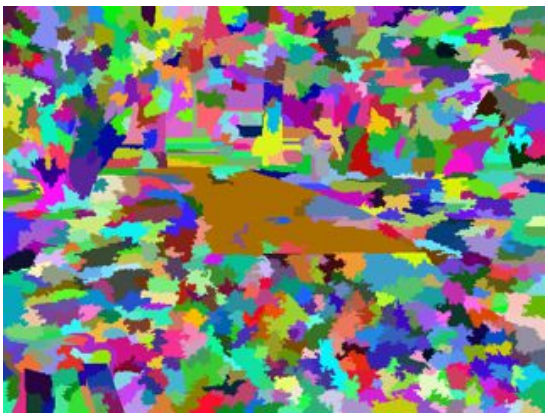
Video image pixels are rendered with an alpha value between 0 and 1. The alpha value is defined by the ghosting mask.

4.5 Results

We applied this method in the project SMARTVidente for subsurface infrastructure visualization. Underground infrastructure such as electrical and communication cables, water and heating pipes; and general features of the environment, are shown to users in a registered AR display. Usually, no surface structure model of the environment is available for these kind of applications, therefore 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) that determines which camera image pixels have to be preserved or replaced by virtual content. The final compositing using the ghosting map, virtual content and the video image is 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 more

(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.

occlusion cues. Figure 4.5 shows another example of using image-based ghostings. Instead of displaying underground infrastructure information, in this picture the room inside the building is made visible. In this case the simple blended overlay (Figure 4.5, Left) could lead to the impression that the room is virtually 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. Thus 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

Figure 4.11: Field trials with expert users.

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. Whereas 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 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 Expert interviews

To get a first idea of the applicability of our approach we asked professional users from the civil engineering industries about the image-based ghosting visualization in a surveying task. Therefore, we collected data in a field trial from 16 participants (12m/4f) using a questionnaire. From these participants, eleven had experience in traditional surveying techniques and five users had only theoretical experience. All participants had to survey an object in the physical world with the interactive surveying application. Afterwards we used questionnaires and a general interview to learn about the user experience. From the expert interviews we learned, that the participants confirm the high potential of AR for time savings and error avoidance for the task planning and surveying, but that they also have a high expectations on the accuracy of the system. Another important finding from the interviews was that most of the users gave a high priority to a correct depth perception. In the questionnaires the experts were asked to rate the system using a 7-point Likert scale. The visualization techniques "image-based ghosting" was rated above average (avg. 5.87, stdev 1.02).

Figure 4.12: Test scenes for the survey. Left) Condition Alpha Blending A, Middle) Condition Edge-based Ghostings G_E, Right) Condition Image-based Ghostings G_{IB}

4.5.2 User Evaluation

In addition to the expert interviews, we investigated the depth perception of users in an more extensive user study. For this purpose 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) [5]. The goal was to investigate if the image-based ghostings are performing 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.

Hypotheses We hypothesized that participants understand the subsurface location of virtual objects better using the image-based ghostings (G_{IB}) than using alpha blending or the state-of-the-art 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 Edges 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 tests 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. To achieve these goals we decided to prepare a set of static AR scenes in advance using the 3 different visualization techniques. This further has the advantage of being 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

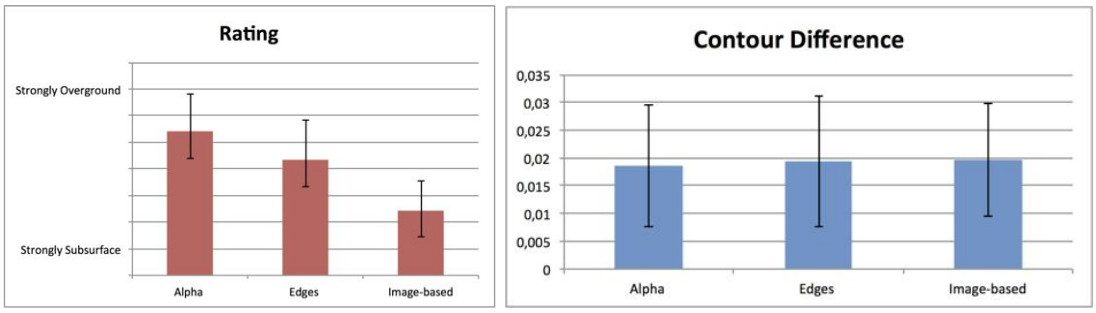


Figure 4.13: Results from the pilot study.

types of content, content that belongs to a scenario of inspecting subsurface infrastructure (red pipes) and abstract content (red spheres with different sizes).

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 a 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 pressing on the 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 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 result in the contour difference that was used to determine the ability of users to correctly understand the shape of the object.

This task was repeated for 12 different scenes using the same visualization technique but showing different content. After finishing these scenes, the user was asked about his 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. The overall study duration for each participant was approximately thirty-five minutes.

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

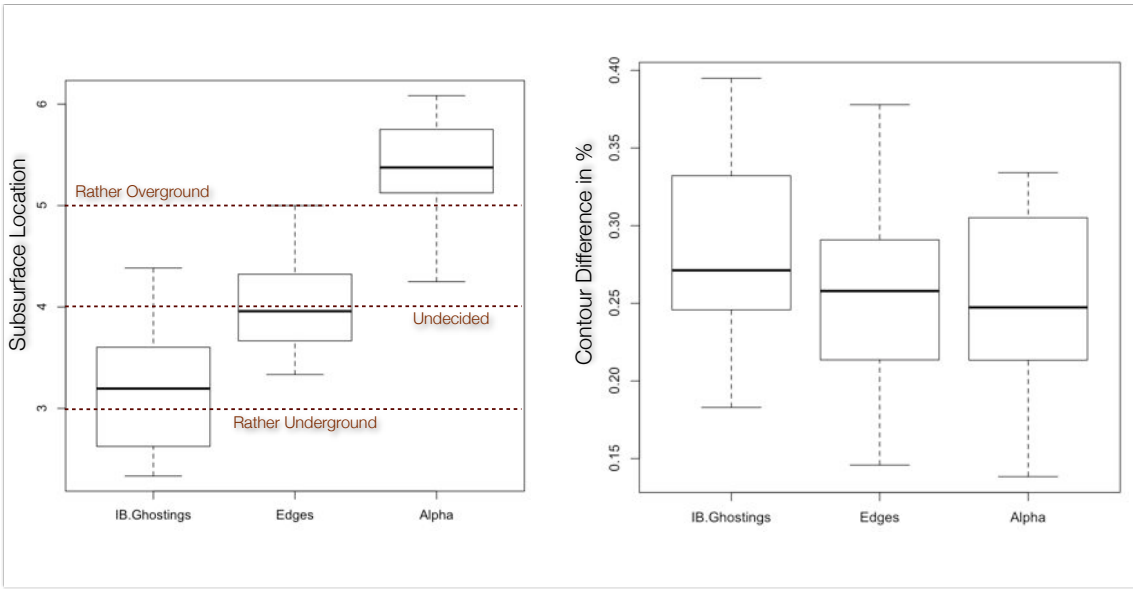


Figure 4.14: Results user study. Left) Results of the depth perception task. Right) Results of the accuracy test.

remove the abstract shape condition for the contour drawing since there the subjects reported that these shapes were too simple and to easy to complete and on the other hand quite exhausting since they were spheres.The pilot also showed that the participants seem to perceive the subsurface objects more being located underground while using the technique G_{IB} (compare Figure 4.13, Left , average rating 2.43). Contrarily, for G_E they seem to be rather undecided (average 4.33) and for A they seemed to rate the location more 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. $\vec{a}_1$

Participants We invited 12 people from different universities to take part in this experiment (5 female, 7 male, age ranging from 22 to 35). 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 47.82 with a p-value nearly 0. We can clearly reject the null hypothesis of equal means for the overall depth perception rating of all three visualization techniques.

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 = 3.0, compare with Figure 4.14, Left) of the subsurface location of the virtual objects than the simple blending A (M = 5.0, A - G_{IB}: p= 1.0e-10) and state-of-the-art ghosting G_E (M=4.0, G_{IB} - G_E: p = 0.0021). G_E also performs better than A(A-G_E: p= 3.3e-06). 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 will have a stronger perception that objects are located subsurface.

The output of the ANOVA for overall contour deviation shows that F= 1.129 and has a p-value p=0.336. We cannot reject null hypothesis of equal means for the accuracy of outlines during usage of the three visualization techniques. This means 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.

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 augmented image. 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

In this chapter, we continue the work on creating physical pictorial cues automatically. In contrast to the methods of chapter 4 that use no additional source of information but the camera images, we will describe methods that benefit from a sparse description of context information, such as given by a GIS database. We will discuss methods that directly use 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, in a lot of cases the sparse information is not sufficient for providing a pictorial cues since it misses physical world detail. We will address this problem in section 5.3 by investigating methods that allow supplementing sparse information with additional data from camera images. If an accurate registration is available, we can combine the sparse representation with the video image representation of the physical world to derive more accurate occlusion cues from the environment.

5.1 Introduction

Professional geographic databases such as GIS provide a rich set of information about physical world objects. Whereas in the last chapter we only used this information to visualize pipes and cables that are buried in the ground and therefore not visible in a

normal view of the physical world, in this section we will show how digital representations of physical world objects can be used to derive physical pictorial cues.

5.1.1 Sparse Geometries as Physical Pictorial Cues

We can support the scene comprehension by rendering sparse geometric representations of important physical objects in addition to the virtual invisible 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 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
- Manholes

In the composition of 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 geometries representing physical world objects.

GIS databases contain a huge amount of information. Thus 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 occluded objects. The advantage of this method is that it is still working in cases if the accuracy of the data is slightly offset. A drawback of this visualization technique is that the user has to mentally restore the relationship between the virtual representation and the physical world object.

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. For this purpose, we compute an occlusion mask M containing the sparse geometries. The final image is then composed by by $C = (1-M)V + P$. Where V is the virtual subsurface geometry, and P is the camera image representing the physical environment. We use the stencil buffer to compute the masking. 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 location where the mask entries are positive. Finally, this results in the composition C as shown in Figure 5.2 (Right and Middle) where at pixels representing the curbstone the video image is rendered instead of the virtual hidden pipe.

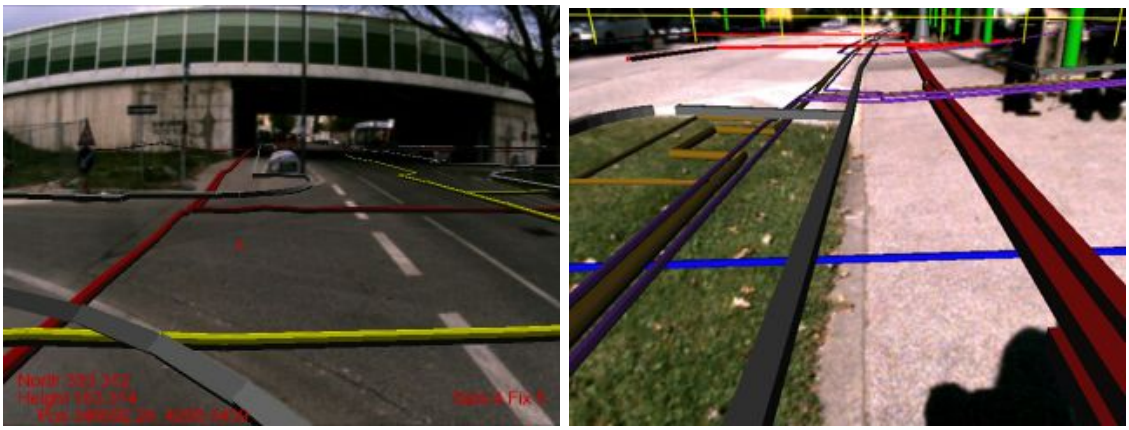


Figure 5.1: Virtual representations as physical pictorial cues. The virtual counterparts of the physical curbstones and grass boundaries are rendered in grey on top of the occluded pipes.

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

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 information in the database since otherwise important occlusion cues are missing.

When we use physical object representations for occlusion culling, it may happen that too much information of the underlying content is discarded. We can create dynamic occlusion masks instead of static occlusion masks to avoid this problem. Instead of rendering the video image completely opaque at the location of important physical world objects, this method generates a importance mask that varies over time. A method that uses such dynamic occlusion masks was introduced by Mendez to reveal the hidden content of a paper box [78]. 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 [86] to create a dynamic mask. The advantage of this technique is that it reveals hidden structures over time, so the user can see different parts of the hidden object at different points in time. Furthermore, it does not depend on the amount of features of the

Figure 5.3: Using sparse data in combination with Perlin noise for occlusion culling. (Left) Perlin noise mask mapped to the objects of interest. (Right and Middle) 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.

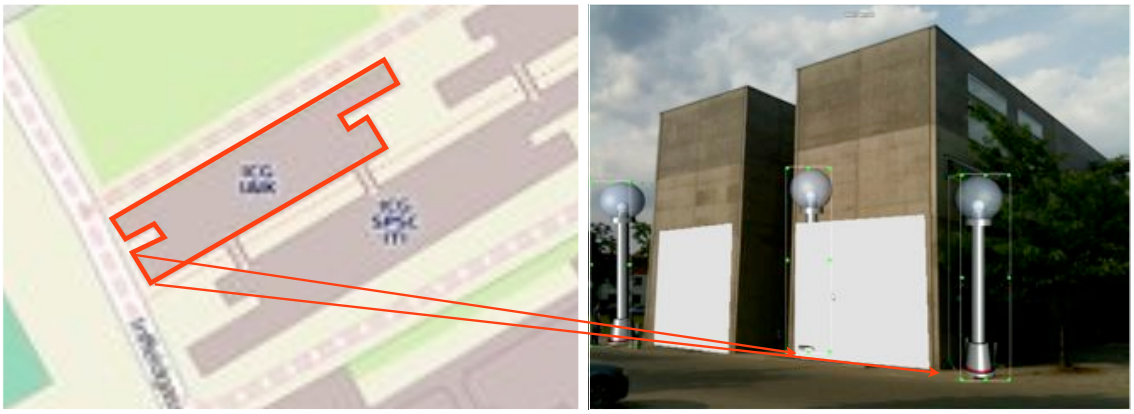


Figure 5.4: Problems in occlusion management resulting from inaccurate GIS data.

occluding object, thus it can also be applied for less textured regions.

The previously presented visualization techniques often suffer from the sparseness of the GIS data. Objects are represented by simple 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, it only represents the point or line on the surface of the earth but does not describe the 3D shape. We refer to such models as *sparse models* (Figure 5.5, (Left)) similar to sparse point clouds created from image-based reconstruction. If the data is too sparse, for instance if information about the height, dimensions or extend 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

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.

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.

5.2 Background

Creating 3D model information from images is a traditional research area in computer vision. Unfortunately, many techniques require specific data such as multiple images from meaningful viewpoints [**?**] or specialized hardware such as stereo cameras or depth sensors to create dense depth maps. Thus, 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. [47] 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 [112]. Another technique

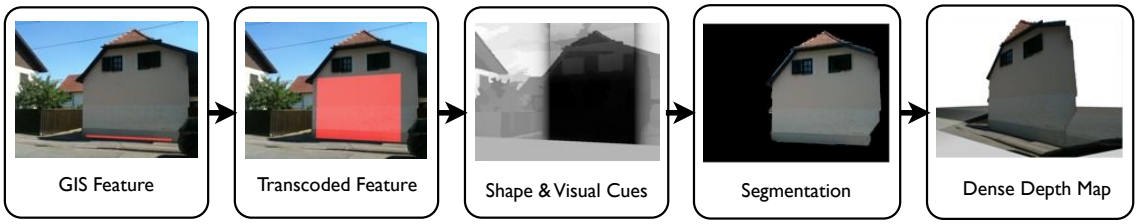


Figure 5.6: Overview of our approach. The AR system provides a registered background image, which is combined with GIS model data to derive visual and shape cues and guide a 2D segmentation process. From the segments and the corresponding 3D location a dense 3D depth map is formed.

to create pop-up objects was introduced by Wither et al. [116]. 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 these kind of 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 [106] or prior knowledge about the architecture [23]. Grzeszczuk et al. replaced the interactive input with information from GIS databases allowing for a fully automatic computation of compact architectural 3D models [42]. 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 AR view is rather rare [**?**]. Thus, it is not possible 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 [94]. 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. (3) The segmentation process returns a 2D image area that corresponds to the 3D model feature 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. Thus, 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 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 they provide 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 methods 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 inform a foreground color model for the segmentation
- Additional energy terms to steer the segmentation

Support Shape The GIS contains only sparse and abstract geometric representations of real 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 geo-referenced positions 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 [95]. 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 real 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. However, projecting the transcoded geometry into the image allows for sampling pixels in the coverage area to extract a foreground appearance model of the object's segment. This model can include gray values, color values in different color spaces, texture information, or binary patterns as described by Santner et al. [94].

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. [29], 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 such terms is the indicator function for the projection of the support shape itself. However, a

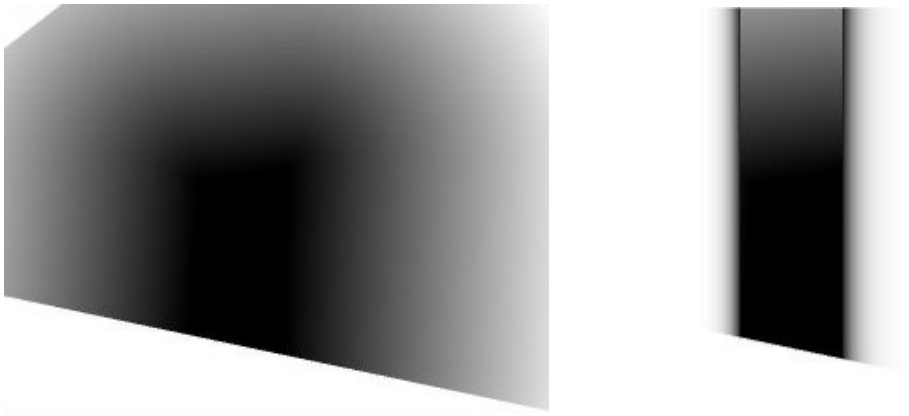


Figure 5.7: Distance-transform of shape cue. Left) Isotropic distance transform of shape cue. Right) Anisotropic distance transform of shape cue.

sharp boundary would defeat the purpose of the 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.

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 objects corresponding to the data given by GIS. 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

$$f_{vis}(p) = \|V(p) - V(S)\|^2. \tag{5.1}$$

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

$$f_{vis}(s) = \min_{n \in O} \|V(s) - V(n)\|^2 \text{ where } n \text{ neighbors } s. \tag{5.2}$$

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 5.2) to a single cost function $C(s)$ for the segment s

$$C(s) = f_{shape}(s) + f_{vis}(s), \tag{5.3}$$

and thresholding the cost function for determining the label:

$$L(s) = \begin{cases} 0, & \text{if } C(s) \geqslant T \\ 1, & \text{if } C(s) < T. \end{cases} \tag{5.4}$$

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 [110]. Santner et al. successfully applied Total Variation

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.

for interactive image segmentation [94]. 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 if it is foreground (as 0) or background (as 1)

$$\min_{u\in[0,1]} E(u) = \int_\Omega |\nabla u| d\Omega + \int_\Omega |u - f| d\Omega. \tag{5.5}$$

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:

$$f(x, y) = \alpha f_{shape}(x, y) + (1 - \alpha) f_{vis}(x, y) \tag{5.6}$$

$$= \alpha f_{shape}(x, y) + (1 - \alpha) \|V((x, y)) - V(S)\|^2. \tag{5.7}$$

$V(S)$ 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 (5.5). 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)).

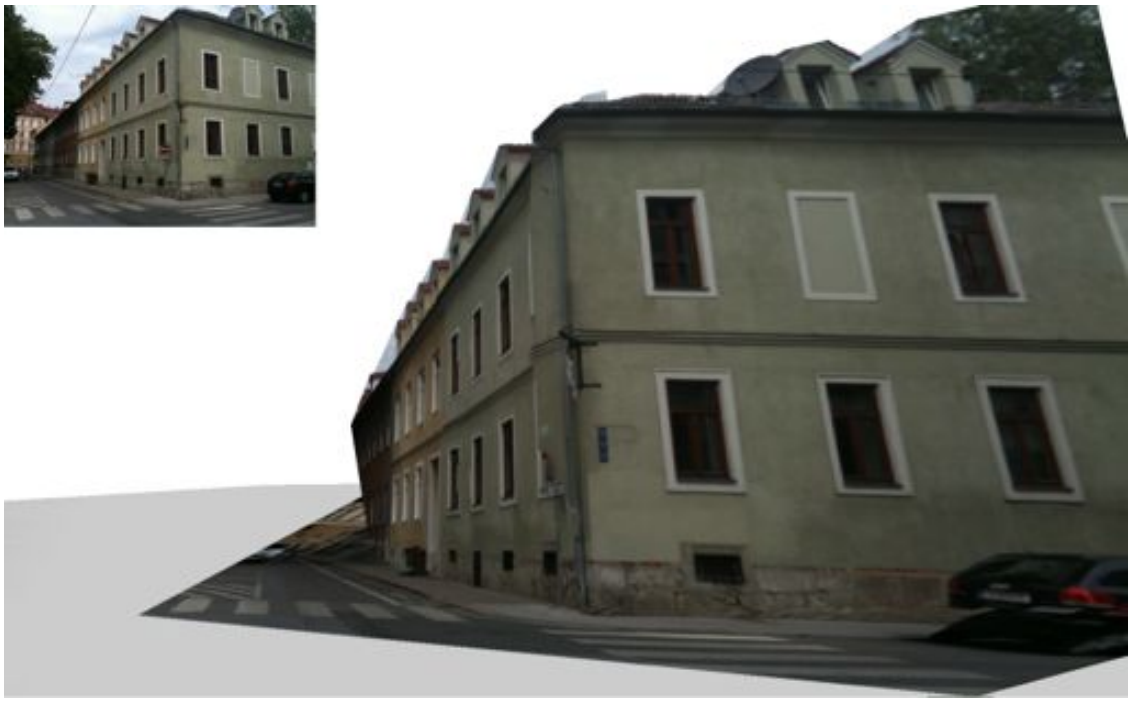


Figure 5.9: Extracted pop-up model created with our method for the image shown in the inset.

5.3.3 Geometry Creation

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 use the support shape extracted in section 5.3.1 and compute the final geometry by back-projecting the outline of the segmented object onto the support shape. First, the outline of the segmentation is extracted from the image by 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. 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 that 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 5.10). As input for the projective texturing mapping, we set the texture matrix up to use the same camera model as the calibrated real camera.

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 on the camera image.

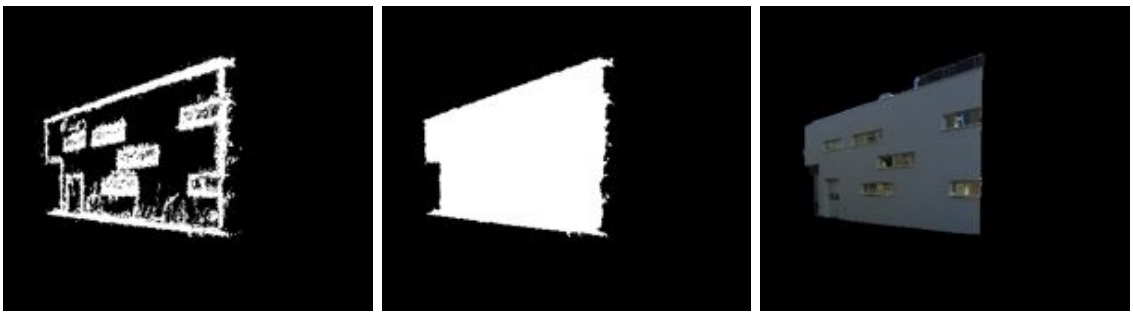


Figure 5.11: Computation of segmentation error. (Left) Reprojected 3D points of the point cloud. (Middle) Computed filled polygon of the reprojected points. (Right) Extracted segmentation.

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.

Accuracy We tested the accuracy of the dense depth estimation against a 3D ground truth model. As ground truth geometry we used a 3D reconstruction created by SfM based on a set of aerial images acquired with an octo-copter [51]. Unfortunately, the result of the SfM method is rather sparse. We addressed this problem by applying the method of Furukawa et al. [34], to compute a semi-dense point cloud. Known GPS positions of the octo-copter allow geo-referencing the point cloud and aligning it with known GIS data. As input to the dense depth estimation we use a geo-referenced 2D CAD plan of the the building provided by a construction company and geo-referenced images of the building.

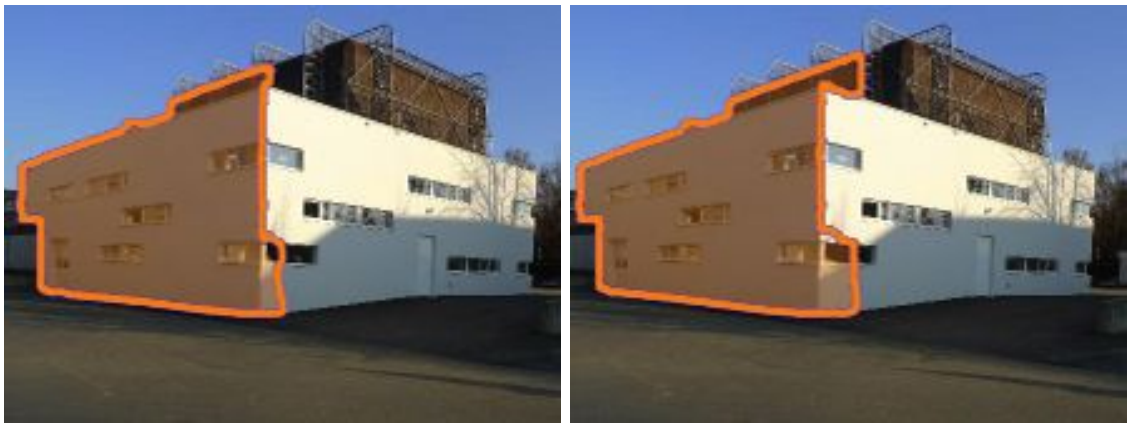


Figure 5.12: Segmentation results for selected offsets. (Left) Segmentation result for 0.5m offset. (Right) The segmentation with an 1m offset shows a larger error but still a visually acceptable result.

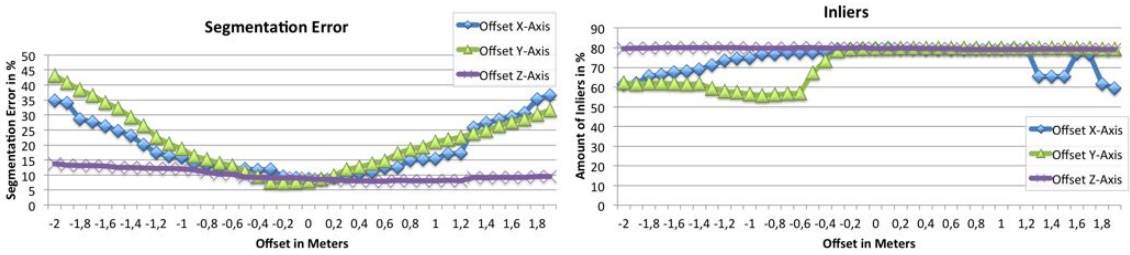


Figure 5.13: Accuracy measurements for different simulated GPS offsets. (Left) The segmentation error decreases with decreasing offsets. (Right) The amount of inliers is increasing with decreasing offsets.

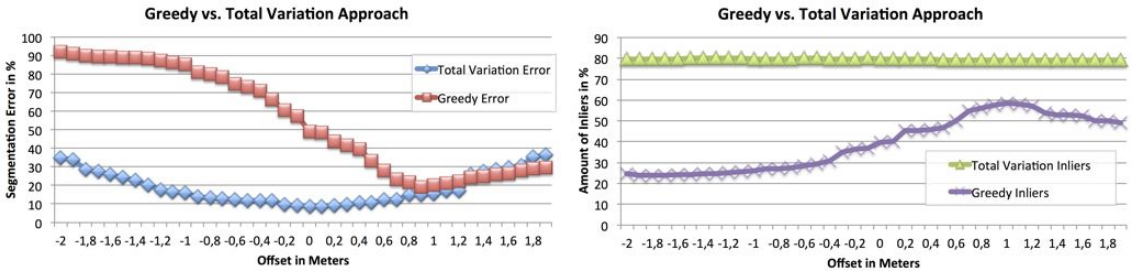


Figure 5.14: Accuracy measurements both segmentation methods. The segmentation error for the Total Variation method is lower than for the Greedy method. Bottom: The amount of inliers is higher for the Total Variation method.

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. Since this measurement can only provide information if the extracted object is big enough but not if the outline of the extracted object is similar to the reference object, we compute a second value, the *segmentation error*. The segmentation error is computed by calculating the difference in pixel between the extracted polygon and the polygon created by re-projecting the 3D points and computing the outline of the resulting 2D points.

Another interesting aspect is that these measurements 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. 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 we use in our outdoor AR setup [100].

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 measurments to analyze the difference between the greedy-based approach and the Total Variation approach. We used the same methods to determine the segmentation error as well as the amount of inliers as described in the previous section. As shown in Figure 5.14 the segmentation error for the Greedy method 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 accuracy of the Total Variation method is higher 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 to decide whether virtual content is occluded or visible, we can achieve much 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,

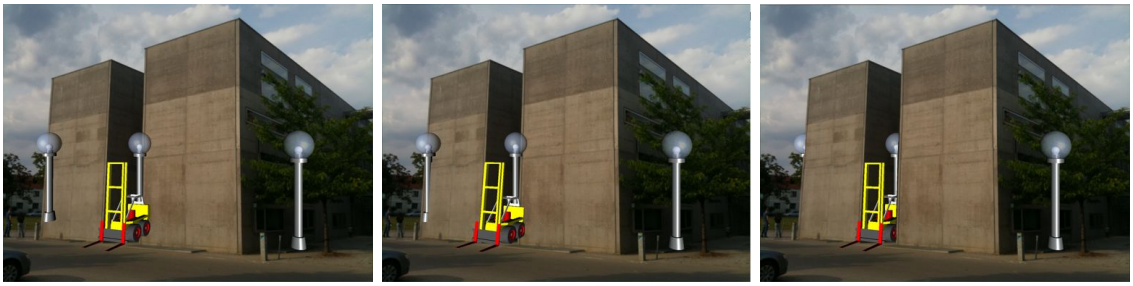


Figure 5.15: Occlusion management using the dense depth map. (Left) Without depth information no occlusion cues can be provided. (Middle) Using sparse geometries for occlusion culling can only create partially coorrect occlusion and may result in a incoherent visualization. Right) Using dense depth maps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

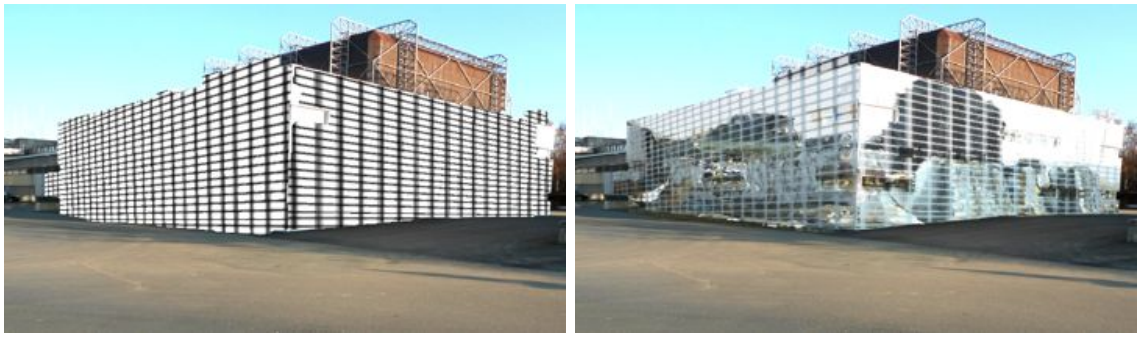


Figure 5.16: Occlusion management using importance maps. (Left) The importance map consisting of a grid is mapped to the dense representation of the physical wall. (Right) X-Ray visualization using the pop-up model combined with a checkered importance mask.

- Rendering occluding parts transparent,
- Using an importance mask to render the occluder partially opaque.

If the occluding object is only occluding 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 advanced occlusion management techniques, such as the importance maps presented by Mendez and Schmalstieg [79]. Their technique requires a phantom representation of the scene, which is usually not available. For this purpose we can use the dense depth maps of each object as the corresponding phantom object (Figure 5.16, (Right)) and map an importance map texture onto it.

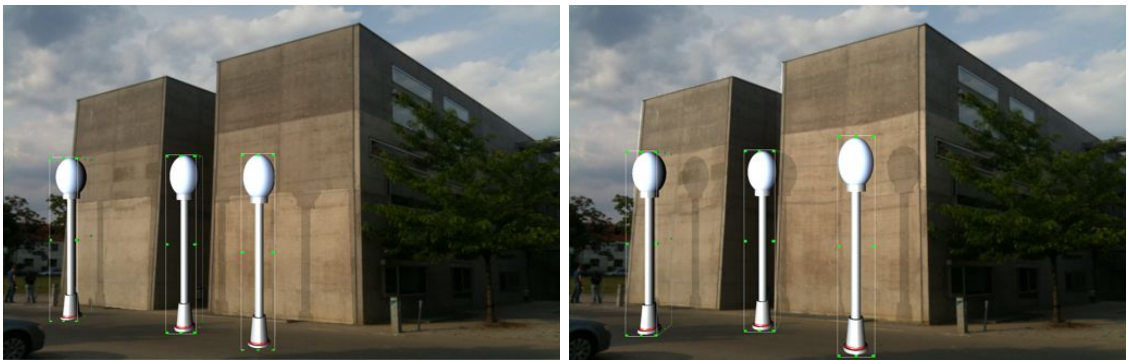


Figure 5.17: Using the dense depth maps for shadow rendering. Left) Sparse map used for shadow rendering can provide shadow only partly. Right) Dense depth maps allow creating convincing shadows over the complete geometry.

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. To create a shadow of a virtual object onto a physical object, an impostors of physical objects is required. Another requirement for correct shadow computation is that the physical light sources are known.

For shadow rendering, we can either use a scenegraph or shadow volumes. 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 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. Shadow volumes render shadows as follows:

Listing 5.1: Shadow rendering of based on shadow volumes.

```
Render video image.
Render impostors of physical objects into depth buffer.
Render shadow volumes of virtual objects into stencil buffer.x
Render these virtual shadows into the color buffer.
```

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.

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.

Annotations Annotations of the physical world can benefit from using our dense depth maps. Previous methods for annotating the physical environment work for instance in the 2D image space. Thus, only they only allow one to view the annotations from a specified location [70]. Other methods apply additional input devices such as laser range finders [116] 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 [117] 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, especially 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. Dense depth maps allow the user to survey real world objects in an Augmented Reality view. For instance, to survey the width and heights of windows, the user can select 3D points directly on the facade of the building by calculating the intersection point between the dense depth map of the facade (Figure 5.18, (Right)). In this way polygons can be created and the length of line segments can be determined.

5.6 User Survey

To gain some insights into the fidelity of our automatic depth estimation approach, 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 augmented reality scenes. We

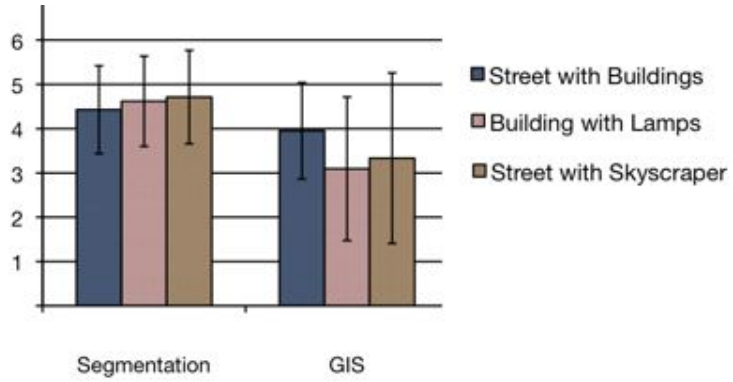


Figure 5.19: Results for different test scenes for the question "I understood the arrangement of virtual and real objects in the presented image".

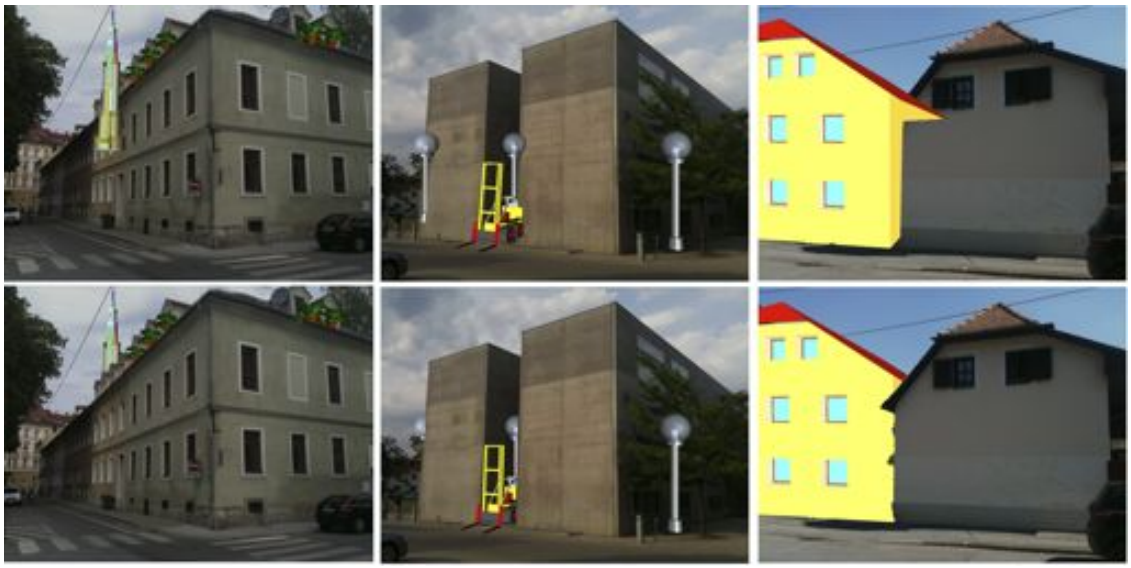


Figure 5.20: Test scenes for the survey.

used 5 scenes where we compared the results of our depth estimation approach for occlusion handling to the same scene without occlusion handling and where only transcoded GIS data was used for occlusion handling. As shown in Figure 5.20 people rated their comprehension of the scenes with the extracted depth estimation for the most scenes with an average value around 5 on a 6-point Likert scale (disagree - agree). The scenes with an occlusion management based on the sparse data from the GIS database were rated in the most cases slightly better than the examples without occlusion management, but still with a lower rating than the scenes based on the depth extraction. The standard deviation shows that people seemed to be more undecided for the scenes without occlusion management and with occlusion management based on the sparse data.

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 the 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. Whereas 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 [108] 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.

Finally, we are planning to investigate the different application scenarios of the dense depth maps with expert users from the construction and engineering industries to obtain information about the applicability in real-world scenarios.

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 the spatial relationships. Nevertheless, there is often the lack of absolute depth measurements in AR compositions. This is in particular the case for visualizations of objects that have no connection to the ground plane [20], such as X-Ray visualizations 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).

However, for several industrial outdoor applications, such as information query for digital assets or flight management of aerial vehicles absolute depth information is needed. 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 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 the object. 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 from chapter 4 provide a coherent integration of virtual subsurface objects into the physical world (Figure 6.1, Left). However, the depth estimation of these subsurface objects is still complicated, since the existing 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 that separate 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 show a connection between the virtual subsurface object and the surface. This is further supported by combining the magic lenses with a scale visualization. The user-centric magic lens provides the user with a view into the ground in front of her. 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 6.2: Physical vs user-centric virtual cues for MAV nagivation. Left) Occlusion cues for a planned waypoint of a MAV. A critical position is highlighted. Right) User-centric virtual cues visualize the distance between the user and the waypoint.

6.2.2 Virtual Cues for Floating Objects

Another application that benefits from user centric-virtual cues is AR flight management for MAV. 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 the flying MAV and simultaneously making use of a 2D map interface for flight path inspection. These problems comprise the mental effort required for mapping 2D map positions to the current physical environment and the estimation of the distance and height of the aerial vehicle. AR visualization can provide flight relevant information overlaid to the current environment and 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 a building (Figure 6.2, Left). But also for the flight management it can be advantageous to understand the exact depth between the MAV and the user. This information is not provided by the occlusion information.

In order to allow 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, we render in addition to a virtual representation of waypoint a rectangular geometry connecting the user's ground position with the waypoint. The rectangular object is thereby starting at the user's ground position, over to a point on that is created when projecting the waypoint on the ground plane and finishing in the waypoint itself. By texturing the additional geometry with a 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 the relationship between virtual data and the physical world. For instance, an virtal 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 or additional geometries 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 data. It is important to note that automatic creation of these 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, Middle). These excavations provide the user with depth cues about the position of the underground object and are supplemented with depth indicators, such as a soil texture with 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 parts of the excavation that would be naturally visible when looking inside an excavation opaque. Parts that would be not naturally visible, such as the side walls of the excavation

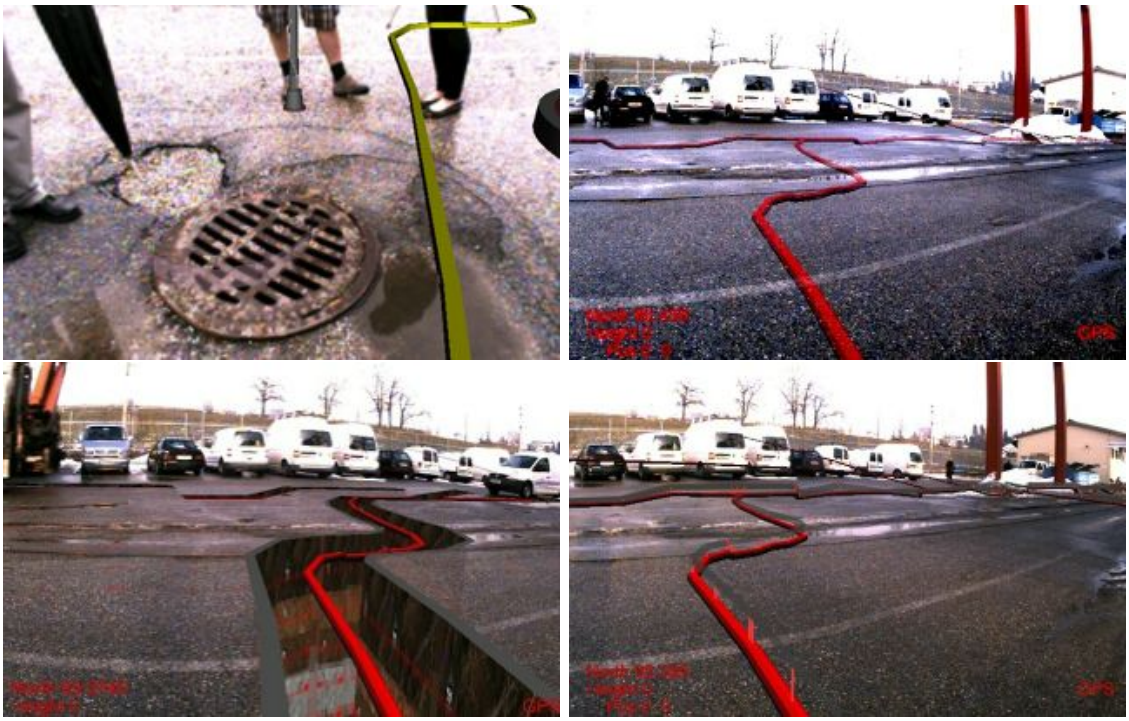


Figure 6.4: Different visualizations of an electricity line feature. Top Left) A yellow rectangular extrusion as graphical representation. Top Right) A red cylindrical extrusion. Bottom Left) Showing an excavation along the electricity line. Bottom Right) Showing virtual shadows cast on the ground plane.

from the outside, we render with an transparent overlay.

Reference shadows Reference shadows show 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 instance, for digging tasks they can help to find right positions to start the digging for a specific object of interest (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).

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 [20]. 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 at the surface.

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) Connection line combined with shadow.

Figure 6.6: Visualization techniques for virtual floating objects. Left) Visualization without virtual depth cues. Right) Visualization with depth cues for floating objects.

These lines allow 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 impression, 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 6.6, Right).

Figure 6.7: Virtual junctions. Left) Using a junction to support the surveying of a reference point. Right) Junction highlighting the reference point on the curbstone corner.

Junctions Abstract data has often 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 (geo-spatial bench marks). Such a visualization is for instance useful for accuracy tests or understand 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 ground plane and can support the visualization of this abstract data.

We used such junction visualizations for analyzing the accuracy of our tracking system based on survey marks. In Figure 6.7 (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 6.7, 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 Depth 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 opening, and a frame geometry that provides information where the magic lens is connected to the ground. The rendering is done as follows:

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 using the stencil buffer
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), an area outside the magic lens where the virtual pipes are displayed transparent. A 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

The user-centric 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 this connection we use again a rectangular graphical representation (Figure 6.2, Right). The waypoints are available as geo-referenced data in WGS84[1] format. Due to the single precision ability of OpenGL, we have to convert the geo-referenced 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 geo-referenced 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
```

6.4.3 Implementing Data-centric Depth Cues for Aerial Vehicle Navigation

In addition to the user-centric depth cues, we discussed depth cues that support the spatial relationship between the information that is relevant for aerial vehicle navigation,

[1] World Geodetic System 1984

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 have 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 has been 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 [118], 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 geo-referenced 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 data-centric depth cues data from GIS databases is more challenging than for single data points as described in the previous part of this section. Data-centric cues require that various visual representation of the same GIS data can be created and managed. However, 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 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.

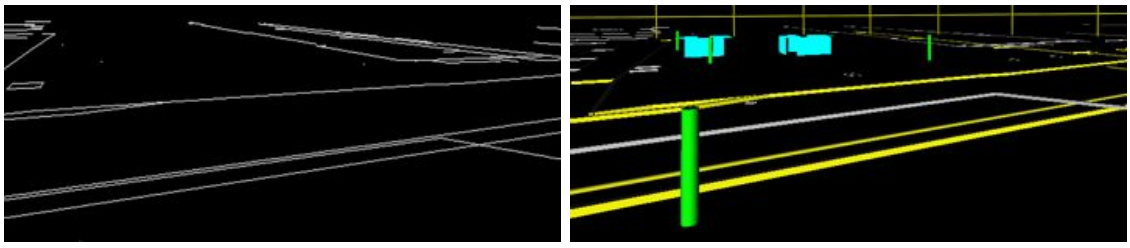


Figure 6.8: GIS data model vs. transcoded geometries. Left) GIS data model, all data is represented by lines and points, which makes interpretations difficult. Right) Transcoded geometries, GIS data is transcoded to a comprehensible representation showing cylindrical objects representing trees and cylindrical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

So far both levels exists separately and can not interchange data. We add a new data layer that serves as transmission layer between both 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 surveyed line of points (compare Figure 6.8, (a) and (b)). Especially 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 updating the features. To support these operations we developed a bi-directional transcoding pipeline that realizes the conversion from GIS features to com-

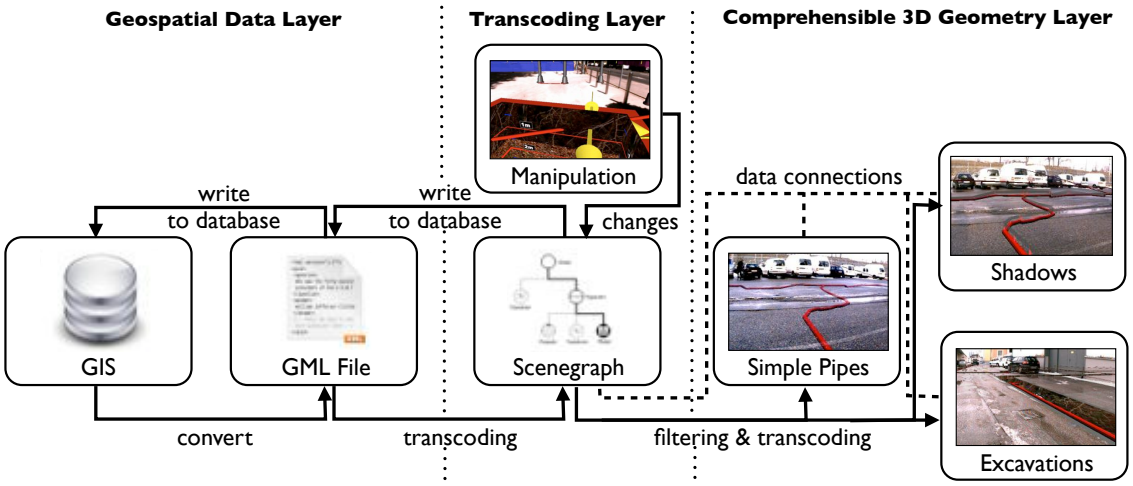


Figure 6.9: Overview of the bi-directional transcoding pipeline. Data from the geospatial database is converted to a simple GML exchange format. The GML file is imported to the application and transcoded into the transcoding layer representation. Filtering and transcoding operations map the transcoding 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.

prehensible 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. 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 before, we use FME for extracting the features from the geo-database. The user interactively selects objects of interest in the back-end GIS, which is then exported to a GML-based file format containing a set of features. Each feature thereby describes either a as-planned or an as-built object in the physical world. The feature's geometries are only stored as 2.5 data. The AR visualization requires a 3D representation. Thus, the first step is to convert the 2.5D information into a 3D object by using a Digital Terrain Model (DTM) and known laying depths of the subsurface objects. This step has to be done offline before starting the AR system since it requires external software and interactive selection of the export area. The following steps of the transcoding pipeline can be done during runtime but are not in realtime.

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

From Transcoding Layer Data to Comprehensible Geometries So far we converted the 2.5D data into 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 should 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 extruded circle (Figure 6.4, b) or as an extruded rectangle to show an excavation around the pipe (Figure 6.4, Bottom Left and 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 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 very flexible system that can support many applications.

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 to not only configure 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 circular extrusion representing a 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.

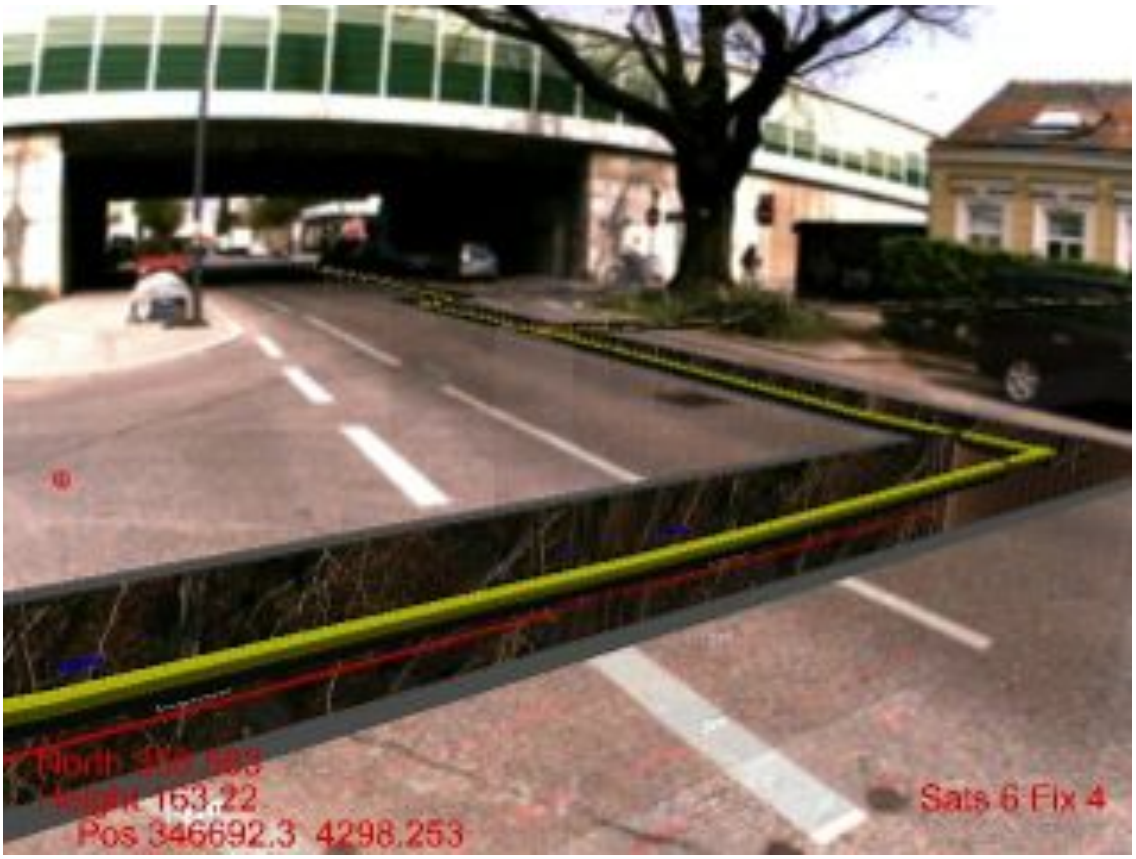


Figure 6.10: Excavation along a yellow pipe. Excavation and pipe are only rendered opaque at pixels that are covered by the top of the excavation. Outside this mask virtual content is rendered transparent.

```
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.

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 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 and the 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 ovleray 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 underground objects and a DTM. The intersection points are used to create a semi-transparent flat shape that is located on the ground level.

Listing 6.5: Rendering of reference shadows.

```
Render video texture
Render shadows
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. 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 reference shadows.

```
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. 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 where the filtering-transcoding concept defined a junction geometry.

Listing 6.7: Rendering of reference shadows.

```
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 to test the interactive data roundtrip and analyze the surveying accuracy. Additionally, we gained first feedback with the setup in workshops with expert users from civil engineering companies.

Field Tests We used data from conventional GIS provided by two civil engineering companies to test the system with real-world data in field tests. Based on this data, we performed first field-trials using the surveying system with 16 expert participants (12m/4f) from a civil engineering company. Users were asked to survey an existing pipe with the system and afterwards to complete a short questionnaire. Results of the test showed that users rated the suitability of the AR system for "As-built" surveying over average on a 7-point Likert scale (avg. 5.13, stdev 1.14) and compared to traditional surveying techniques as quite equivalent (avg. 4.43, stdev 1.03). The simplicity of surveying new objects was rated above average (avg. 5.44, stdev 0.96). And while the outdoor suitability of the current setup was rated low due to the prototypical character (avg. 3.28, stdev 1.20), the general usefulness of the AR application was rated high (avg. 5.94, stdev 1.19). We also asked the participants about selected virtual pictorial cues and the ratings were above average. The reference shadows in combination with connection lines were rated with 5.63 (avg. 5.63, stdev 0.80) and object-aligined cutaways were rated high (avg. 6.19, stdev 0.66).

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. Whereas, user-centric cues can 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 functionality 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

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 the already complex physical environment with complex information is likely to be subject to information clutter.
2. Complex data may occlude information about the physical environment.
3. Complex virtual information may be a 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 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

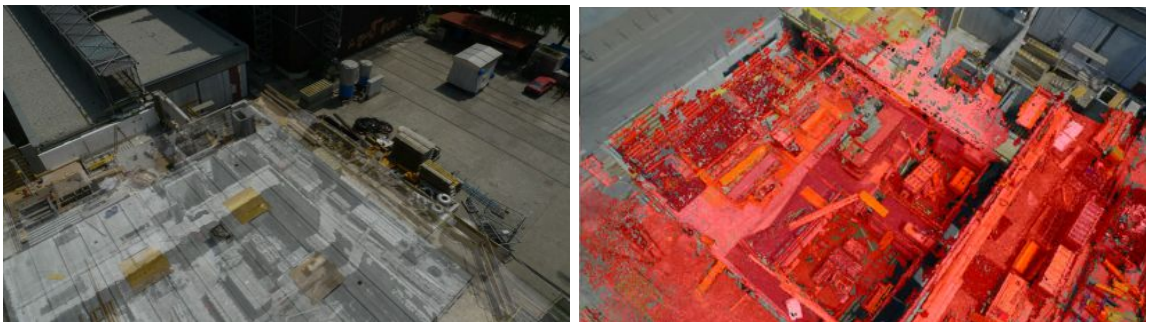


Figure 7.1: Simple blending of complex data. The current state of the construction site is overlaid with 3D data of one previous point in time. Left) Transparent overlay. Right) Overlaying a colored point cloud for highlighting the difference between virtual data and physical environment.

support the comprehension of complex data. These methods allow to provide 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 visualizing 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, the progress information can be captured by using aerial vision. By flying over a construction site with a MAV, the construction site can be easily reconstructed on a regular basis. Thus, different states of the progress can be represented by a 3D point cloud or a 3D mesh.

Overlaying this kind of information 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. Additionally, a lot of the physical world context is occluded by virtual data. Thus it is complicated to understand the current state actual environment as well. Accordingly, the user does neither understand the previous state nor the current state, which means there is no benefit of using such a visualization.

Another challenge that was not considered so far arises when visualizing more than only one point in time. Instead of having one data point for each point in space, there are now multiple data entries representing different points in time for each 3D point in space. This time-component of the visualization adds to the complexity of the visualized information (Figure 7.2).

The main research question of this chapter is how to visualize this kind of complex data in an AR overlay effectively. We will start in the next section by investigating traditional methods for the visualization of complex data representing one point in time

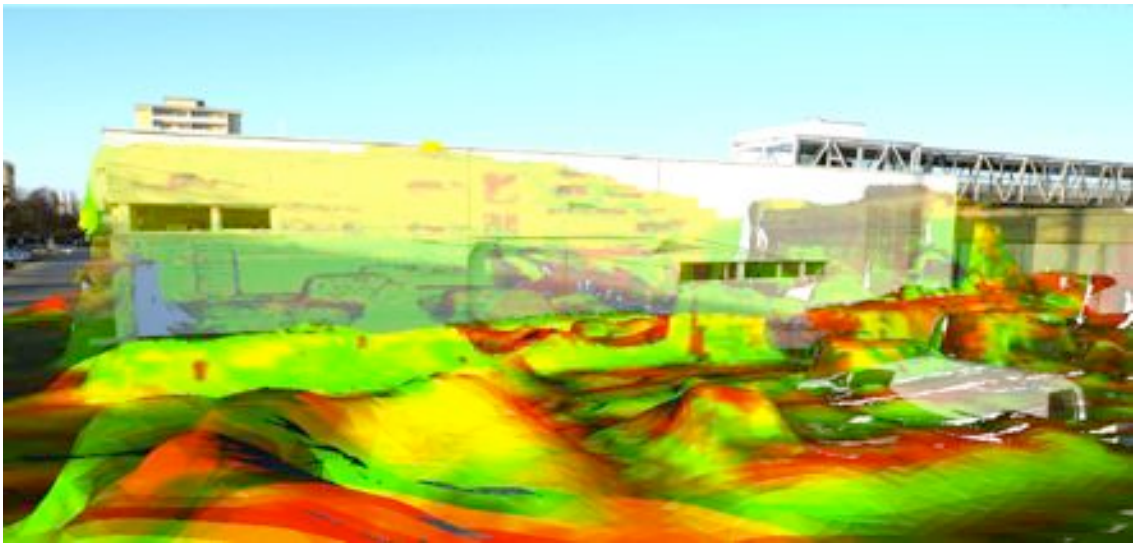


Figure 7.2: Visualizing two different points in time in one view with different shadings leads to image clutter as well.

or a small number of points in time. Thereby, a spatial information filtering controlled by focus&context techniques allows to compare a limited number of previous points in time to the current state. Afterwards, we will continue with methods for information abstraction that aim to reduce complexity when visualizing more than two data sets representing different points in time.

7.1 Traditional Information Filtering Tools for Visualizing Complex Data

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 where a previous point in time is displayed. The information filtering then only displays this kind of 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.

7.1.1 2D Focus&Context Tools

2D focus &context tools help to address the problems of information clutter and occluded physical world information by allowing the user to define a focus region in image space. Within this 2D focus region the virtual information presenting a previous status of the construction site is displayed. The region that does not contain the focus area is called context region. In the context region the current status of the construction site is displayed

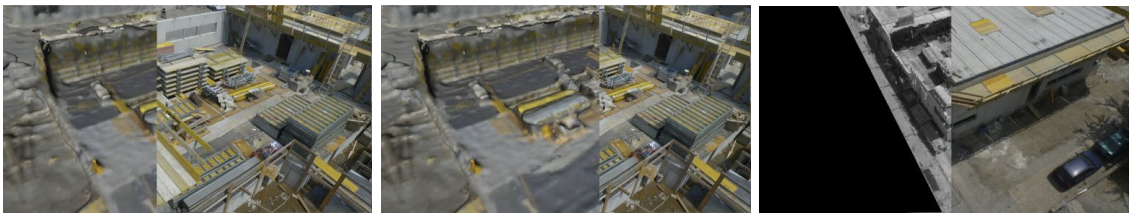


Figure 7.3: Side-by-side visualization using 2D sliders. The mouse input defines the border between video image and virtual content. Left and Middle) The virtual content is overlaid on the right side of the border. Right) On the right side of the border only the virtual data representing a previous point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

using the current camera image of the physical environment. The interactive tools that we will discuss here comprise 2D sliders, 2D magic lenses and selection of 2D regions.

2D Slider Tools that are often used to provide an interactive side-by-side visualization for before and after comparison of urban or historical scenes are 2D sliders. For instance, 2D sliders combined with aerial photography were used to visualize the amount of damage during the 2011 Tohoku earthquake and tsunami in Japan[1]. Another application example is the documentation of change in urban scenarios, where buildings are displayed in their current state and in the past[2]. To achieve such a visualization using photographs it is important that both images 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 old photo. For this purpose they use 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 and pose estimation [7]. 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 enviroment, the technique moves from computational rephotograph to AR (Figure 7.3). Similar to the methods for photography the the user can interactively move a slider in image space to control the border between video image and the virtual overlay containing the 3D information. In our application the 3D information contains a previous state of the construction site, but the technique could also be used for different urban scenarios as long as the 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 then the border's x-coordinate. For other fragments the virtual data is either overlaid to the video image (Figure 7.3, Right) 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.

[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`

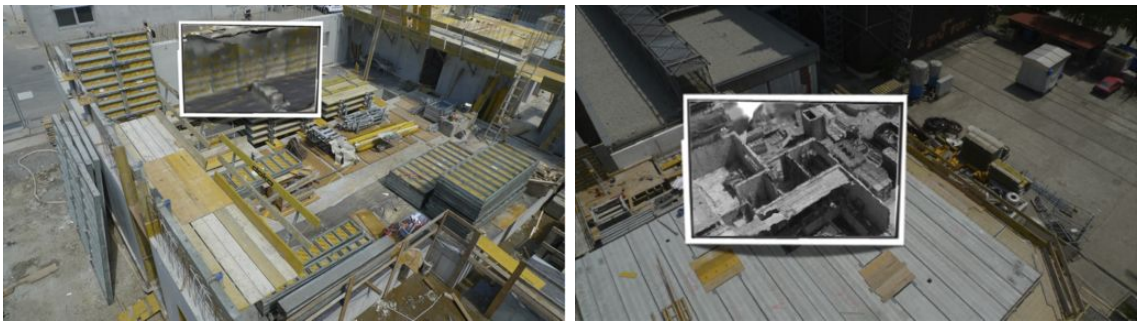


Figure 7.4: 2D Magic Lens. Inside the focus region (highlighted in white) the virtual content is displayed. Left) Overlay of virtual data inside the context region. Right) Grey shading inside the focus area supports the differentiation between virtual information and physical world.

This technique is similar to magic lens interfaces from Information Visualization [11]. In contrast to the 2D slider, a 2D magic lens allows the user to move a dedicated focus area to the regions of interests. The 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 test 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 where the past view is shown, is the selection of image region. Some artists such as Larenkow used such a technique to define interesting image regions 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 important buildings in Berlin with views from the 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 provide the user with an interactive tool that allows one to select an image region by clicking on them 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 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 this 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

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

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

focus area and used to visualize the virtual content (Figure 7.5 Right).

Importance driven filtering Another image-based technique preserves the natural important image structures by putting the visibility in dependency to the image structure's importance. The user can change the amount of image structures that are visualized by modifying the threshold of importance. With this approach the user can exclude less important image structures depending on his preferences. To achieve this, 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 main disadvantage that they can not address the problem of 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.

7.1.2 3D Focus&Context tools

To address the problems of 2D focus&context tools, 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 user may wants to inspect one specific element that is occluded by other previous structures. By defining a focus area in 3D, it is possible to only visualize information for this specific region.

Another interesting aspect of the 3D tools is their ability to convey information about depth. This supports the spatial relationship of current objects and older objects in the scene.

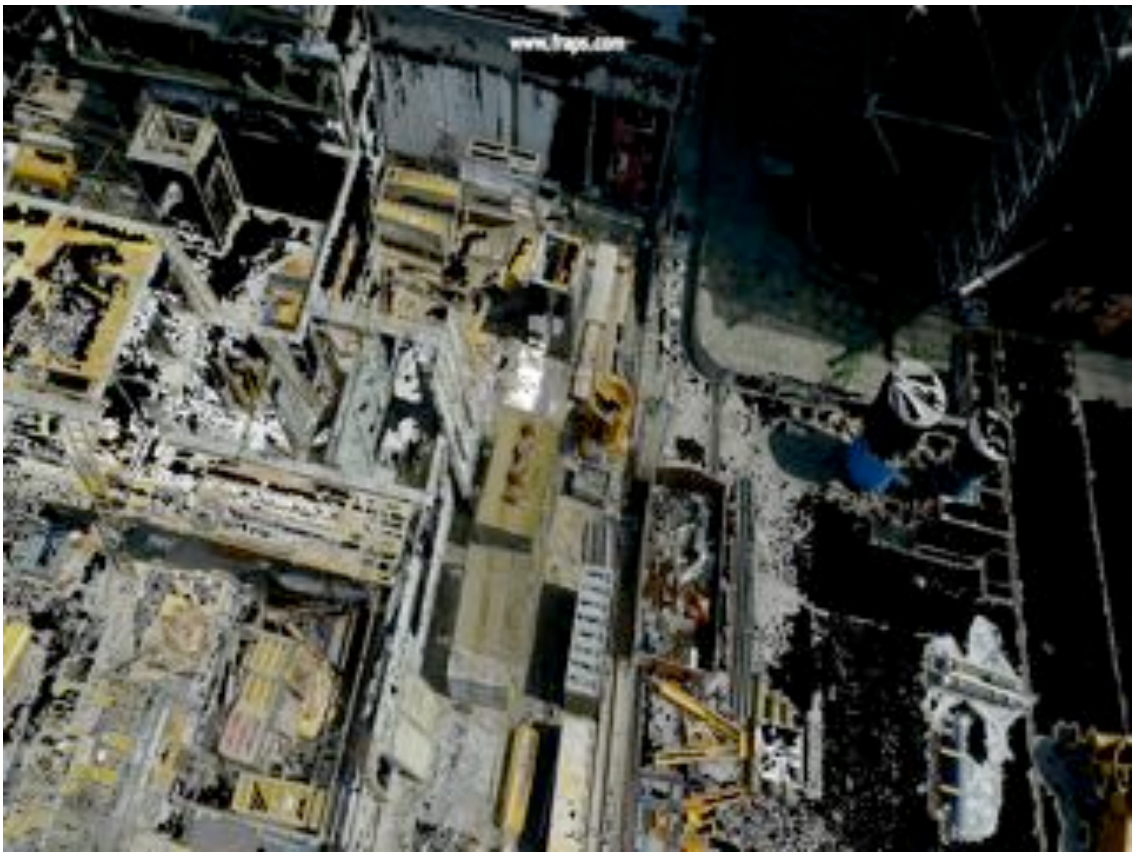


Figure 7.6: Importance driven Ghosting.

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 is that this separation is done in 3D. The 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. The alignment and the dimension of the slider wall can be defined by the user by using a 3D manipulation tool. The intersection plane between virtual geometry and sliding wall provides information about the depth and height of the virtual content.

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 it self to the planes in the scene. This could be done by detecting planes in the 3D point cloud.

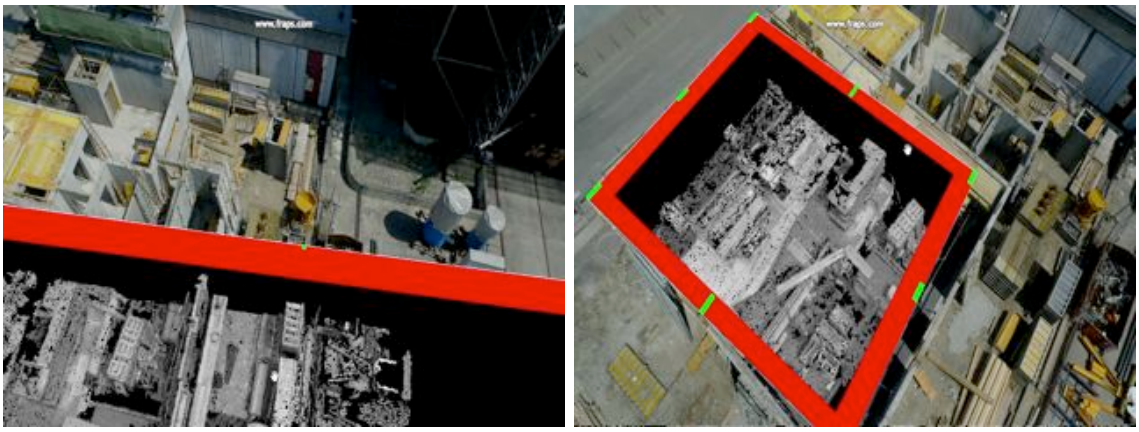


Figure 7.7: 3D tools for information filtering. Left) A 3D slider allows the user to divide the view in an area where a previous status of the construction site is shown and a area where the current status is shown. Right) The 3D magic lens magic lens defines a three dimensional region where the virtual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

3D Color Coded Magic Lens So far we only discussed visualization tools supporting the overlay of one data set, such as a 3D point cloud representing one previous point in time. Since the construction site monitoring and documentation requires the visualization of multiple points in time in one view to inspect progress, we further investigated methods that allow to display multiple complex datasets. For this purpose we visualized different points in time using different visualization techniques to make a differentiation between them easy. A simple approach to achieve such a color coded rendering is to render the different reconstructed data with different color buffers enabled (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. Combining a color coding with multiple magic lenses allows the comparison of multiple points in time.

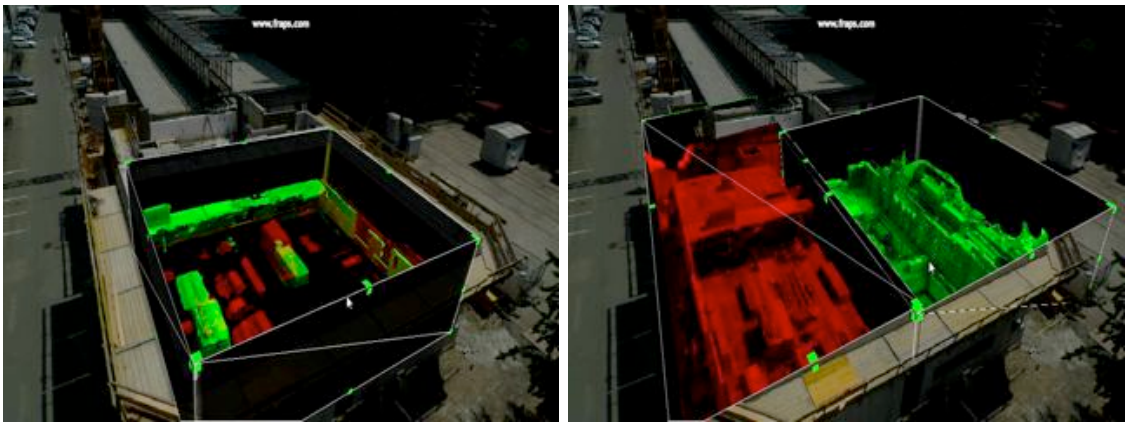


Figure 7.8: 3D focus&context tools using different color codings. Thereby green represents an earlier stage of the construciton site as red. Left) Two color codings are used in the same 3D magic lens. Right) Two 3D magic lenses display different points in time.

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. Visualization of two consecutive points in time in separated views. Clutter and complex changes make it hard 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.

7.1.3 Limits 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 the 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 often not be perceived [105]. However, 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 reference to the visualization of time-oriented data, we face the limitation that with a higher number of points in time information clutter will occur.

7.2 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 [85, 102] or visualizing a single previous snap-shot of a building [60]. 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. Thus in AR it is still complicated to follow changes over a time period. In order to analyze the change over time, one has to study the object of interest at different points in time. It is necessary to compare different datasets representing different points in time among themselves and with the current real world situation.

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 and detail techniques. The presentation of different layers of abstraction and detail allows one to thor-

oughly analyze dynamical scenes. One the one hand, overview techniques allow 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. By registering the 4D visualization with the real world, the user can inspect changes in relation to the real world context. The technique we propose in this section follows the *Visual Information Seeking Mantra* of Shneiderman [103]. 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 this section we will show how the mantra can be applied for visualizing multiple complex datasets in AR. In particular, we apply this concept for 4D datasets.

In contrast to the work of the previous section and earlier work (such as [37]), the goal of this section is to visualize a changing scene over time in an AR view. The main requirements for the visualization are (1) to preserve context of the changing environment and (2) track changes over time.

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 data. Existing techniques range from very simple line plots [45] and bar charts [45] up to visualization suites [72] which allow a combination of visualization techniques in an intelligent way (see [2] for an extensive review on visualization techniques of time-oriented data). However, while existing visualization techniques for time-oriented data commonly have been designed to study the data in front of a PC, we aim at an on-site data visualization with AR. In addition to traditional visualization techniques for time dependent data, we provide the user with contextual information about the real world relation of the data he studies.

We focus on the visualization of 4D data coming from multiple 3D reconstructions. Typically, this data would be further explored in a Virtual Reality (VR) view or registered to individual images. However, changes of complex scenes are difficult to understand and furthermore, these visualizations are missing the context of the real world. The main idea of our approach is a multi-level overview and detail approach to manage the visual complexity and to tailor the visualizations to an augmented reality view to preserve the context of the real world. Applying these methods to construction site monitoring tasks has the power to improve a construction site manager's ability to investigate and understand progress.

In order to visualize changes in a comprehensible way, perceptional issues like change blindness have to be addressed [104]. Change blindness describes the problem to notice changes. This problem often appears when a lot of changes occurs in the view of a person. To avoid this, it is important that the user can keep track of selected changes for instance by providing visual additional hints [82]. This is particularly true for outdoor AR visualizations, since the environment is changing all the time and important changes in the data may be concealed. The traditional information filtering tools did not addressed change blindness (section 7.1). 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-separated visualizations such as sliders.

In this section, we present a visualization concept that displays multiple points in time

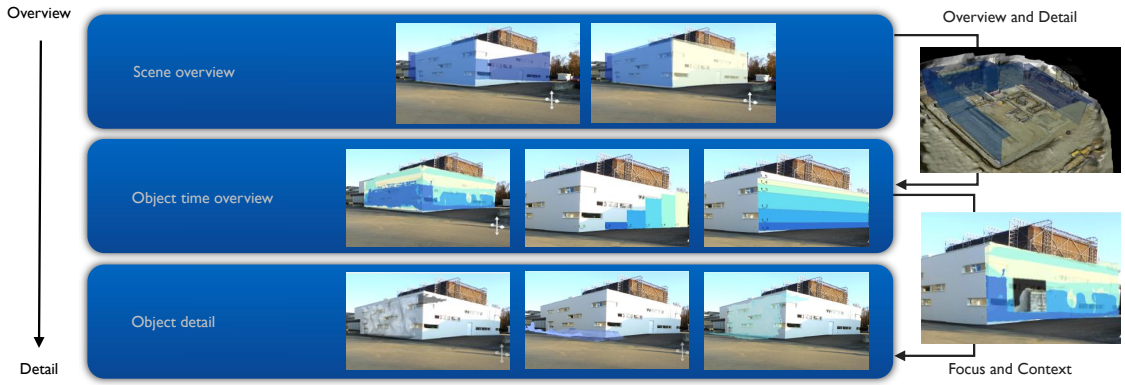


Figure 7.10: 4D visualization concept. Multiple level of detail allow to explore time-oriented data in an Augmented Reality visualization. The highest level of abstraction provides just enough information to provide an overview of the data in relation to the objects in the AR environment. The second 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.

in one view, while still providing a detailed view on the data itself (Section 7.2). Furthermore we show how to implement this visualization concept (Section 7.5) and apply the 4D overview and detail AR visualization for construction site monitoring and documentation (Section 2.4).

The general idea of our interactive 4D visualization follows the Information Seeking Mantra. For this purpose, we provide three visualization levels varying in their level 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 and display the virtual content on different scales. They show information on a scene overview scale, on an object time overview scale and on an object detail 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 to move between the different visualization level and to relate them to each other. We first discuss each level of abstraction in detail before we describe how we move from one to the other.

7.3 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 filtering of information, the second level L1 presents information about the variation in time of an attribute, in relation to the object's geometry. For example, the evolution of an object's degree of completion can be analyzed in more detail through color coding parts of the 3D object, contour diagrams or simple charts which are registered in 3D space. The third visualization level L2 uses the highest degree of detail by presenting a complete 3D rendering of the objects at a different points in time. By augmenting the scene with geometry from previous points in time, the user is able to travel through time and inspect 3D structure in detail, even in relation to the real world environment of his current point in time. His travel depends on his prior analysis of a higher level of abstraction, which allows for more effective comparison of multiple points in time in single view than detailed renderings of complex 3D structures.

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. Abstract visualization that show summary information without any detail can support user in finding these objects.

Golparvar-Fard et al. [38] used color coding 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 show 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 where multiple objects are visible.

We use such a per-object-abstraction for overview visualization on the *scene overview level*. However, combined with filtering techniques, we use this visualization layer as a starting point for further inspection on a per object-basis. As shown in Figure 7.10 (Top) real scene objects are colored with colors representing their current status. Additionally, measurements can be displayed on the object. A limitation of this visualization level is that it can only represent one single point in time or a single value to summarize the data in one view.

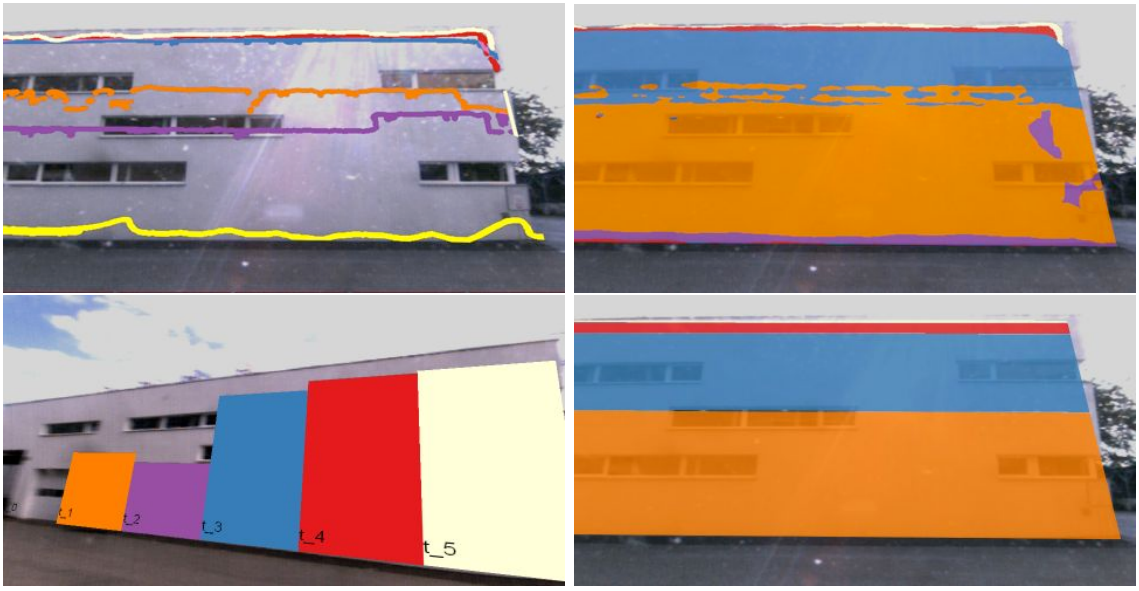


Figure 7.11: Object time overview visualization: providing information of multiple points in time for a single object. The multiple points in time are thereby color coded. Left: Heightlines showing the different height of the wall at different points in time. Middle Left: Geometric Completion shows the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

L1: Object time overview level The visualization of L0 provides a quick overview over all areas of interest in the scene, but restricts inspecting a selected area 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 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 the different heights of an area at different times (compare Figure 7.11, Left).
- Geometric completion: abstract information with geometric reference. For instance a diagram showing average completion, color coded completion (compare Figure 7.11, Middle Left and Middle Right).
- Block diagram for average completion: block diagram showing the average completion in relation the geometry of the object of interest (compare Figure 7.11, Right).

L2: Detail Level The last visualization level allows to inspect the detailed geometry and appearance of an earlier version by rendering the representation of the earlier version itself. In the overlay of the 4D data, the user can inspect the progress on the object in a detailed way. Furthermore, color-coded shading of 3D structure can be applied to support visual discrimination objects at different points in time (Figure 7.12).

Figure 7.12: Color-coded shadings of a wall geometry at different points in time.

7.4 Transitions between Visualization Levels

After introducing the visualization levels, the second challenge is the interactive transition between them. In this section we discuss a concept for interaction methods that allow the user to navigate between the visualization levels. Additionally, details on the implementation of interaction techniques are then given in Section 7.5.3. A user starts in the first level and selects objects of interest for further inspection in the other levels. For example, by filtering the objects based on their progress in L0, only objects with a certain amount of progress or completion are shown. Afterwards the user can select or physically move to an object of a certain amount progress or completion. We connect all levels by interactive view manipulation techniques providing visual continuity when drilling down for more details.

Managing contextual and focused information has a long tradition in information visualization. In our approach, we use in each level the real world structure as well as abstract representations as contextual information next to detailed information, which are currently in the focus of an analysis. This allows us to apply similar visualizations for moving between the three abstraction levels in the visualization space. As described by Cockburn et al. [17], overview and detail techniques may separate focus and context information temporally or in space by applying, for instance, zooming techniques or by presenting the data in separated views. Focus&context techniques integrate these information in one single view and present context and focus information together.

7.4.1 Overview and 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

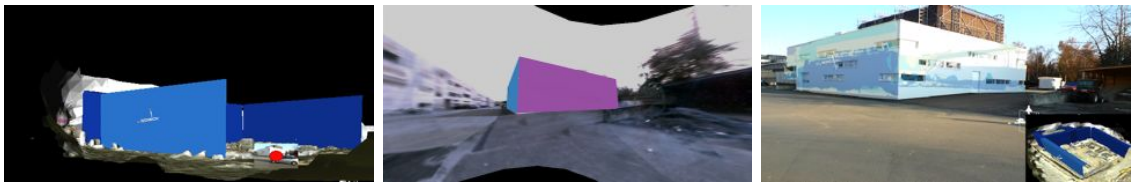


Figure 7.13: Overview and detail techniques: Left: Zooming to an overview with abstract visualization for multiple objects. Middle: Using an extended field of view for providing an scene overview. Right: Birds eye view in WIM with abstract visualization for multiple objects.

of the visualization levels makes the transition between the first and the second level a perfect candidate for overview and 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 allows us to comprehensively transition between the first and the second level of abstraction.

Zooming The zooming interface allows the user to zoom out of his current view. Based on the amount of zooming this provides an overview of the scene where the user can inspect multiple objects at once. With larger distance to the scene, the interest on detailed information gets less and more abstract representations of the scene objects are presented (Figure 7.13, Left).

World in Miniature For the WIM 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).

7.4.2 Focus and Context

Since a full scale of both L1 and L2 visualizations is critical for their interpretation, spatial overview and detail techniques are not suitable to switch between them. Also, temporal overview and detail techniques are not suitable, since they demand a rather high amount of workload [104]. Therefore, we use focus& techniques to combine the visualizations in a single view and within correct scale.

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

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 [17]. We adapt this concept in such a way that the area inside the magic lens shows virtual representation

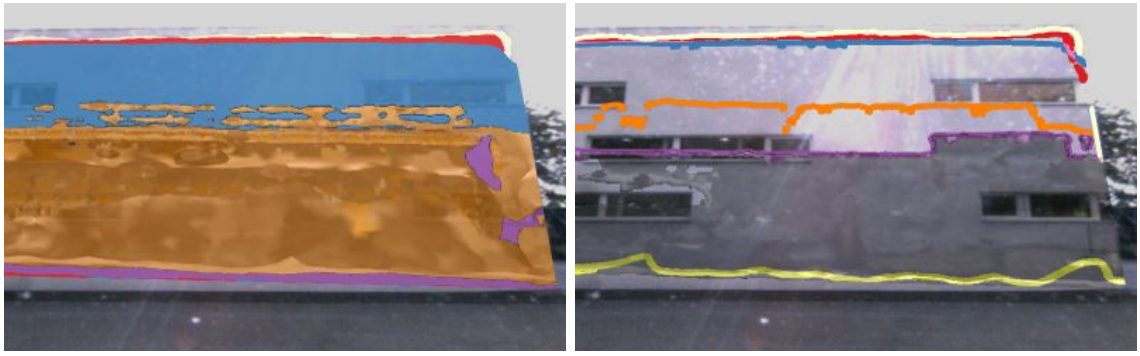


Figure 7.14: Transitions between Visualization Levels with Overlay: Abstract representations such as (Left) geometric completion and (Right) height-lines are combined with the object's detailed view to simplify the transition between both.

from a former point in time while the area outside shows the abstract progress information as well as the real world environment (Figure 7.15).

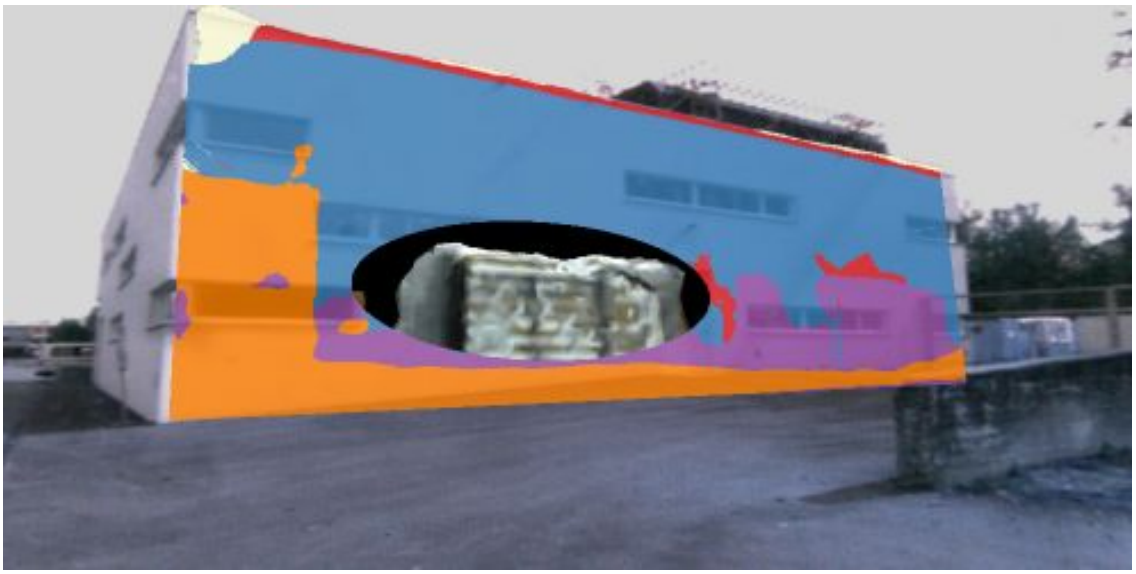


Figure 7.15: Transitions between Visualization Levels with a Magic Lens. The magic lens area provides a detailed view on a scaffold used to built the wall.

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 [107]. 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

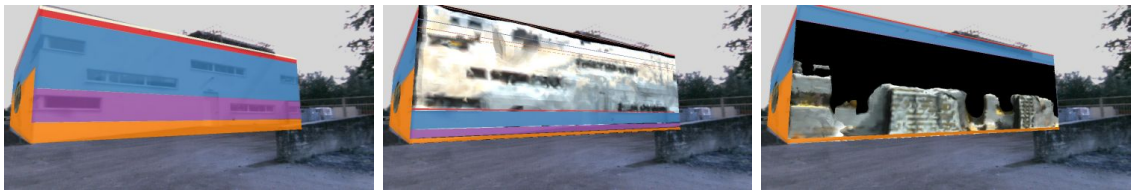


Figure 7.16: Transitions between Visualization Levels with Distorted View. Left: Average Completion Diagram. Middle and Right: By selecting a point in time within the completion diagram the user gets a detailed visualization of the wall at the point point in time he selected. To provide still the abstract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

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.

7.5 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.

7.5.1 Extracting Time-oriented Data

Visualizing the reconstructed meshes in an AR view by using simple blending techniques is cumbersome and may lead to an incomprehensible visualization (as demonstrated in Figure 7.9). To avoid this, we abstract time-oriented data for each object in the scene. Therefore, 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).

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. The result of the rendering is captured in an off-screenbuffer and saved with the region of interest for each point in time and can be rendered to the visible face of the region of interest.

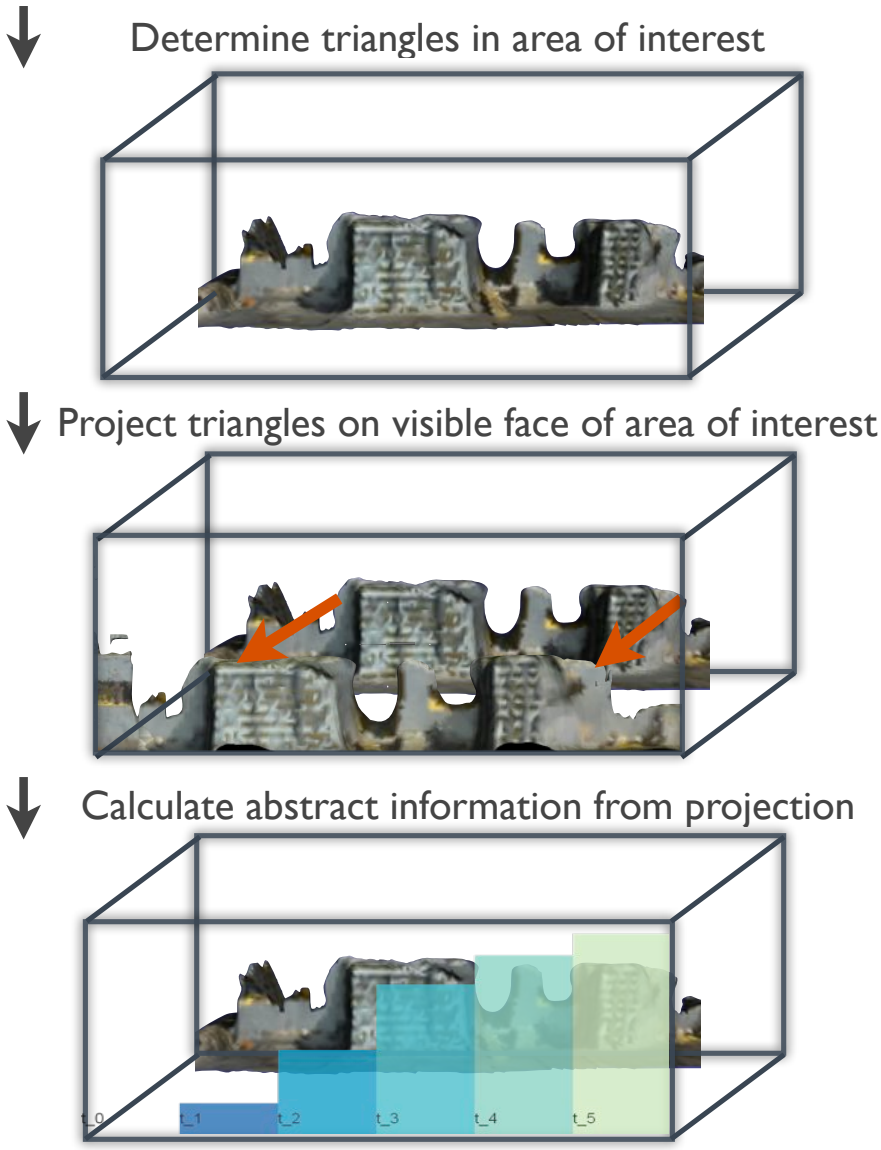


Figure 7.17: Extract time oriented data for areas of interest. First, triangles are tested, if they are in the region of interest. Second, projecting the determined triangles onto visible face of region of interest. Third calculate abstract representations from the projection.

7.5.2 Implementation of Visualization Levels

The extracted time-oriented data for the dynamic scene can now be visualized using different representations. These representations vary in the level of abstraction that are used for different purposes as described in Section 7.2.

Scene Overview The scene overview visualization level L0 is used to present overview data of multiple objects at multiple points in time. To avoid clutter due to the presented information 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. Measurements that can be computed directly are completion of the object or progress. Completion describes the area of the region of interest that is already filled and progress describes rate of completion over time.

To compute the degree of completion, the time-snapshots from Section 7.5.1 are binarized to compute the occupied pixel of the surface. The completion is then computed as the area of occupied pixels *Occupied* divided by the projection surface of the region of interest $InterestArea$:

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

Then, the progress can then be computed as:

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

These measurements are presented by color coding an object or by textual annotations. Other summary values such as criticality, cost or time deviation provided by external sources can be used as well.

Object Time Overview As described in Section 7.5.1 the main purpose of this visualization level is to present multiple points in time for one object of interest. Since the visualization of detailed reconstruction data of various points in time would lead to clutter, we use an abstraction to present multiple points in time in one view. By presenting this information in relation to the object's geometry as well as to the real world, the geometric context of the time evolution is visible as well.

By using the projections of the changing geometry, we can compute time-oriented representations such as height lines, geometric completion and block diagrams showing the average height.

The height lines are computed by using the binarized time-shot image and 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.

To compute the geometric completion diagram G, we determine the difference of *Occupied* between point in time t and color code this area with a color coded referencing to the point in time:

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

The geometric average completion diagram for multiple points in time is computed by using Equation 7.1 and coloring all pixels below the average completion height with the corresponding color value. The block diagram is computed in a similar way, average completions 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 real scene with all the computed object time overview information as shown in Figure 7.11.

Object Detail The last visualization level allows the user to retrieve detailed information about the objects 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 of the region of interest for a selected point in time. The visualization of this data can be as triangles as shown in Figure 7.17 (Top) or as screenshot of the reconstructed data as shown in Figure 7.17 (Middle). To support the differentiation of the data of different time steps each time step can be visualized with different color codings by using different shadings.

7.5.3 Interactive Transition between Visualization Levels

As described in Section 7.4 our overview and detail visualization approach allows to interactively navigate between the different visualization levels. More precisely, the user 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 and a 2D GUI. An overview of the interaction methods are described in Figure 7.18.

Transition from L0 Scene Overview to L1 Object Time Overview By using overview and detail techniques such as zooming or WIM, the user can get an overview of the complete scene and search for interesting or conspicuous objects. For this purpose the user selects either the zooming or WIM mode in the 2D GUI. For controlling the transition itself, the user manipulates a slider in the GUI (Figure 7.18). In the WIM mode the scene overview level L0 is displayed in the same view as the AR visualization. To display the WIM in the same view as the actual AR scene, we use an offscreen rendering of a virtual birds eye view of the scene (Figure 7.13). For the zooming mode, we provide two different methods that allow the user to get 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 and (2) an AR zooming mode where the slider is used to control the angle of view of the camera rendering virtual and real scene. By using the panoramic images for environment representation, this allows the user to increase his field of view of the virtual scene while preserving the context to real 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

Interactive Transition Modes		Interaction Methods						
		Change Camera View	Filtering	Select Attributes	Select Time-Visualization Method	Select Objects	Select Points on Object	Select Points in Time
Overview & Detail	WIM	2D GUI Slider	2D GUI Spinbox	2D GUI, Combobox	-	Mouse-Input on Scene	-	2D GUI Slider
	Zooming	2D GUI Slider	2D GUI Spinbox	2D GUI, Combobox	-	Mouse-Input on Scene	-	2D GUI Slider
Focus &Context	Overlay	-	-	-	2D GUI, Combobox	Mouse-Input on Scene	Mouse-Input on Object	Mouse-Input on Object Overview, 2D GUI Slider
	Magic Lens	-	-	-	2D GUI, Combobox	Mouse-Input on Scene	Mouse-Input on Object	Mouse-Input on Object Overview, 2D GUI Slider
	Distorted View	-	-	-	2D GUI, Combobox	Mouse-Input on Scene	Mouse-Input on Object	Mouse-Input on Object Overview, 2D GUI Slider

Figure 7.18: Overview of interactive transition modes with their corresponding interaction methods.

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 The transition between object overview level L1 and the object detail level L2 allows the user to relate abstract overview information visualizing multiple points in time to concrete appearance and geometry of the region of interest. Presenting both levels in one view by using F+C techniques as described in Section 7.5.2 supports the mental connection between them. Focus and context can be used to combine abstract time-oriented values such as completion, progress or cost with the concrete appearance of regions at certain points in time. For instance the transition between the completion color coding is implemented as magic lenses, view distortions or overlays.

The user can interactively select points of interest from the abstract representation of multiple points in time and subsequently the detail information of the select time is shown as focus and context visualization. To provide a selection mechanism for the points on the surface of the region of interest, the mouse coordinates have to be converted from the camera image space into the image space of the surface of the region of interest. For this purpose, the mouse coordinates are converted to world 3D points and by using the object of interest's transformation matrix the mouse coordinates are mapped into the local coordinate system of the object. 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 and combines the abstract information A with the 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 7.19, 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 the detail information. For this purpose abstract information for points larger than the selected point in time is moved to the beginning of the diagram (Figure 7.19, Bottom). For instance applied for the color coded completion, the user can select a part of the color coding 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 geometric correct location (Figure 7.19 and Figure 7.16).

7.6 Application: Construction Site Documentation and 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

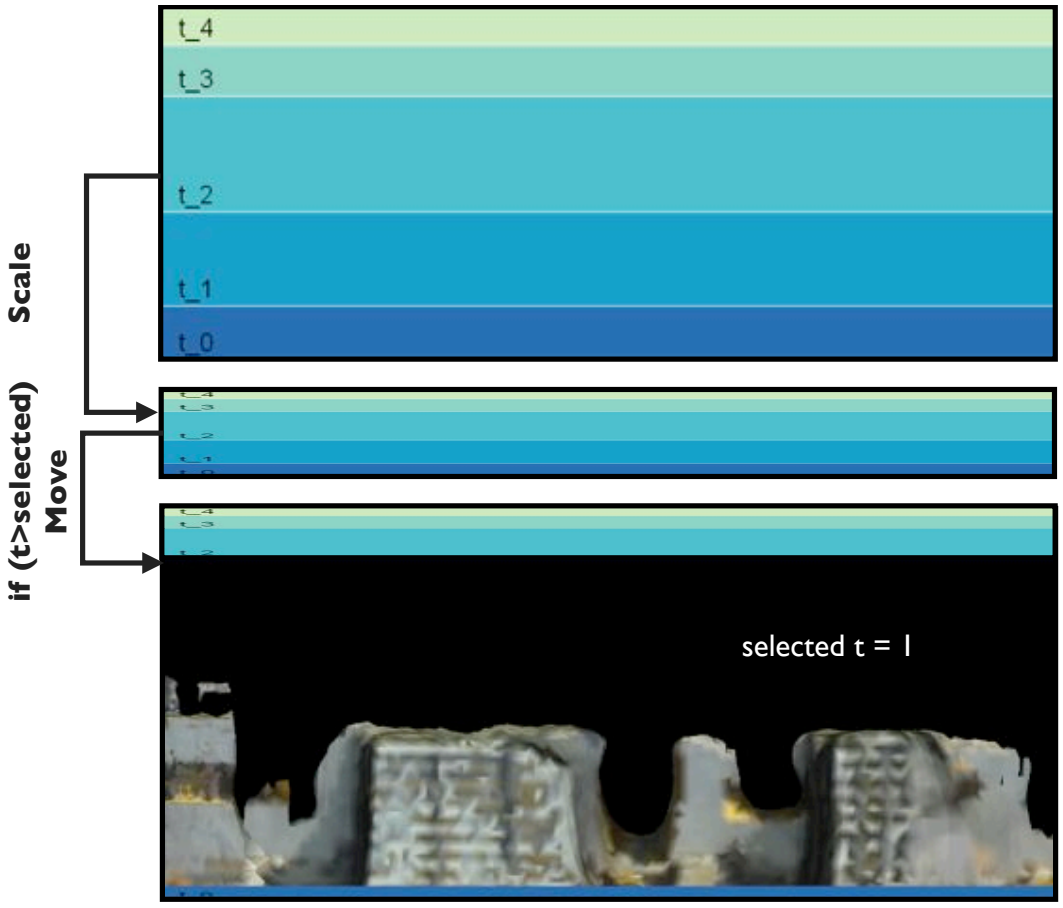


Figure 7.19: Computation of distorted view. The abstract information (average completion) is scaled down in such a way that it does not occlude to much of the detailed information. Afterwards all abstract information representing a time larger than the selected point in time are moved to the beginning of the diagram.

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.

To visualize the 4D data in context to real world, we apply the concept for overview and detail visualization of 4D data in AR. This allows us to visualize geometric changes of construction sites over time directly on-site. The main requirements for the visualization is (1) to preserve context of 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

Figure 7.20: 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) Therefore, the user may want to inspect this point in time further by using the interactive tool to combine abstract information with reconstruction data.

visualization for 4D data. The main idea of our approach is to visualize this data in an AR view to preserve the context of the real world, which makes it for a construction site manager easier to reproduce and understand progress, and possible mistakes directly linked to the real construction site.

Reconstructed 3D models of construction sites are already applied to visualize the progress of construction-sites. By using the as-built 3D reconstruction to analyze the current status of the site in relation to the as-planned model and overlaying registered photographs with the as-built status, the approach of Golparvar-Fard et al. enables an overview of the construction site status [37]. Prepared BIM data is color-coded depending on the progress, completion or other values and overlaid to a distance view of the construction site. Nevertheless, this approach does not provide a possibility to inspect the data in detail over time. With our approach the construction site staff are able to inspect progress in relation to the real-world.

Data To derive the 4D information that represents the as-built status of an observed construction site at different point in time, 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 an MAV and capturing a set of meaningful camera images on a regular basis allows creating 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 We used the mobile setup 3.3 to visualize the status of a construction site at different points in time with the proposed visualization concept. By presenting progress and completion information about the site, construction site stuff has the possibility to inspect the progress of the site right in place. This allows to search for problematic points or bottlenecks. Figure 7.20 shows one of the scenarios where such a 4D visualization may be helpful to support a supervisor on 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. After further inspections of the 3D reconstruction of this

point in time, he noticed that a scaffolding was replaced by a concrete wall and thus, the reason for the regression was not a mistake in the building process.

7.7 Summary

The aim of this chapter was to contribute to the research of the visualization of complex data and time-oriented data in AR. In particular, since fast developing 3D reconstruction techniques allow to provide more and more 3D data and arise the question how to visualize it. For this purpose, we started with techniques that rely on information filtering to avoid information clutter and provide the user with means for comparing a limited number of different data sets. Thereby the user can apply interactive focus&context techniques to control the area of interest.

However, the information filtering methods have some limitations. Firstly, it is complicated 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. To approach these problems, in the second section, we introduced a method that uses information abstraction to display more data in one view. By following the information visualization mantra, we are able to switch between a very abstract visualization to more detailed one.

We showed visualization techniques for displaying multiple 3D data sets (so called 4D data) on-site with AR. 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 visualization mantra of Shneiderman [103] and provide an overview of multiple objects, over multiple points in time, while still being providing detailed information about an object for a select point in time. The visualization levels are supported by interactive transitions between them that provide the important geometric reference between abstract overview visualization levels among each other and finally also to the real world object.

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 and monitoring. For the visualization of construction site progress, we used construction-site related measurements such as completion or progress 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 we focus the visualization on 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 since we plan to apply the visualization technique for other use cases as well, we want to extend this approach for other geometric shapes. The basic idea is that all of them can be approximated by multiple planar subsurfaces, this allows us to display the information on these subsurfaces. 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 overlaying professional virtual information from ACE industries to the user's view in outdoor environments. In this context, we identified the following problems as being most influential in refraining the user from understanding the embedded information:

- Insufficient scene integration,
- Insufficient depth cues,
- Information clutter

To gain more insight into this topic, we discussed how other researchers approached these problems in other 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 that reflects the different existing visualization techniques in AR. These adapted visualization pipelines allow one to identify required data sources and data flows for a previously classified technique. For instance, if an application designer knows that she wants to address the problem of insufficient scene integration, she can use the classification and their corresponding visualization pipelines to easily identify which data source she needs and what result can be achieved using the corresponding techniques. The classification helped us to address the problems that were not solved so far.

We started by implementing techniques that address the problem of an insufficient scene integration in scenes where nothing more than a semantics 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 everything we want to visualize is located under the ground is given. Thus, we do not need any additional depth information. Nevertheless, it is important to provide the user with a set of meaningful depth cues. Otherwise the virtual underground object will be perceived as floating over the ground due to an incomplete scene integration. The question what is an meaningful depth cue was answered by previous work with using edges as occlusion cues. However, we made the experience that these information is not always sufficient, in particular when working in urban scenarios. In street scenes we often experienced outstanding elements such as street markings or colored elements that were not reflected by the existing approaches. Edges seemed to be not able to provide enough cues for a correct scene appearance. To address this problem, we developed a method that extracts more importance information from camera imagery and uses it as occlusion cues. This information include salient regions, edges and texture details. Additionally, our method uses perceptual grouping to analyze the characteristics of each region. In image regions that were identified to contain not enough importance information, the method adds artificial occlusion cues such as stipples and hatching. With a user study, we could confirm 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 (**H2**). As expected the study showed that for these urban scenes, edges are not sufficient for providing enough visual cues for a convincing scene integration, but that our method is able to provide enough occlusion cues that the users are able to understand the subsurface location of virtual objects.

Nonetheless, not for all applications the depth order is known as for the visualization of subsurface infrastructure. For instance, when visualizing as-planned infrastructure objects, such as lamps or new buildings, the depth order between the virtual objects and the physical objects in the environment is not always known. However, 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. Unfortunately, they are often too sparse to provide correct occlusion cues or shadows. To address this problem we developed a method that allows to combine sparse depth information from GIS database with accurately registered camera imagery from an AR setup. Based on these camera images, we can compute a segmentation that uses the projecting 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 database, we are able to compute a 2.5D model of the physical world object. We showed how this 2.5D model can then be used to create correct occlusions or shadows. This confirmed 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 were able to achieve a convincing integration of virtual content into the physical outdoor environment. However, these techniques often allow the user only to understand the order of virtual and physical objects. This is not enough for applications that require a more exact depth judgment. For instance, for construction

workers it is often important to determine the depth of buried objects to adjust the digger or to determine the location on the ground were they should start to dig for an subsurface object of interest. For these application ordinal depth cues are not enough and the existing depth cues are not enough. In these cases we need to provide the user with more than just natural depth cues. For this purpose, we developed a set of additional graphical hints, which we call virtual cues. These cues aim to support the user in determining the depth or the height of virtual objects. They focus on restoring the pictorial depth cue *height in visual field*, since this cue is able to provide absolute depth measurements for objects that are connected to the ground. These cues comprise virtual excavations with virtual scale textures, connection lines, artificial shadows and virtual junctions and were used to support 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 geo-referenced data into a graphical representation. The challenge is increased when working with GIS databases that require some flexibility for data modifications. To address this problem, we further proposed an approach that allows to maintain consistency between a set of graphical representation and the data in the geo-spatial database. We achieved this by storing the graphical representations and pure data in different data levels. A transmission layer builds the connection between both and updates them if changes due to interactive modification appear. This confirms our third hypothesis **H3**. We tested this approach with different GIS databases from several professional companies from civil engineering. In addition to these tests, we conducted interviews with the expert users from these companies. The results showed a positive feedback for the proposed visualization techniques.

The last problem that we addressed in this thesis is the problem of information clutter that appears when visualizing complex data. In the discussion of previous work, we showed that there are approaches available that 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 the progress on construction sites. We showed that it is possible to use these methods as long as one wants to visualize a small number of different data sets (confirming our forth hypothesis **H4**). Nevertheless, as we demonstrated in chapter 7, for visualizing the progress of construction sites it is important to be able to visualize a higher number of different datasets in one view. Only this allows one to search for sources of constructional flaws or to monitor the progress over a bigger time period. For this purpose, we proposed an approach that allows to visualize more than two or three 3D datasets at the same time in AR overlay. The method combines abstraction with focus&context techniques and allows inspecting and comparing different points in time in one view and selecting points of interest for further inspections. This finally confirms our fifth hypothesis **H5** assuming that there are automatic mehtods that use abstraction to remove information clutter for an AR visualization.

The techniques developed in this thesis have the main goal to advance the usage of AR in professional applications from ACE industries that focuses on outdoor usage. We were able to show that it is possible to achieve comprehensible and convincing visualizations that help the user 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 using the adequate visualization techniques in combination with accurate registration techniques.

8.2 Lessons learned

The findings of this thesis provide insights how information can be integrated successfully into an AR overlay. We showed that AR requires visualization techniques that adapts to different conditions, such as:

- The current physical environment,
- The requirements of the user,
- The available data.

If for instance the *current physical environment* has a big 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 big amount of important image structures with visualization techniques that do not preserve them. Our research showed that is more likely to achieve a convincing scene integration if these structures are maintained. In chapter 6, we showed that it can be helpful to provide the user with additional graphical hints, if tasks require to do absolute depth estimations. The last item of the list has been shown in chapter 7 by demonstrating that complex data requires specific filtering and abstraction operation to provide comprehensible visualizations. This is especially the case when visualizing multiple datasets in one view.

Similar requirements exists for adaptive interfaces [69] 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 [41]. The adaption to the physical environment was also discussed by researchers, but except for location-based systems is rather uncommon [15]. In AR the physical environment has a larger influence, since the visualization is in direct relationship to the environment.

In this thesis, we showed that the adaption of the visualization is possible by:

- Analyzing the physical context using geo-spatial data,
- Analyzing the physical context using camera imagery,
- Deriving additional graphical hints from the available data,
- Reducing information clutter using information filtering and abstraction.

These findings augmented the findings from our work from 2009 [57].

There are some more specific insights from our work that can be helpful for further research in AR visualization. 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 [24, 64].

Another interesting experience that we had during our research is that collaborations with industrial partners provide interesting 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 help to show the usefulness of the developed visualization techniques.

8.3 Future work

We can further apply the findings about adaptive visualizations in AR to derive guidelines for more general new interfaces that build up on a combination of virtual and physical information. This especially interesting, with for the new generation of display systems that focus more on directly embedding information into our physical surroundings instead of simply showing it on a closed display, such as Google glass[1] or even more conceptual interfaces like the Sixth Sense [81]. For these kind of interfaces sophisticated technologies that provide intelligent ways of integrating virtual content into our physical environment become more and more important.

The group of Maes summarizes interfaces that integrate virtual information into our physical environment under the title *Fluid Interfaces*[2]. 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 Displays and and even 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, head up)
- 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 of important physical world information, similar methods like the preservation techniques from chapter 7 could be applied. The main question here is the same as for the X-Ray visualization, how to find a good compromise between visibility of virtual information and the information about the physical environment.

Additional graphical hints about important physical world objects can guide the user to pay attention about these objects. For instance, if the user is reading a online newspaper on his mobile phone, a graphical hint can indicate the distances to street crossings helping to avoid dangerous situations with cars.

[1] `http://www.google.com/glass`
[2] `http://fluid.media.mit.edu`

Furthermore, for the the integration of complex information into these kind of interfaces, methods that avoid information clutter such as information filtering and abstraction can help to improve the comprehensibility of these visualizations.

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
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
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

INFORMED CONSENT FORM

RESEARCH STUDY: Ghostings in Augmented Reality

INVESTIGATOR: Stefanie Zollmann (+43 316 873 5079)

INTRODUCTION
You are invited to take part in a research study. Before you decide to be part of this study, you need to understand the risks and benefits. This consent form provides information about the research study. A staff member will be available to answer your questions and provide further explanations. If you agree to take part in the research study, you will be asked to sign this consent form.
Your participation in this study is voluntary. You are free to choose whether or not you will take part in the study.

PURPOSE
The Institute for Computer Graphics and Vision of the Graz University of Technology is carrying out research to gather feedback on Ghost-Visualizations Techniques in Augmented Reality.

PROCEDURES
The intention of this study is to investigate different Ghosting techniques. We present you a set of different scenes and visualizations. For each scene and each visualization you will be asked to rate your depth perception. After rating you will be asked to outline the virtual objects you see in the scene with your mouse pointer.

POSSIBLE RISKS
You have to watch a computer screen displaying the interface for an extended period. You have to interact with the computer using a mouse and a keyboard. Should you feel any discomfort and feel that you cannot continue with the experiment or if you would like to take a break please advise the experimenter.

BENEFITS
The results of this experiment will help to evaluate new Ghosting techniques for X-ray Visualizations in Augmented Reality.

COMPENSATION
By signing this consent form you acknowledge and agree that, in the event that this research project results in the development of any marketable product, you will have no ownership interest in the product and no right to share in any profits from its sale or commercialization.

PAYMENT FOR INJURY OR HARM
Taking part in this research study carries the risks explained under the section entitled "POSSIBLE RISKS". If you require immediate medical care and are not a hospitalized patient, you should seek immediate medical treatment. Otherwise the investigator will help you get the care you need. The Graz University of Technology, Institute for Computer Graphics and/or principal investigator will not pay for the care. Likewise, the Graz University of Technology, Institute for Computer Graphics and/or principal investigator will not pay you for pain, worry, lost income, or non-medical care costs that might occur from taking part in this research study.

RIGHT TO WITHDRAW FROM STUDY
Participation in this experiment is voluntary. You are free to withdraw your consent and discontinue your participation in the experiment at any time, without prejudice to you.

CONFIDENTIALITY
The data collected from your participation shall be kept confidential, and will not be released to anyone except to the researchers directly involved in this project. Your data will be assigned a "Subject Number." When reporting on the data collected from this experiment, only this subject number will be used when referring directly to your data.

QUESTIONS
Should you have any questions about the study, the principal investigator may be reached at:

Stefanie Zollmann
Institute for Computer Graphics and Vision, Graz University of Technology
Inffeldgasse 16 / Second floor, A-8010, Graz, Austria
Telephone: +43 316 873 5079 – Email: zollmann@icg.tugraz.at

SIGNATURES
By signing this consent form, you affirm that you have read this informed consent form, the study has been explained to you, your questions have been answered, and you agree to take part in this study. You do not give up any legal rights by signing this informed consent form. You will receive a copy of this consent form.

_____	_____	_____
Participant (Print name)	Signature	Date

If you sign underneath, you authorize publication of photographs, videos, sound recordings or other materials taken from this study for scientific purposes. You understand that TU Graz will own the copyright to these materials and may grant permission for use of these materials for teaching, research, scientific meetings, and other professional publications. These materials may appear in print and online and the public may have access to them.

	_____	_____
	Signature	Date

INVESTIGATOR STATEMENT
I certify that the research study has been explained to the above individual by me or my research staff including the purpose, the procedures, the possible risks and the potential benefits associated with participation in this research study. Any questions raised have been answered to the individual's satisfaction.

Stefanie Zollmann	_____	_____
Investigator (Print or type name)	Signature	Date

If you have any questions or concerns about this experiment, or its implementation, we will be happy to discuss them with you.

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?

1	2	3	4	5

Visualization Technique "A" – Visualization "Subsurface"

A_A: The subsurface visualization using the X-Ray technique "A" was confusing.

I strongly disagree							I strongly agree

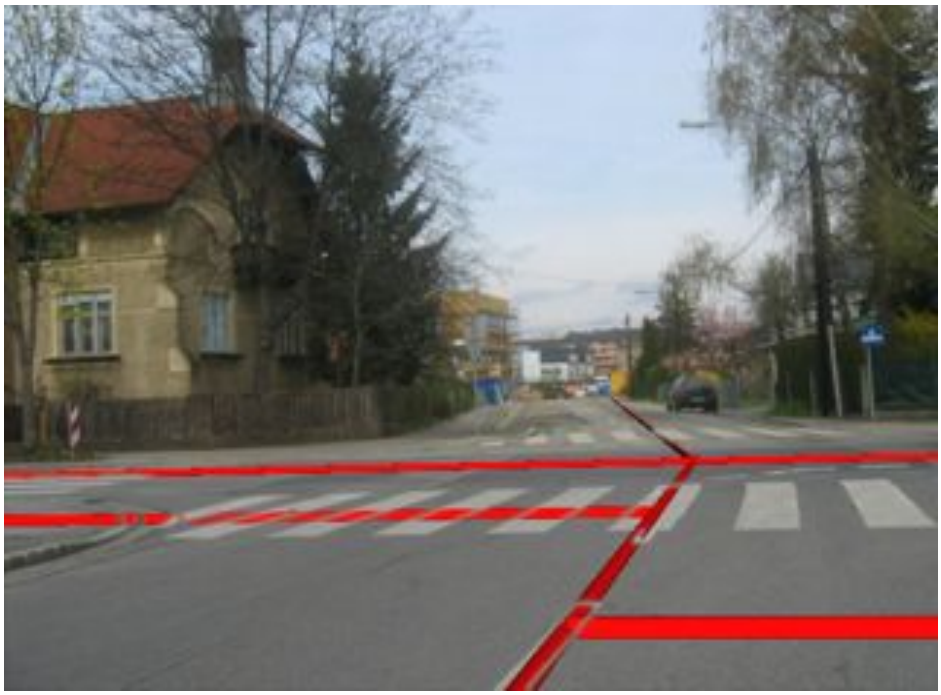
A_B: The subsurface location of virtual objects in the scene was hard to understand using visualization technique "A" .

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**.

Alpha Blending: _____

Edges: _____

Image-based Ghostings: _____

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?).

Alpha Blending: _____

Edges: _____

Image-based Ghostings: _____

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?).

Alpha Blending: _____

Edges: _____

Image-based Ghostings: _____

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)*, number Ic.

[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] Avery, B. (2009). *X-Ray Vision for Mobile Outdoor Augmented Reality*. PhD thesis.

[5] 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.

[6] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385.

[7] Bae, S., Agarwala, A., and Durand, F. (2010). Computational rephotography. *ACM Transactions on Graphics*, 29(3):1–15.

[8] 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.

[9] 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*.

[10] 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.

[11] 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.

[12] 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.

[13] Bruckner, S., Grimm, S., Kanitsar, A., and Gröller, M. (2005). Illustrative context-preserving volume rendering. In *Proceedings of EUROVIS*, volume 1.

[14] 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.

[15] Brusilovsky, P. (2001). Adaptive hypermedia. *User modeling and user-adapted interaction*, pages 87–110.

[16] Canny, J. (1986). A Computational Approach to Edge Detection. {*IEEE*} *Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698.

[17] 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.

[18] Coffin, C. and Höllerer, T. (2006). Interactive perspective cut-away views for general 3D scenes. *3D User Interfaces (3DUI 2006)*, pages 25–28.

[19] 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*.

[20] Cutting, J. E. (1997). How the eye measures reality and virtual reality. *Behavior Research Methods, Instruments, & Computers*, 29(1):27–36.

[21] 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.

[22] 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.

[23] 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.

[24] DiVerdi, S. and Hollerer, T. (2006). Image-space correction of AR registration errors using graphics hardware. In *IEEE Virtual Reality Conference (VR 2006)*.

[25] 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.

[26] 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.

[27] Feiner, S. and Seligmann, D. (1992). Cutaways and ghosting: satisfying visibility constraints in dynamic 3D illustrations. *The Visual Computer*.

[28] Feiner, S. K. and Duncan Seligmann, D. (1992). Cutaways And Ghosting: Satisfying Visibility Constraints In Dynamic 3d Illustrations. *The Visual Computer*, 8:292–302.

[29] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181.

[30] 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.

[31] 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.

[32] 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.

[33] 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)*.

[34] Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Towards Internet-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 1434–1441.

[35] Goldstein, E. B. (2001). *Sensation and Perception*. Wadsworth Publishing Company, 6 edition.

[36] 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.

[37] 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.

[38] 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*.

[39] 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.

[40] 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)*.

[41] 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)*.

[42] 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)*.

[43] Haber, R. and McNaab, D. (1990). Visualization idioms: a conceptual model for scientific visualization systems. *IEEE Computer Society Press*.

[44] 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.

[45] Harris, R. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press.

[46] Hodges, E. R. S., editor (2003). *The Guild Handbook of Scientific Illustration*. John Wiley & Sons, Hoboken, NJ, 2\textsupe edition.

[47] Hoiem, D., Efros, A. A., and Hebert, M. (2005). Automatic photo pop-up. *ACM Transactions on Graphics (TOG)*, 24(3):577.

[48] 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.

[49] Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(April):629–642.

[50] 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.

[51] 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. ISPRS*.

[52] 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*, volume 22, pages 3–11. IEEE COMPUTER SOC.

[53] Kalkofen, D. (2009). *Illustrative X-Ray Visualization in Augmented Reality Environments*. PhD thesis.

[54] 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.

[55] Kalkofen, D., Mendez, E., and Schmalstieg, D. (2009a). Comprehensible Visualization for Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):193–204.

[56] Kalkofen, D., Tatzgern, M., and Schmalstieg, D. (2009b). Explosion Diagrams in Augmented Reality. *IEEE Virtual Reality Conference (VR 2009)*, 0:71–78.

[57] 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)*.

[58] 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*.

[59] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proc. Symposium on Geometry Processing*, pages 61–70.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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*.

[67] 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"*.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] Lex, A., Streit, M., Schulz, H.-J., Partl, C., Schmalstieg, D., Park, P. J., and Gehlenborg, N. (2012). {StratomeX:} Visual Analysis of {Large-Scale} Heterogeneous Genomics Data for Cancer Subtype Characterization. *Computer Graphics Forum {(EuroVis} '12)*, 31(3):1175–1184.

[73] 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.

[74] 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)*, volume 15, pages 175–184.

[75] 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*.

[76] 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*.

[77] Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, volume 8. W.H. Freeman.

[78] Mendez, E. (2010). *Visualization On the Usage of Context for Augmented Reality*. PhD thesis.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] Perlin, K. (1985). An image synthesizer. *ACM SIGGRAPH Computer Graphics*, 19(3):287–296.

[87] 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.

[88] 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.

[89] Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 1, pages 10–17 vol.1.

[90] 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.

[91] Rosten, E., Reitmayr, G., and Drummond, T. (2005). Real-time video annotations for augmented reality. *Advances in Visual Computing*.

[92] 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.

[93] 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.

[94] Santner, J., Pock, T., and Bischof, H. (2011). Interactive multi-label segmentation. *Computer Vision- ACCV 2010*, (813396):397–410.

[95] 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)*.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] Schoenfelder, R. and Schmalstieg, D. (2008). Augmented Reality for Industrial Building Acceptance. In *IEEE Virtual Reality Conference (VR 2008)*, pages 83–90.

[103] Shneiderman, B. (1996). The Eyes Have It: {A} Task by Data Type Taxonomy for Information Visualizations. In *IEEE Visual Languages*, pages 336–343.

[104] Simons, D. J. (2000). Current Approaches to Change Blindness. *Psychology*, 7:1–15.

[105] Simons, D. J. and Levin, T. (1997). Change blindness. *Trends in Cognitive Sciences*, 1(7):261–267.

[106] 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.

[107] Spence, R. and Apperley, M. (2011). *Bifocal Display*. The Interaction Design Foundation, Aarhus, Denmark.

[108] Steiner, J., Zollmann, S., and Reitmayr, G. (2011). Incremental Superpixels for Real-Time Video Analysis. In *Computer Vision Winter Workshop*.

[109] Tatzgern, M., Kalkofen, D., and Schmalstieg, D. (2013). Dynamic Compact Visualizations for Augmented Reality. In *IEEE Virtual Reality Conference (VR 2013)*.

[110] 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.

[111] 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)*.

[112] 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.

[113] 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.

[114] Wagner, D. and Mulloni, A. (2010). Real-time panoramic mapping and tracking on mobile phones. In *IEEE Virtual Reality Conference (VR 2010)*.

[115] 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.

[116] 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)*, number section 4, pages 65–68. Ieee.

[117] Wither, J., DiVerdi, S., and Höllerer, T. (2009). Annotation in outdoor augmented reality. *Computers & Graphics*, 33(6):679–689.

[118] 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)*.

[119] 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.

[120] 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)*.

[121] 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.

[122] 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.

[123] 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.