Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Annotation in outdoor augmented reality

Jason Wither [a,b,*], Stephen DiVerdi [a,c], Tobias Höllerer [a]

[a] Four Eyes Laboratory, University of California, Santa Barbara, CA, USA
[b] Nokia Research Center Hollywood, Santa Monica, CA, USA
[c] Advanced Technology Lab, Adobe Systems Inc., San Jose, CA, USA

ARTICLE INFO

Article history:
Received 12 November 2008
Received in revised form
23 April 2009
Accepted 5 June 2009

Keywords:
Augmented reality
Annotation
Taxonomy
Online content creation

ABSTRACT

Annotation, the process of adding extra virtual information to an object, is one of the most common uses for augmented reality. Although annotation is widely used in augmented reality, there is no general agreed-upon definition of what precisely constitutes an annotation in this context. In this paper, we propose a taxonomy of annotation, describing what constitutes an annotation and outlining different dimensions along which annotation can vary. Using this taxonomy we also highlight what styles of annotation are used in different types of applications and areas where further work needs to be done to improve annotation.

Through our taxonomy we found two primary categories into which annotations in current applications fall. Some annotations present information that is directly related to the object they are annotating, while others are only indirectly related to the object that is being annotated. We also found that there are very few applications that enable the user to either edit or create new annotations online. Instead, most applications rely on content that is created in various offline processes. There are, however, many advantages to online annotation. We summarize and formalize our recent work in this field by presenting the steps needed to build an online annotation system, looking most closely at techniques for placing annotations from a distance.

1. Introduction

Augmented reality (AR) annotations are a powerful way to give users more information about the world around them. The advantage of AR over books or other offline data sources is that the information can be presented at the same location as the object it relates to. This provides context for the information, often making it more engaging and easier to understand. The ability to contextualize and localize virtual information is one of the greatest strengths of augmented reality technology, which is why annotations make up a large portion of all AR content. Interactive guide systems are illustrative examples of how AR annotations can be useful. They cover a wide variety of applications, including helping a user move around a college campus by directly labeling buildings [1], helping a user to read a subway map [2], and giving more information about different installations in a museum [3]. Because of their role in a diverse set of AR application, annotations convey their information to the user in many different ways.

In Section 2 of this paper, we present an annotation taxonomy to organize and better understand the many properties of AR annotations. By creating our taxonomy, we hope to more clearly define what constitutes an AR annotation and to provide a framework for further discussion of annotation techniques. We use our taxonomy to classify 26 completed research applications in which annotations are prominently featured. We focus on applications, rather than techniques, in order to emphasize practical uses of annotations.

As a result of our classification, we identify trends in how annotations are used both within and across application domains. One such trend is that there are two general types of annotations primarily in use today: the first directly adds information about a particular real world object, and the second adds information in a more indirect or abstract way. This dichotomy of annotation types illustrates the broad range of applications that AR annotation can be useful for.

Another result of our classification is the identification of portions of the annotation space that are relatively unexplored. By highlighting these areas, we hope to illustrate continuing and future areas of research to improve annotation techniques. For example, we found that there are very few applications using annotations that can be either edited or created online. This presents an opportunity for significant advances in AR annotation.

There are many examples that demonstrate the advantages of online annotation creation over offline techniques. Applications in

* Corresponding author at: Nokia Research Center Hollywood, Santa Monica, CA, USA.
E-mail addresses: jason.wither@nokia.com (J. Wither), sdiverdi@adobe.com (S. DiVerdi), holl@cs.ucsb.edu (T. Höllerer).

0097-8493/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2009.06.001

fields like construction, where the real world is not static, can benefit greatly from online annotation. Most offline annotation is done using some virtual representation of the real environment. However, in a changing environment such as a construction site, the virtual representation used to build annotations must constantly be updated to reflect the real world. By creating annotations directly in the real world, no model of the environment is necessary.

Similarly, applications in which building a model of a real world object is the primary task can greatly benefit from being done on site. By doing the modeling in an outdoor AR application the user can work in the environment being modeled, saving time and making it possible for more of the modeling to be done automatically. For example, textures could be automatically gathered and applied while the user does the actual modeling, and computer vision could aid in the construction of the model.

Another compelling reason to enable online annotation creation is to leverage the full user base of future AR applications for content creation. In many current AR systems, one of the biggest drawbacks is the lack of content, largely because it is very difficult for an individual or small team to create content on a global scale. However, when AR becomes popular on mainstream devices, there will be a large group of users who will also be able to add content. With online annotation techniques, this could lead to an explosion of user created content similar to the increase of online content with web 2.0 applications.

In Section 4 of this paper, we discuss the steps necessary to design and implement a system for online annotation creation. Within this space we will focus on interaction techniques for creating new annotations by defining four approaches to place new annotations in 3D space from a distance. We compare these techniques, discussing when each is most useful and presenting exemplary implementations of each technique.

2. Annotation taxonomy

In this section, we first propose a general definition of what constitutes an AR annotation by specifying two simple properties that all annotations must have. We then further explore the space of AR annotations by looking at several dimensions along which annotations can vary. This creates a picture of what types of annotations are currently used in different applications domains, and other general annotation trends. It will also give an idea of areas where more work could enhance existing annotation types or develop new types of annotation altogether.

While we know of no other taxonomies of annotation in augmented reality, Hansen [4] has constructed a taxonomy of ubiquitous annotation systems. His taxonomy is much broader, comparing annotation in augmented reality to that in virtual reality, ubiquitous computing, and web presentation. Because of the broad range of annotation types discussed in Hansen's taxonomy he was not able to discuss many of the AR specific aspects of annotation that we examine in our taxonomy. Our more narrow focus allows us to gain novel insight into AR annotation trends, beyond established results.

2.1. Annotation definition

Our definition of AR annotations is quite general to encompass a broad range of uses. Within this general definition, we further divide the space into different categories depending on different annotation characteristics. Our definition of an AR annotation is simply:

> *An augmented reality annotation is virtual information that describes in some way, and is registered to, an existing object.*

This definition allows for virtual information to be presented in a variety of ways, including text, pictures, models, sounds, or even haptic feedback. The relationship between the virtual information and the existing object does not have to be clearly defined—there are many cases where this relationship is not obvious, even though the information is intended as an annotation. A stricter definition might be limited to content that clearly relates a single object, such as tour guide annotations [5]. However, this is overly restrictive, as it makes classification as an annotation dependent on the author's intent and user's interpretation. Rather than try to determine on a case by case basis if the relationship is direct enough, we chose a more encompassing definition. This allows us to consider many uses for annotation that might have not been so apparent otherwise.

We also take a very broad view of what can be the existing object. It can range in size and specificity from a particular part of a machine, to the whole machine, a room full of machines, or a whole factory. There is also no requirement that it is a single physical object. It could be all the windows on the side of a building or a semantic area like the engineering part of a college campus. Again, the purpose is to allow our definition to be very broad so as to not arbitrarily throw out some types of annotations.

To fit within our definition there are two essential components that every AR annotation must have. Each annotation must have some spatially dependent component that links it to the existing object being annotated, and some spatially independent component which includes the virtual content. While other properties might change the usefulness of an annotation we feel that these two components are necessary and sufficient to define what is and is not an annotation.

2.1.1. Spatially dependent component

The spatially dependent component that every annotation must have is a link between the virtual and the real world. This means that not only do the virtual annotations have to be registered in some world coordinate system, but they also have to be registered to a particular object. The virtual content of the annotation must in some way be related to the world around it.

This does not necessarily require a direct semantic relationship between the virtual content and the object that is being annotated—such as the virtual content containing the name of the object—but there must be some relationship between the two. To illustrate this concept, consider an indoor tracking system that was set up in one room where specific objects were annotated. If that tracking system was then moved to another room, and the same application was run the virtual objects would appear in the same relative coordinate system that they had in the original room. However, they would no longer be annotations because they would no longer have any spatial relevance to their surroundings. Likewise, if a real world object is annotated, and then the real world object is removed the virtual content is no longer an annotation because the real world object is no longer present. An example of virtual content in an AR environment that is not an annotation can be seen in Klein and Murray's paper on tracking for AR [6], where registered virtual content was used to demonstrate the quality of the tracking. That virtual content had no relationship with the surrounding environment, however. Many ARToolKit [7] applications also do not employ annotation. Although the content is registered to a marker it does not have a relationship with anything physical other than the tracking fiducial. There are many unpublished examples of applications like this, such as the downloadable ARTetris game created at the HIT Lab New Zealand [8]. This game could be played anywhere because the content is not related in anyway to the world around it.

2.1.2. Spatially independent component

All annotations must also have a spatially independent component. This means that there must be some difference between the virtual content and what the user sees of the real world. For example, a perfect model of an object (which could be used for rendering correct occlusions) is not an annotation even though it is highly spatially dependent. As soon as some spatially independent content is added, however, the virtual object can become an annotation. If a perfect virtual model did not have the same texture as the object but instead was all green, we would then call it an annotation because it is in some way adding to or changing the real world object. When virtual content does not modify the user's perception of the scene in some way though, it cannot be called an annotation no matter the spatial complexity.

2.2. Dimensions of annotation

While the two previously mentioned requirements are sufficient to give a broad definition of annotations, there are many other ways that annotations can vary. Here, we present six orthogonal dimensions to describe annotations. By using these dimensions, we can both better classify existing work and suggest areas for future work. Hansen [4] described four challenges for ubiquitous annotations—anchoring, structure, presentation, and editing—which are related to our six dimensions of AR annotations. Anchoring, structure, and presentation all describe how and where an annotation is placed with regard to the object it annotates. We cover this with location complexity and location movement. Hansen describes editing as the challenge of creating new annotations. Our taxonomy represents online editing and creation of annotations with the interactivity dimension. In addition to these dimensions, we also define content complexity, semantic relevance, and annotation permanence, to quantify the annotation's content.

2.2.1. Location complexity

The first of our taxonomy's dimensions is the complexity of the annotation's real world location. Since all annotations must have some spatially dependent component, there is always a location associated with the annotation. The simplest location an annotation can have is a single 3D point. In this case the annotation would only have position information and any orientation for the annotation would have to be arbitrarily defined by the application. It is still possible to have complex content registered to a single point, however. For example, a virtual animated character could be registered at a single point. The model complexity in this case does not translate to location complexity because the entire model is only fixed to one real world point and the model does not rebuild or extend something that exists in the real world.

The next step in complexity of an annotation's location is orientation. With both position and orientation, it becomes possible to do things like orient textual annotations to a building surface. While it is often enough to locate an annotation at a single 6DOF point, having higher location complexity is also possible. Higher location complexity means that the object being annotated is no longer simply represented by a point, but instead by a representative 2D or 3D region. This region can be something like a bounding box around the object or an accurate model.

Increasing location complexity should not be confused with increasing the complexity of the annotation itself, however. A very simple textual annotation could still have high location complexity if instead of being registered to a point, it is registered using an aligned model of a real object. Higher location complexity is useful for giving a more exact picture of what is being annotated, as well as affording easier interaction with annotations. Outlines or bounding boxes can be drawn around physical objects that have been annotated to highlight the entire object. Users can then use VR-like interaction techniques to select objects to view their annotations, since the entire physical object is associated with the annotation.

In some cases the object being annotated is not a single real world object, but a group of objects or a semantic region like the interior of a room [9] or a building courtyard [10]. In these cases the region being annotated is somewhat abstract and is often best represented by a bounding region. Frequently the annotation will also move throughout the space being annotated. For example, in AR Façade [9] the virtual characters are annotating the room they are in and are free to move throughout that space. While their location complexity is moderately high (a bounding region of the room), their location movement is very high as they move freely throughout the region they annotate.

2.2.2. Location movement

While all annotations have to be registered to a real world object, the location of the actual annotation can move greatly depending on application and user preference. This dimension describes how much the virtual part of the annotation moves. Many types of annotations can exhibit location movement, including view managed labels [11], navigation paths that move with the user [12], or animated characters in the scene [13]. However, the cause of the motion is not important. This dimension measures both how far the annotation can be from the annotated object and how much freedom of movement there is. It does not attempt to measure the movement of the real object or any movement contained within the annotation.

2.2.3. Semantic relevance

Semantic relevance is a measure of how closely the annotation is related to the physical object it is annotating. There are many ways an annotation can relate to its annotated object. The list of descriptors that we feel best encompasses the types of semantic relationships between annotation and object: names, describes, adds to, modifies, and directs to. While each of these descriptors can have high or low semantic relevance, for the most part annotations that have certain semantic descriptors will have higher relevance than others. For example, annotations where the semantic descriptors *names* or *describes* can be used to describe the relationship will generally have higher semantic relevance than those where the relationship is better described as *adds to*, *modifies*, or *directs to*. While the latter set of modifiers can relate directly to the annotated object they are also frequently used to provide information that is not as directly related to the object. Conversely, the first set of modifiers nearly always provides direct information about the object being annotated.

To clarify the types of relationships described by each semantic descriptor, below is a list of the descriptors with definitions and examples.

- *Names*: The annotation simply provides a name for the object it is annotating. An example of this can be seen in the Touring Machine developed by Feiner et al. [1], where campus buildings are annotated with their names.
- *Describes*: In this case, the annotation provides more information than just the name of the object. Often this is a textual description like those presented by Reitmayr and Schmalstieg [12]. *Describes* can also be used for annotations like those used by Mizell and Caudell [14] to describe how to complete a task.
- *Adds to*: This descriptor best describes cases when the extra virtual content added to the real world somehow changes it, rather than just adding extra information. One such example is

new objects added to a park, such as trees, statues, or picnic tables [15]. Another example is a completed version of a building under construction or in ruins [16]. AR games like AR Quake [10], AR Pacman [17], and AR Façade [9] are also good examples.
- *Modifies*: The *modifies* descriptor describes annotations that visually change existing objects. This type of change is illustrated by both X-ray vision techniques [18], where a structure is modified so that internal objects are visible, and historical guides [19], where a modern building can be replaced or modified to appear as it did in the past. There is a distinction here between *modifying* a completed building by changing its appearance and *adding to* a ruined building. The first case changes existing real world content by overwriting it with virtual content, while the second case adds content where there would otherwise be nothing.
- *Directs to*: The *directs to* descriptor can be used for annotations that direct a user to a certain destination, either by laying out a path to that destination [12] or by displaying an arrow or some other information [20] to direct the user to the annotated object.

2.2.4. *Content complexity*

Annotation content complexity can vary greatly. The simplest annotation content is just a point marking some object of interest. On the other end of the spectrum, content can consist of animated 3D polygonal models with audio. This delimits the very broad range of content possible for annotations. In general content complexity can be determined by both the amount of information that is imparted to the user by the annotation and the visual complexity of the annotation.

Simple annotations, like text labels that just name an object, have very low content complexity. More extensive text that also describes the object has a slightly higher complexity. As the visual complexity increases from text to image to video, the content complexity naturally also increases. 3D content is not always more complex than 2D content. For example, a bounding box is less complex than video content. As things like polygon count and texture use increase in 3D content, the visual complexity increases and correspondingly so does the content complexity. Generally, this increased visual complexity will also increase the information that is contained in the annotation, increasing the content complexity further. We classify things like text labels as having low content complexity, images and simple 3D content as medium complexity, and detailed 3D content as high content complexity.

2.2.5. *Interactivity*

Interactivity is another important aspect of any annotation. We distinguish four distinct states of interactivity along a continuum. First are the annotations that are created in an offline process and are static during the application [21,22]. The user can only view these annotations. Next are the annotations that are interactive but not editable [9,3]. The user may be able to interact with these annotations in some way, perhaps even in a complicated way, but cannot add to or change any of the information that is stored by the system. Further interactivity comes with the ability of the online user to edit the content of existing annotations [12]. This requires direct interaction between the user and the annotation, allowing the user to make changes to existing content or add new content to previously constructed annotation locations. The most interactive are the annotations that users can create while using the system [23,15]. This is similar to editable annotations, while also allowing the user to pick the location for the annotation.

2.2.6. *Annotation permanence*

There are many reasons why an annotation may not always be visible to a user. A common reason to selectively hide some annotations in an application is the threat of information overload. If there are too many annotations visible at once, it becomes impossible to determine which annotations correspond to which objects. We have found that there are four basic strategies that control annotation permanence, allowing only the most relevant annotations to be visible at any time. We list these strategies below with descriptions and examples:

- *Temporally controlled permanence*: In this approach annotations are only visible for a certain amount of time and are often temporally ordered as well. This is particularly useful for annotations that are directions for a user, such as the assembly instructions in Feiner et al.'s work [24]. In this example, having annotations only visible at certain times creates an implicit order that gives even more information than having them all visible at once would.
- *User controlled permanence*: This approach allows the user to directly control which annotations are visible at any given time. One example of this is Bane and Höllerer's [18] X-ray vision system where the user can select which annotations to show by looking around and moving a viewing volume closer and further away.
- *Spatially controlled permanence*: Spatially controlled permanence can be achieved with hot-spots [25,3], where annotations are only visible for a user at a specific location. Another option that can be used when the system has a complete world model is to use real world objects to occlude more distant annotations [26].
- *Information filtered permanence*: It is also possible to filter the visible annotations based on a system unique to each application. Filtering in this way might include highlighting certain annotations that are particularly salient to the user's current view or attention [27]. Similarly, annotations could be visible based on other aspects of application state, such as in AR games [17,10], where virtual object visibility is determined by game progress.

2.3. *Application areas*

We have plotted exemplary applications from the five most common categories of AR applications along the different axes of our taxonomy, using an approach similar to that used by Elmqvist and Tsigas [28]. We chose these categories of applications based on previous categorizations [29] and activity of recent research. The resulting chart can be seen in Fig. 1. The applications we chose from each application area are listed below. We feel that this set of applications is representative of the most common applications in each domain, while also covering each domain's breadth of possible applications. An important note is that we only chose examples where a complete application has been demonstrated. There exists other previous work, including our own [30,23,26], that demonstrates techniques for things like creatable content. However, these works do not include full applications demonstrating those techniques, and so are not included in our taxonomy.

- *Assembly and construction*: Baillot et al. [31], Feiner et al. [24], Mizell [14], Piekarski and Thomas [15], Tang et al. [22], and Zauner et al. [32].

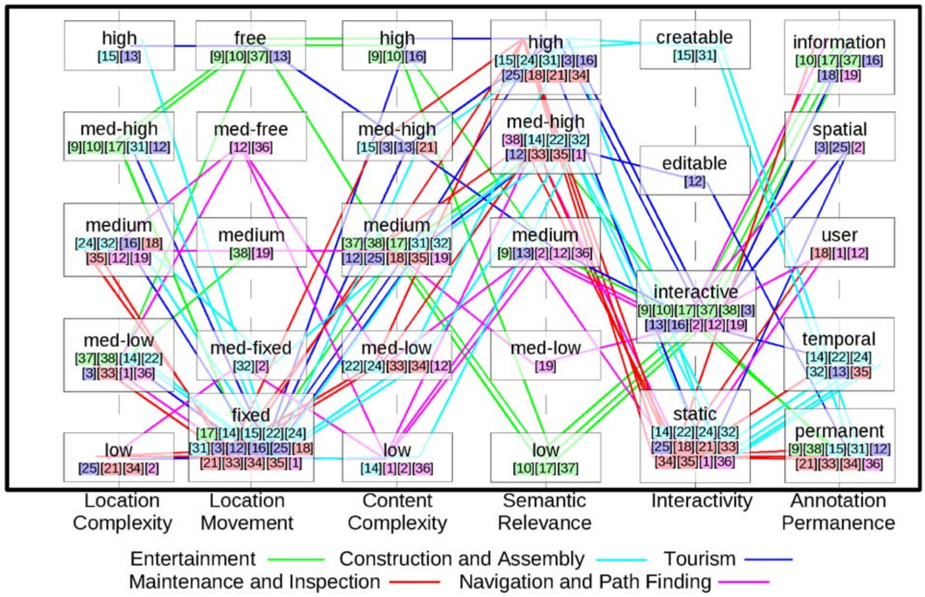

Fig. 1. The results of 26 applications plotted in our taxonomy. Different colored lines represent different application areas. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- *Maintenance and inspection*: Bane and Höllerer [18], Georgeli et al. [21], King et al. [33], Olwal et al. [34], and Platonov et al. [35].
- *Navigation and path finding*: Eaddy et al. [2], Feiner et al. [1], Guven and Feiner [19], Reitmayr and Schmalstieg [12], and Tonnis et al. [36].
- *Tourism*: Dow et al. [25], Reitmayr and Schmalstieg [12], Schmalstieg and Wagner [3], Schmeil and Broll [13], and Vlahakis et al. [16].
- *In situ entertainment*: Cheok et al. [17], Dow et al. [9], Lindt et al. [37], Thomas et al. [10], and Wagner et al. [38].

3. Discussion

There are several trends that become apparent by examining the different dimensions of our taxonomy. Some dimensions do not figure prominently in this discussion primarily because variation along those dimensions is very application specific. For example, while content complexity is linked somewhat with content creation techniques, it can also be designed based and can vary from application to application. The type of annotation permanence used in an application can also change based on design requirements that differ across the application spectrum. While these dimensions are useful in distinguishing one annotation from another they are not useful in determining larger trends within the annotation space.

3.1. Two primary categories of annotations

Identifying trends in existing work is a major motivation for creating a taxonomy. One such trend that our taxonomy illustrates is that there are two distinct, commonly used annotation creation techniques. We call these *direct* and *indirect* annotation.

Direct annotation consists of static information about a single real world object. This type of annotation, which is exemplified in the work by Mizell and Caudell [14] on providing instructions for bundling wires, seems to be most prevalent in construction, maintenance, and tourism applications, where general informational annotations are common. This type of annotation can be of varying location complexity, but generally has little location movement since information is about specific objects. Likewise, the semantic relevance is generally high because the main goal of the annotation is to provide more information about the object. The semantic descriptors that go along best with this goal of providing more information about the annotated object are the *names* and *describes* descriptors. We have grouped annotations of this type in Fig. 2. Fig. 3 shows a different visualization, in which we have plotted the location movement and semantic relevance axes of our taxonomy, as these axes best distinguish different styles of annotations. In this chart, the direct annotations form a tight cluster in the upper left quadrant.

The second primary type of annotation is more abstract or indirect in its relationship to the real object. Dow et al.'s [9] AR Façade work provides a good example of this type of annotation, which does not necessarily describe a particular object in the real world, but instead provides virtual content that exists in a defined real world space. For the sake of our discussion, we call these annotations *indirect*. In general, indirect annotation has less semantic relevance to the annotated object and greater freedom of movement of the annotation. This is shown in both Fig. 4, where only indirect annotations are plotted, and Fig. 3, where they are in the lower right quadrant. As can be seen, this type of annotation is not as homogeneous as direct annotations, which makes sense because annotations of this type are trying to accomplish a variety of things, while the purpose of direct annotations is more specific. The semantic descriptors that best describe the relationship between the annotation and the object for indirect annotation are *adds to*, *modifies*, and *directs to*. These three descriptors can also be used for more direct annotations. However, unlike the *names* and *describes* descriptors, they can also be used to describe indirect annotations.

Indirect annotation is most common in entertainment and navigation applications. In the case of entertainment applications, the lack of semantic relevance is primarily because the annotations are part of the overall game, as in AR Façade. The fact that the annotations are part of a larger application also means that individual annotations are generally more free to move throughout the scene. Together, these two properties can transform the scene into something different, such as transforming a campus into a Quake level, rather than just adding more information about the campus.

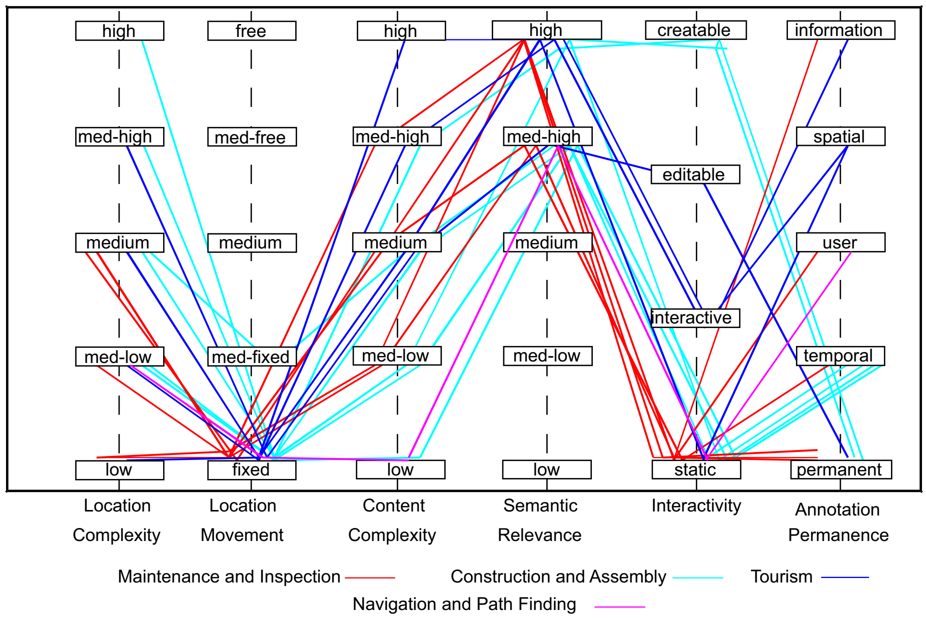

Fig. 2. Applications that make use of direct annotation. This style of annotation generally has very little location movement, and high semantic relevance.

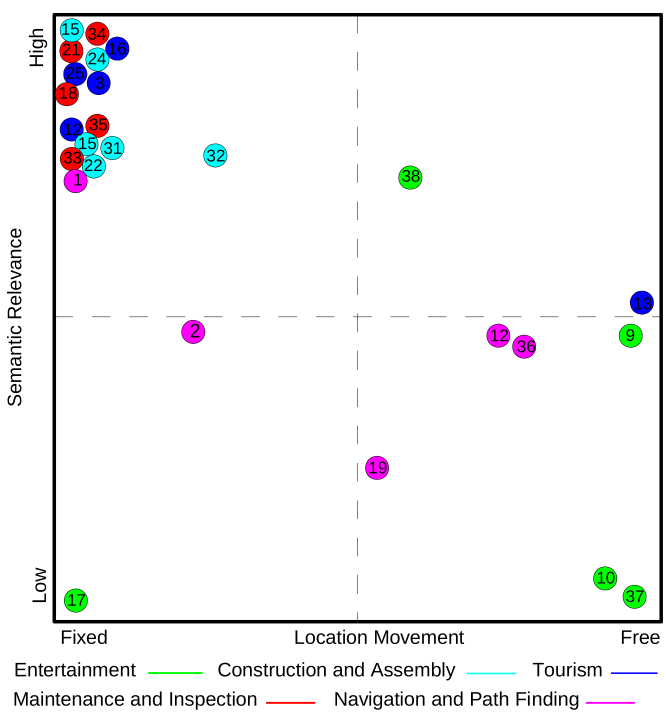

Fig. 3. The location movement and semantic relevance axes of all annotations. These two axes combined illustrate different styles of annotation.

In the case of navigation, annotations can frequently be used as virtual way points directing a user. While these annotations do not directly describe the object they are nearest to visually, they can be thought of as an extended annotation of the object they are guiding users to. In both of these cases, the annotations have less direct semantic relevance to their surroundings and are much freer to move around, since they are not describing a single object directly. They still should be considered annotations though, since they both still add virtual information to an object, meeting our requirement of having spatially dependent and spatially independent content.

In Fig. 3, we have not yet discussed the upper right or lower left quadrants. These quadrants are also interesting, although not as heavily populated. In the upper right quadrant are annotations with high semantic relevance that are still free to move. An application that employs view management [11] to move annotations in the user's field of view is a good fit in this quadrant. Conversely, the lower left quadrant has static annotations that have little semantic relevance. Two uses for annotations in this quadrant are art and entertainment [17], where annotations might be static, like a painting or graffiti on a wall, but have little to do with their location.

While currently both direct and indirect annotations occur predominantly in different domains, it is interesting to think about what applications would look like in each domain if they used the non-dominant style of annotation for that domain. For example, consider a tourism application that uses indirect annotation. Schmeil and Broll [13] have developed an application in this space with an avatar-driven tour guide. Their guide also provides other functions, but the interesting observation in this case is that an annotation (the avatar) can also describe the world around the user. Without providing direct, fixed, annotations, it can still describe objects in the scene by virtue of the avatar walking around and gesturing to objects being described. In this way, the avatar annotation can indirectly annotate different objects in the scene.

Construction, assembly, maintenance, and inspection applications also generally use direct annotation, whether it is to direct users on how to complete a task [24], or to inspect a factory [21]. However, it could also be possible to have applications with indirect annotation in these domains. One possible application that would use less direct annotation is a more general training application instead of step by step assembly instructions. This could be done with an avatar or animated virtual representations of system parts the user is working with, so the user can achieve a more holistic understanding of how things function, rather than just learning a series of steps.

It is also possible to have an entertainment or navigation application that uses direct annotation. One example is an AR

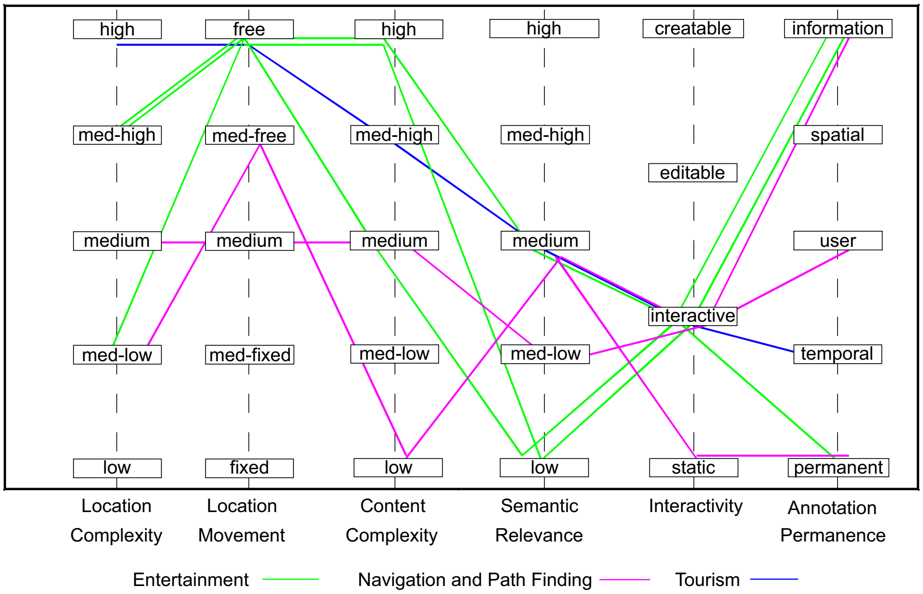

Fig. 4. Applications that make use of indirect or abstract annotation. These annotations generally have more freedom to move, and less semantic relevance than direct annotations.

game that relies on direct interaction with the real world. This could either be a treasure hunt style game (find object *X* and annotate it) or an educational style game such as Expedition Schatzsuche presented by Schmalstieg and Wagner [3]. We have classified their application as a tourist application since it also acts as a museum guide, but it could just as easily be classified as an entertainment application.

3.2. Content creation

Another trend our taxonomy highlights is that very few existing applications have complex content, and those that do generally only operate in a limited area. There are even fewer applications that allow users to edit or create new annotations online. We feel that these two trends are directly linked. While it is possible to create complex content for any application, it is generally very time consuming. Within the scope of AR, it can be even more difficult because registering the content in a global coordinate system is also difficult. However, being able to create or edit annotations online is also very powerful. There are many new applications that become possible with online annotation and editing, and others that become much easier to build. Such applications would in turn enable many more people to create content. Locating the content correctly also becomes a much more easier process, since it is then possible to directly see the intended location.

To illustrate the advantages of online content creation and editing more clearly, here are several scenarios where online content modification is necessary.

First, online content editing includes multiple types of operations. It can mean just changing the content's appearance, or it can mean editing the location (spatially dependent component). Being able to edit the spatially dependent component of the annotation requires very similar tools to creating new annotations. Somehow the user must be able to correctly position existing annotations in the real world.

Architecture or landscaping applications provide a good class of application to show how moving new or existing annotations can be useful. Consider a user who wants to landscape the courtyard of a new building. Using AR, they could position things like virtual trees, sculptures, fountains, flower beds, and paths, until the desired arrangement is achieved. In such an application, not only is the final position of the annotation important, but it is also important to be able to see the movement of the annotation to that location since there are no strict constraints on the annotations position. Because the object being annotated (the courtyard) is abstract and larger than the annotation, the positioning of the virtual content needs to be free and user controllable, so that the user can place the annotation where they wish within the space. At the same time, the user needs to have some idea of the global placement of objects so things do not end up looking right from just one position and wrong from others. Being able to see the annotations in the real world while they are being placed is very important in this case because the point of the application is to position things where they look best, something that may be harder to do in an offline annotation procedure.

Another case where online annotation creation is useful is in assembly or maintenance applications. In either of these domains, one of the most important uses for AR is to direct the user on how to perform some task. The ability to build the set of instruction annotations in place is intuitive and would save time for the annotation editor. It would also provide extra insurance the annotations were correctly placed, as the editor would view the created annotations in the actual environment.

Similarly, in the field of tourism applications, a tourist guide could be built much more easily on site. Because the application operates over a large area, in situ annotation becomes even more important, especially if the objects being annotated are small enough that global offline data sources like aerial photographs would not be accurate enough for offline annotation. Naturally, tracking accuracy is a problem regardless of the annotation technique used, but for many tourist applications tracking can be recalibrated visually at each point of interest, allowing for accurate local tracking within a global space. The size of objects that can be tracked and annotated will depend on the proximity of the user to the object, making a global data source at one scale less useful for annotation.

As mentioned in the introduction, an even more compelling reason to have creatable annotations would be such that all users could create annotations. In the domain of tourism, this could have many potential applications. For example, an AR tourist guide wiki would allow any user to add spatially referenced information about points of interest.

User creatable content can be useful for more general applications, making an AR 2.0 type of application possible. Inspired by web 2.0, AR 2.0 has been defined by Dieter Schmalstieg and Wolfgang Broll, among others, as a general class of applications that depend on user created content. Aside from tourist information, this could be used for general location specific message boards or social networking. To make this concept more concrete, consider an example application in a mall. Users and businesses could annotate store fronts or individual items that they wish to point out to their friends or customers. Users could also leave other location specific messages for friends, indicating directions to meeting points or recommendations for particular stores or restaurants.

To realize AR 2.0 applications, there are still several technological hurdles to overcome. In the next section we will discuss these hurdles and introduce our work on some of them.

4. Creating annotations

There are several components of a successful annotation system, each of which must be carefully selected or developed. We feel the three areas that are most important are selecting the correct hardware components, developing successful 3DOF and 6DOF tracking techniques, and evaluating the best techniques for completing an annotation.

4.1. Hardware selection

Choosing the best hardware for the task is the first step that must be undertaken when building an AR annotation system. The desired components differ somewhat from many AR systems because input is also very important. Choosing the best devices for input and output in a wearable system is an area of ongoing research. Because the user must be mobile, traditional desktop based hardware choices cannot be used. Devices for outdoor AR have been studied in various contexts, many of which apply to annotation. Both Thomas [39] and Zucco et al. [40] have looked at using different hand held devices for input while using systems with head worn displays. Others like Wagner et al. [38] and Schmalstieg and Wagner [3] have explored using hand held AR displays for displaying AR content to the user and for interacting with existing annotations.

We have conducted two studies to look more precisely at which devices are best for annotation. Our first study [41] compared several mobile devices for moving a distant cursor in 3D, an integral part of many annotation techniques. In this study, we found that when using a tracked HMD, head motion can be an efficient mode of panning a 3D cursor, while a trackball or buttons are best for changing the distance of the cursor. Our second study [42] compared a hand held display with a head worn display for annotation. There are many reasons to pick both types of displays for AR applications, but in this study we specifically were interested in how well each display performed in annotation tasks. We found that a magic lens style hand held display allowed for the fastest input. However, we found no significant difference between hand held displays and head mounted displays when studying how fast users can find objects that need to be annotated in the world around them.

4.2. Tracking

Having accurate tracking is critically important for online annotation creation. While tracking is important for all AR applications, it becomes even more important for annotation because any errors are permanently stored with the location of the created annotation. It is especially important to have accurate orientation tracking, since in many outdoor environments, users are generally annotating larger objects like features on buildings. The large distance to the created annotations will turn any small orientation errors into large position errors.

There are many people working on accurate tracking for mobile AR, of which many of the most successful use either only computer vision or a hybrid of computer vision and other sensors. Reitmayr and Drummond [43] demonstrated an excellent 6DOF tracking system for a mobile user using computer vision and a textured model of the surrounding environment. Klein and Murray [6] have also demonstrated a very robust optical tracking system using a SLAM variant. However, for most outdoor environments, SLAM is not as robust an approach since most objects are far away and users are operating in a large environment. DiVerdi et al. [44] have developed a system for orientation tracking from a static location using optical flow and SIFT. Often this approach can be sufficient for annotation creation. Generally, a user is not going to create new annotations while moving, so sometimes it is only necessary to track orientation when the user stops. This is particularly true for certain AR modalities like hand held AR, where users do not look at the display continuously. Instead, the user only looks at the display when they are engaged with a piece of content, often while standing still.

4.3. Annotation techniques

The last and most unique component for creating an outdoor AR annotation system is developing and testing techniques for creating the actual annotation. Creating an annotation is a difficult procedure because the user has to find an accurate 3D position for the annotation. If the user wishes to create an annotation at their current location [45] this problem is much easier, particularly for a tracked user. However, in most cases, it is more convenient to annotate objects that are not exactly at the same location as the user. Therefore, creating an annotation generally has two steps: locating the object that will be annotated and determining the distance to that object. We have found four general ways of determining the distance: interaction with an existing model, estimation, triangulation, and measurement. Each approach has advantages and disadvantages which are discussed below.

4.3.1. Model based annotation

If there is a model of the environment around the user then annotation is straightforward, as interaction techniques from VR are applicable. For example, ray casting [46] can be used in conjunction with the model. The user simply has to look at and click on the object they wish to annotate, and a ray can be cast out through the correct pixel in the image plane. This ray can then be intersected with the virtual geometry that is aligned with the real world, with the closest intersection point used as the position for the annotation. This technique, demonstrated by Reitmayr and Schmalstieg [12], is very accurate (assuming an accurate model and tracking), and places little burden on the user. However, a complete model of the environment must exist for it to be possible, and even with a complete model new annotations can only be placed directly on modeled structures.

4.3.2. Estimation based annotation

Estimation puts most of the burden of finding the correct placement for the annotation on the user. The user must move a 3D cursor to the location of the annotation they wish to create by estimating how far away it is. In general, users are very poor at making this estimation without training. However, it is possible to improve users' estimations by using virtual depth cues, particularly when there are other objects nearby that are already annotated. In a previous study [30], we found that the accuracy of users' annotations increased (i.e., the annotations were placed closer to the object users were trying to annotate) when using our AR depth cues. However, even with our depth cues and other already annotated objects nearby, users were on average nearly 10% off when estimating the distance to objects they wished to annotate. We developed several depth cues for our study and found that the most preferred cue is the top down view of all annotations in the scene, which can be seen in Fig. 5. This cue allowed users to see all the annotations from a second perspective, giving users a better idea of the distances of the already annotated objects.

One advantage of using estimation as a technique for annotation is that users are not constrained to place things at any particular position in the real world, and can more actively move annotations around. Having this extra freedom can be advantageous in cases like the AR landscaping application previously discussed.

4.3.3. Triangulation based annotation

One way to increase the accuracy of annotations in unmodeled environments is to use triangulation to create annotations. Rays are cast from two different locations at the same object, and the annotation is created at the intersection of the two rays. This technique has the advantage of not requiring the user to make an explicit depth estimate, instead computing the 3D coordinate through some equation. There are several ways to do triangulation. The most straightforward is to have the user look at an object from two different locations. This approach has been used by Baillot et al. [31], as well as Piekarski and Thomas [15] in their outdoor modeling applications. It has the drawback though of requiring users to walk to two locations for each annotation they wish to create—a time consuming process.

It is also possible to create annotations with triangulation from a static location by using aerial photographs [23]. The user casts a ray at the desired object and is then presented with an aerial photograph showing that ray as well as the user's location. The user has to move the annotation to the correct location along the ray to complete the annotation. Using this process, it is possible to annotate many different kinds of features that are visible in aerial photographs. In our work, we broke features that were visible in aerial photographs into three broad categories of objects to annotate—corners, edges, and regions—which can be seen in Fig. 6 in the aerial photograph, and in Fig. 7 from the user's perspective. These types of features allow a user to annotate objects with a varying amount of location complexity. These features could naturally also be used together in an application to place annotations at even more complex locations.

In general, using triangulation and particularly aerial photographs has the advantage over other annotation techniques of allowing users to annotate things that they cannot see during the annotation process. For example, a user could sight down a building wall and then place several annotations along that wall, or using the aerial photographs, a user could annotate features on the other side of a building.

4.3.4. Measurement based annotation

The last way to create an annotation at a distance is to measure the distance. From the user perspective, this approach is as good as interacting with a model. With the right equipment, the user simply has to look at the object they wish to annotate and push a button for the annotation to be completed. In terms of user complexity, this is the ideal interface for creating new annotations and the kind of interface that will make AR 2.0 applications wide spread. The downside of this approach is that it takes specialized equipment to measure the distance between the user and the object to be annotated. To demonstrate this technique, we have integrated a small laser range finder into our wearable system.

Fig. 5. The user preferred top down view aid for annotation when estimating the distance to the object.

Our laser range finder, an OptiLogic RS400, works well because it weighs less than 8 ounces and is small enough to be integrated into either a hand held or a head mounted wearable system. This range finder can be used with objects up to 400 m away and is accurate to ± 1 m or better. The range finder can also give calibrated distance readings at 10 Hz, allowing for real time annotation creation. We have mounted the range finder next to the camera in our video see through system so that the viewing angle of the camera and the laser from the range finder are parallel.

To create an annotation using the laser range finder, users must use display motion (head motion for an HMD) to align the physical object they wish to annotate with a cursor on the screen that shows the pixel that is currently being ranged. Once the object and the cursor are aligned, a button press creates a virtual annotation at the same location as the real world object by casting a ray from the user through the cursor's location in the view frustum and then placing the point at the distance along that ray that is returned by the laser range finder. Because the range finder gives updates at 10 Hz, it is also possible to give annotations higher location complexity. As we discuss in our previous work [26], it is straightforward to measure orientation by sweeping the range finder across a surface, allowing annotations orientation to match the annotated objects. The laser range finder, along with computer vision, also makes it possible to build more complex representations of objects that could be used for annotation as well.

Fig. 6. The aerial photograph view from within the aerial photograph based annotation system. In the lower left corner, an insert of the video feed from the display mounted camera can be seen. The user's position and orientation are represented on the photograph with a small cone avatar. A small set of features have already been annotated—two corners (the green points), one edge (the green line), and one region (the transparent green rectangle). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Conclusions

Annotation accounts for the majority of virtual content in augmented reality. In this paper, we have developed a taxonomy to specify more exactly what constitutes an annotation in augmented reality, and to determine how annotation is being used in state of the art AR applications. We found two general types of annotations commonly in use: those that directly add information about real world objects and those that add information in a more indirect or abstract manner. Direct annotation is primarily used in a straightforward way, to simply add explanatory information about a real world object. Indirect annotation can be used for a wider variety of things which only relate to the object that is being annotated. Annotations in this category can consist of things like directions to an object, virtual actors within a scene, or even content that modifies the existing world in some way. While currently direct and indirect annotations are used predominantly in different application areas, this is not a fundamental requirement, and interesting cases can be made for using non-standard annotation types in many application areas.

In our taxonomy we also found that there are very few applications that have either editable or creatable annotations. One reason for the lack of applications in this space is likely because there has been relatively little work on developing techniques to aid online annotation.

To help fill this gap we have formalized the steps needed to build a system capable of online annotation and presented four approaches for creating new annotations. The first of these techniques uses a model, and VR interaction techniques, to place new content at a distance. The second requires the user to estimate the distance to the objects they wish to annotate, but we found that by adding AR depth cues to enhance the user's perceptual sense of depth, users are able to make fairly accurate annotations while still allowing for continuous object viewing and movement. The third approach to place new annotations at a distance is to use triangulation. We discussed an example of this technique which utilizes aerial photographs as a secondary view

Fig. 7. Example annotations as seen by the user in first person view mode. *Left to right:* (a) corner annotations on the corners of two buildings are rendered as cubes. (b) An edge is annotated with a texture mapped onto the plane of the wall it denotes. (c) A region annotation is rendered as a wireframe bounding box. These renderings are not geared toward a particular application; rather, they are for illustrative purposes. Applications using these annotations would have visual representations tailored to their needs.

on the scene around the user. The last approach is to measure the distance from the user to the object they wish to annotate. Our technique utilizing this approach uses a small laser range finder to measure the distance between the user and the object they wish to annotate. This allows for the easiest and fastest annotations, but requires additional hardware. The combination of these techniques also encapsulates the ways it is possible to create new annotations in an unprepared environment, using user estimation, triangulation, and measurement to determine the distance from the user to the annotation that is being constructed.

While online annotation and editing is not currently a common part of AR applications we feel that it is going to be a very important part of next generation AR systems. With the advent of consumer level AR devices the user base for AR applications will explode. Providing ways for this user base to generate their own content will also greatly expand the amount of available content, making AR even more compelling. However, this scenario of AR 2.0 depends on having fast, easy, and intuitive ways to add online content. It is our hope that this work has both shown the need for online annotation and provided an overview of the work necessary to add it to new applications.

References

[1] Feiner S, MacIntyre B, Höllerer T, Webster A. A touring machine: prototyping 3D mobile augmented reality systems for exploring the urban environment. In: Proceedings of the international symposium on wearable computers, 1997. p. 74–81.

[2] Eaddy M, Blasko G, Babcock J, Feiner S. My own private kiosk: privacy-preserving public displays. In: Proceedings of the international symposium on wearable computers, 2004. p. 132–5.

[3] Schmalstieg D, Wagner D. Experiences with handheld augmented reality. In: Proceedings of the international symposium on mixed and augmented reality, 2007. p. 3–18.

[4] Hansen FA. Ubiquitous annotation systems: technologies and challenges. In: Proceedings of the 17th conference on hypertext and hypermedia, 2006. p. 121–32.

[5] Miyashita T, Meier P, Tachikawa T, Orlic S, Eble T, Scholz V, et al. An augmented reality museum guide. In: Proceedings of the international symposium on mixed and augmented reality, 2008. p. 103–6.

[6] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces. In: Proceedings of the 6th international symposium on mixed and augmented reality, 2007. p. 1–10.

[7] Kato H, Billinghurst M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proceedings of the 2nd international workshop on augmented reality, San Francisco, USA, 1999. p. 85–94.

[8] Artetris ⟨http://www.hitlabnz.org/wiki/ARTetris⟩, 2007.

[9] Dow S, Mehta M, Harmon E, MacIntyre B, Mateas M. Presence and engagement in an interactive drama. In: Proceedings of the SIGCHI conference on human factors in computing systems, 2007. p. 1475–84.

[10] Thomas B, Close B, Donoghue J, Squires J, Bondi PD, Piekarski W. First person indoor/outdoor augmented reality application: AR Quake. Personal Ubiquitous Computing 2002;6(1):75–86.

[11] Bell B, Feiner S, Höllerer T. View management for virtual and augmented reality. In: Proceedings of the symposium on user interface software and technology, 2001. p. 101–10.

[12] Reitmayr G, Schmalstieg D. Collaborative augmented reality for outdoor navigation and information browsing. In: Proceedings of the symposium on location based services and telecartography, 2004. p. 31–41.

[13] Schmeil A, Broll W. An anthropomorphic AR-based personal information manager and guide. In: Stephanidis C, editor, HCI (6), Lecture notes in computer science, vol. 4555. Berlin: Springer; 2007. p. 699–708.

[14] Mizell D. Boeing's wire bundle assembly project. In: Barfield W, Caudell T, editors. Fundamentals of wearable computers and augmented reality. London: Lawrence Erlbaum; 2001. p. 447–67.

[15] Piekarski W, Thomas B. Interactive augmented reality techniques for construction at a distance of 3D geometry. In: Proceedings of the workshop on virtual environments, 2003. p. 19–28.

[16] Vlahakis V, Karigiannis J, Tsotros M, Gounaris M, Almeida L, Stricker D, et al. Archeoguide: first results of an augmented reality, mobile computing system in cultural heritage sites. In: Proceedings of the conference on virtual reality, archeology, and cultural heritage, 2001. p. 131–40.

[17] Cheok AD, Goh KH, Liu W, Farbiz F, Fong SW, Teo SL, et al. Human pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. Personal Ubiquitous Computing 2004;8(2):71–81.

[18] Bane R, Höllerer T. Interactive tools for virtual X-ray vision in mobile augmented reality. In: Proceedings of the international symposium on mixed and augmented reality, 2004. p. 231–9.

[19] Guven S, Feiner S. Visualizing and navigating complex situated hypermedia in augmented and virtual reality. In: Proceedings of the international symposium on mixed and augmented reality, 2006. p. 155–8.

[20] Schwerdtfeger B, Klinker G. Supporting order picking with augmented reality. In: Proceedings of the international symposium on mixed and augmented reality, 2008. p. 91–4.

[21] Georgeli P, Schroederi P, Benhimanei S, Hinterstoisseri S, Appel M, Navab N. An industrial augmented reality solution for discrepancy check. In: Proceedings of the international symposium on mixed and augmented reality, 2007. p. 111–5.

[22] Tang A, Owen C, Biocca F, Mou W. Comparative effectiveness of augmented reality in object assembly. In: Proceedings of the SIGCHI conference on human factors in computing systems, 2003. p. 73–80.

[23] Wither J, DiVerdi S, Höllerer T. Using aerial photographs for improved mobile AR annotation. In: Proceedings of the international symposium on mixed and augmented reality, 2006. p. 159–62.

[24] Feiner S, MacIntyre B, Höllerer T. Wearing it out: first steps toward mobile augmented reality systems. Mixed Reality: Merging Real and Virtual Worlds 1999:363–77.

[25] Dow S, Lee J, Oezbek C, Maclntyre B, Bolter JD, Gandy M. Exploring spatial narratives and mixed reality experiences in oakland cemetery. In: Proceedings of the international conference on advances in computer entertainment technology, 2005. p. 51–60.

[26] Wither J, Coffin C, Ventura J, Höllerer T. Fast annotation and modeling with a single-point laser range finder. In: Proceedings of the international symposium on mixed and augmented reality, 2008. p. 65–8.

[27] Kalkofen D, Mendez E, Schmalstieg D. Interactive focus and context visualization for augmented reality. In: Proceedings of the international symposium on mixed and augmented reality, 2007. p. 191–201.

[28] Elmqvist N, Tsigas P, A taxonomy of 3D occlusion management techniques. In: Proceedings of the virtual reality conference, 2007. p. 51–8.

[29] Höllerer T. User interfaces for mobile augmented reality systems. PhD thesis, advisor—Feiner S, Columbia University; 2004.

[30] Wither J, Höllerer T. Pictorial depth cues for outdoor augmented reality. In: Proceedings of the international symposium on wearable computers, 2005. p. 92–9.

[31] Baillot Y, Brown D, Julier S. Authoring of physical models using mobile computers. In: Proceedings of the international symposium on wearable computers, 2001. p. 39–46.

[32] Zauner J, Haller M, Brandl A, Hartmann W. Authoring of a mixed reality assembly instructor for hierarchical structures. In: Proceedings of the international symposium on mixed and augmented reality, 2003. p. 237–46.

[33] King G, Piekarski W, Thomas B. ARVino—outdoor augmented reality visualisation of viticulture GIS data. In: Proceedings of the international symposium on mixed and augmented reality, 2005. p. 52–5.

[34] Olwal A, Lindfors C, Gustafsson J, Kjellberg T, Mattsson L. Astor: an autostereoscopic optical see-through augmented reality system. In: Proceedings of the international symposium on mixed and augmented reality, 2005. p. 24–7.

[35] Platonov J, Heibel H, Meier P, Grollmann B. A mobile markerless AR system for maintenance and repair. In: Proceedings of the international symposium on mixed and augmented reality, 2006. p. 105–8.

[36] Tonnis M, Lange C, Klinker G. Visual longitudinal and lateral driving assistance in the head-up display of cars. In: Proceedings of the international symposium on mixed and augmented reality, 2007. p. 91–4.

[37] Lindt I, Ohlenburg J, Pankoke-Babatz U, Ghellal S. A report on the crossmedia game epidemic menace. Computers in Entertainment 2007;5(1):8.

[38] Wagner D, Pintaric T, Ledermann F, Schmalstieg D. Towards massively multi-user augmented reality on handheld devices. In: Proceedings of the international conference on pervasive computing, 2005.

[39] Thomas B. Evaluation of three input techniques for selection and annotation of physical objects through an augmented reality view. In: Proceedings of the international symposium on mixed and augmented reality, 2006. p. 33–6.

[40] Zucco J, Thomas B, Grimmer K. Evaluation of three wearable computer pointing devices for selection tasks. In: Proceedings of the international symposium on wearable computers, 2005. p. 178–85.

[41] Wither J, Höllerer T. Evaluating techniques for interaction at a distance. In: Proceedings of the international symposium on wearable computers, 2004. p. 124–7.

[42] Wither J, DiVerdi S, Höllerer T. Evaluating display types for AR selection and annotation. In: Proceedings of the international symposium on mixed and augmented reality, 2007. p. 95–8.

[43] Reitmayr G, Drummond T. Going out: robust model-based tracking for outdoor augmented reality. In: Proceedings of the international symposium on mixed and augmented reality, 2006.

[44] DiVerdi S, Wither J, Höllerer T. Envisor: online environment map construction for mixed reality. In: Proceedings of the virtual reality conference, 2008. p. 19–26.

[45] Rekimoto J, Ayatsuka Y, Hayashi K. Augment-able reality: situated communication through physical and digital spaces. In: Proceedings of the international symposium on wearable computers, 1998. p. 68–75.

[46] Bowman DA. Interaction techniques for common tasks in immersive virtual environments: design, evaluation, and application. PhD thesis, advisor—-Hodges LF, Georgia Institute of Technology; 1999.