Sketching up the world: In-situ authoring for mobile Augmented Reality

AUTHORS' NAMES (BLIND FOR REVIEW)
Department
University
COUNTRY
youremailaccount@xxx.xxx.xx

Abstract

We present a novel system allowing in-situ content creation for mobile Augmented Reality in unprepared environments. This system targets smartphones and therefore allows a spontaneous authoring while in place. We describe two different scenarios, which are depending on the size of the working volume and consequently use different tracking techniques. A natural feature-based approach for planar targets is used for small working spaces whereas for larger working volume a panoramic based rotation tracking is deployed. Both are integrated into one system allowing the user to use the same interaction for creating the content applying a set of simple, yet powerful modeling functions for content creation. The resulting AR content can be shared with other users using a dedicated content server or kept in a private inventory for later use.

Keywords: Mixed and Augmented Reality, Content Authoring

1 Introduction

Augmented Reality (AR) is a term for a view of a physical real-world environment whose elements are augmented by virtual computer-generated imagery [1]. In the late 1990ies, first experiments were conducted on presenting geo-referenced content using Augmented Reality as interface. The Touring Machine [4] introduced mobile outdoor AR and showed a campus tour guide.

Even with the limited possibilities of this first prototype, the power of in-situ presentation of geo-referenced information became apparent. The emergence of the World Wide Web as a mass phenomenon around the same time prompted Spohrer in his seminal paper to suggest the "WorldBoard" [13], a combination of distributed online information systems and geo-referenced indexing. Information could be published like in the Web, but indexed by a geographic position rather than by a symbolic URL. He explicitly highlights to use AR as interface to present the published information to the user. Unfortunately, his vision has not been fully realized yet, likely because key technologies were not mature enough.

Nevertheless nowadays the first commercial AR applications like Wikitude[1] were released to the market. That shows the potential of mobile AR. Wikitude, Layar[2] and other AR platforms form the genre of so called AR-browser applications, as their main goal is to browse information, which are visually overlaid over the real world. These AR applications were made with the mass market in mind and target millions of users. Thereby solving technical problems like tracking and visualization are only some important aspects for mobile AR applications. Especially for the end user the availability of content has a high significance. This fact is widely known as the content problem and addressed in the research for AR authoring toolkits, allowing the users to create content for AR. But so far most applications are desktop based and are designed for professional users.

In the following we want to present our work on AR authoring and content distribution, which targets an inexperienced audience and allowing them to create content for Augmented Reality. This will allow the users to actively participate the content creation process by creating content in an AR environment, which can be useful among others for:

- Creation of virtual duplicates of real objects e.g. buildings, which can later be place in a different environment.
- Creation of 3D models that can be used in AR based games, phantom rendering for occlusion handling.

---

[1] http://www.wikitude.org
[2] http://www.layar.com

- Sketching of annotations and 2D drawings, which can be used to place or highlight information, which are displayed using Augmented Reality.

In this paper we will start by introducing existing work focusing on authoring tools for AR as well as content creation for AR. Later we introduce the concept of *Augmented Reality 2.0* (AR 2.0) as the next evolution of Augmented Reality. We present why new approaches for authoring in AR are important in an AR 2.0 system before we start to present our system, which is based on three parts: tracking, modeling and content providing.

2   Related Work

Research of content authoring has a long tradition in augmented reality. There are a dozen of desktop-based authoring solutions, which extend existing 3D modelers like Maya[3] and Blender[4]. DART [9] for example, is one of the best-known authoring toolkits for AR. It comes as a plugin for Macromedia Director targeting professional content producers.

As DART does not allow in-situ authoring other research was conducted for systems allowing authoring of content while in place. OutlinAR [3] presents an approach for in-situ modeling. With OutlinAR the user is able to build wireframe models by using a tracked handheld camera mouse, which is plugged into a standard computer.

Hengel et al. presented an approach [5], which is based on a SLAM (Simultaneous Localization And Mapping) tracking system presented by Klein et al. [8]. They use real time camera tracking, and high-level automated image analysis, for an immersive modeling process, which generates three-dimensional models of real objects. While this system allows spontaneous authoring of AR content on desktop computers it is computational too expensive to run on a phone.

Simon et al. presented an approach for modeling in an outdoor environment [12]. He identifies parallel lines and other information in the image to establish a coordinate system. Based on that he can create geometry, which can also be used for tracking.

Pikarski et al. [10] proposed a 3D constructive solid geometry approach for constructing models within the Tinmith AR system. While Baillot et al.

[2] introduced a more CAD-like interface for creating content in AR.

Despite these existing solutions for creating AR content within desktop/notebook applications only little investigations have been done in the field of content authoring for mobile devices such as smartphones as it is needed for an AR2.0 strategy. In general, we can distinguish cross-platform authoring (at the desktop) and in-situ authoring using smartphones. Only few of the latter exist. Cross-platform content authoring conducted on desktop PCs is presented in [17] and uses a PC simulating the behavior and the interface of a smartphone.

One of the few examples of in-situ authoring conducted using a smartphone is presented by Henrysson et al. [6] by building a simple scene using a smartphone as interaction device to interact with augmented Lego blocks. Later work showed how to manipulate an augmented mesh using a smartphone.

However, none of the above solutions has been designed with a truly large user base of thousands or ten thousands users in mind using smartphones as their primary device for AR content and content creation. This is particularly important, as in the spirit of AR 2.0 every consumer of AR content is potentially also an author. Thus they are not suitable tools for user-centered content creation. Furthermore the existing solutions for smartphones are only designed for small working volumes and require a prepared environment. In the following we want to present our approach for creating content for AR using smartphones.

3   Introducing AR 2.0

Assuming millions of people using mobile AR applications the amount of content should reflect this huge user base and cannot only be created by a few professional content creators.

Fortunately, the recent development of the World Wide Web showed how to create content for millions of users: User participation. We think this is transferable to the described content problem in AR, which implies that the only solution is to allow the user to participate in such a way, that they can create their own content rather than only using content created by a small number of professional modelers. The trend of making users to content creators and not only consumers is one of the main characteristics of the Web 2.0. Following this we define AR 2.0 as the next step of Augmented Reality

---

[3] http://autodesk.com

[4] http://www.blender.org/

Table 1: Comparison of Web 2.0 and AR 2.0 characteristics.

Web 2.0 Characteristics	AR 2.0 Characteristics
Large number of users and web sites (already true for Web 1.0)	Large-scale in number of users as well as working volume
No clearly visible separation between accessing local data and remote data	No clearly visible separation between visualizing local data or remote data
Applications running in a browser behave like local applications, encouraging the user to interact with them	Applications locally running on the device can transparently download modules or new features from remote servers
A huge amount of non technical people retrieve data and contribute or modify it as well	Non expert users can create or update the AR content at specific locations
Information from different sources can be combined and create a new value-added application, in so-called *Mashups*	*Mashups* which access data from sources like traditional web services and combine them with AR content to display them in three-dimensional space

[11] where users can create and share content (see Table 1).

If AR applications are going to be deployed on a massive scale in an AR 2.0 approach, there are several key areas of technology that are needed:

1. A low-cost platform that combines AR display, tracking and processing
2. Mobility to realize AR in a global space
3. Large-scale AR tracking solutions, which work in real time
4. Easy to use authoring tools for creating AR content
5. Backend infrastructure for the distribution of AR content and applications

1. and 2. are solved by using smartphones as platform, in the following we want to present our current work, which focuses on an prototype of an mobile authoring solution for non-professionals. Furthermore, we will present the used tracking solution and the backend infrastructure to distribute the content. All this together allows inexperienced users the authoring of AR content while in place.

4 Tracking for mobile interaction

There are different scales in which in-situ authoring solutions for smartphones could work. In this work we differentiate between small working volumes like desktop environments and larger volumes like outdoor scenes. Small workspaces are characterized that the working volume is mostly bound to one specific object, which can nearly always be seen in the phones camera view.

Larger working volumes are characterized that the environment around the user contains many objects, which cannot always be seen. Both scenarios require a different tracking technique, which is why we treat them separately in this paper.

4.1 Tracking infrastructure for small working volumes

We decided to use a natural feature-based approach for tracking the smartphone in case of a small working volume. Therefore we employed the natural features based tracking system presented in [16] as this system works in real-time on smartphones while being robust against partial occlusions. This tracking approach tracks the position of the phone using a feature database containing features of an object in the current camera view. Thus it requires a feature database of the currently tracked target to be available on the phone.

As we assume an unprepared environment we created a tool, which allows the user to create this feature database using her smartphone. Therefore the user has to point her phone towards the object she wants to use as tracking target, which should have a planar surface. After sketching the outline of the planar surface, the system start to analyze the selected region concerning amount of features and their distribution. Furthermore the system analyzes if the target was captured frontal by checking if the target has a rectangular shape (see Figure 1). We allow small errors as they can be corrected by
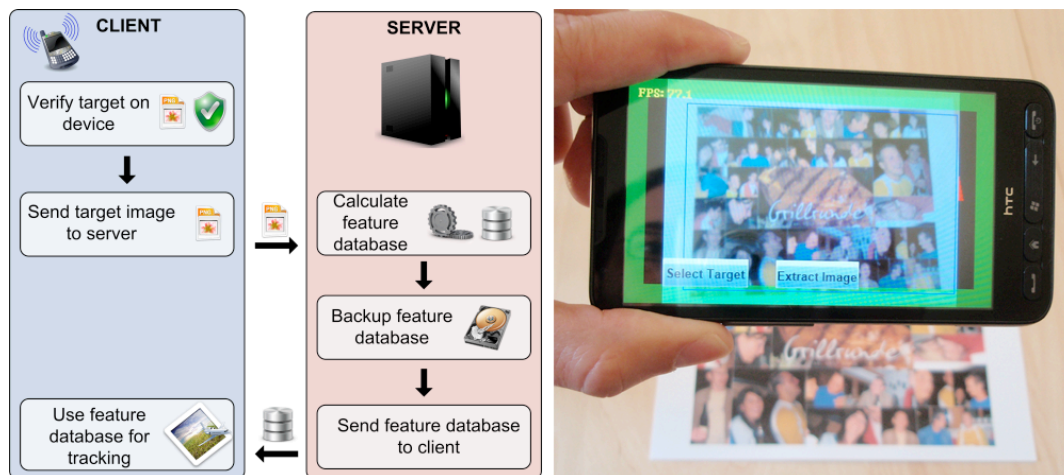
Figure 1: Showing the online creation of a feature database, which can be used for natural feature-based tracking. Left: Illustration of the needed steps to create a feature database using a remote server. Right: Showing the client for creating the feature database. The green display border indicates that the outlined object in the current camera fulfills all criteria's for a good target for natural feature based tracking.

reprojecting the area into a rectangle, but larger errors produce problems creating the feature description. A last check calculates the size of the object in the camera frame alarming the user if the object is too small to create a feasible tracking database from it.

Once all requirements towards the target are fulfilled the system sends the picture of the target to a server and the server creates the natural feature database. Optionally the client running on the smartphone can send the GPS address of the current position (if available) to geo-reference the newly created feature database.

Once created, the natural feature database is then transmitted back to the client. Furthermore, it is stored on the server optionally using the transmitted GPS position as a filter in case another client is later asking for a tracking target on that position.

This means that the user running her AR application on the phone can create a new natural feature database for tracking using her smartphone or query the server for already existing ones. Both assures that the phone is tracked in 6DoF in a previously unprepared environment with minimal effort and minimal delay as the creation of the natural feature database takes 2s in average and the transmission takes between 5-10s assuming a typical database size of less than 200k and a 3G internet connection.

4.2 Tracking infrastructure for large working volumes

As mentioned previously we define a large working volume by having an environment with many

objects, which are not always in the current camera view. The typical scenario for using such a large working volume is authoring in outdoor environments. We therefore replaced the tracking for small working volumes requiring a static tracking object with an approach assuming a static position and allowing only rotational movements. In outdoor environments these assumptions are appropriate, since typically the user will stay at one position and explore the environment by rotating the phone.

Tracking in outdoor environments has very challenging requirements, such as an accurate registration to a given coordinate system, robustness and real-time processing. To satisfy these requirements we used a sensor-fusion based approach [15] that combines panorama-based vision tracking, accelerometer and compass data. Thereby the magnetic compass and the accelerometers provide absolute orientation measurements in the earth coordinate system while the panorama-based vision tracking provides relative orientation tracking.

Since the output of the accelerometers and the compass may suffer from jitter and external influences, visual tracking provides stable but relative orientation measurements. Therefore the panorama-based tracking creates panoramic maps online and tracks simultaneously from the map [14]. Natural feature mapping and tracking is used to generate the panoramic map of the environment from the live camera stream while using it for tracking at the same time.
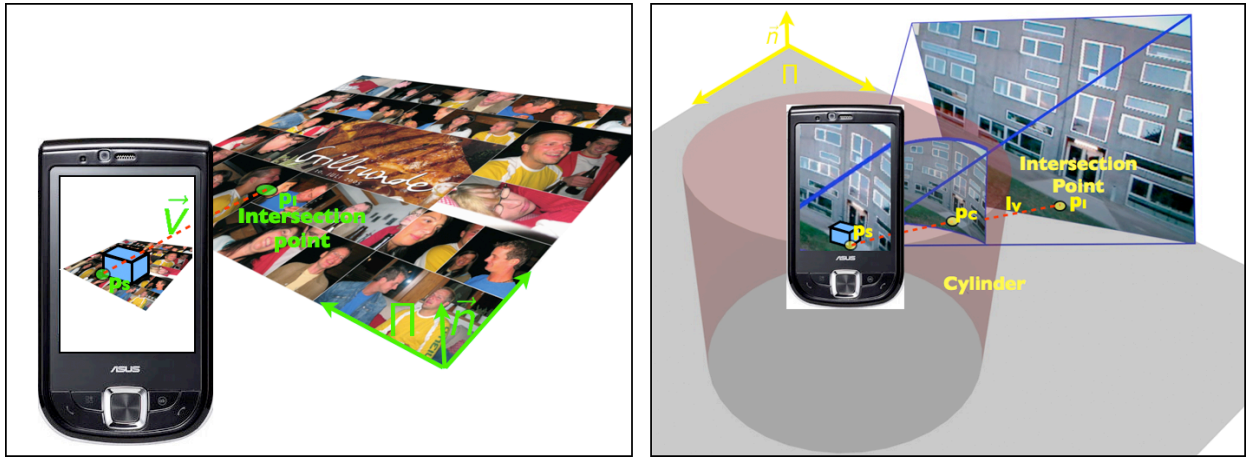
Figure 2: Illustrating the tracking and estimation of the 3D coordinates of the current modeled object, Left: Tracking the phone position in a small workspaces using a natural feature-based tracking of a known planar surface, Right: Tracking the phone position in a large environment using GPS and panorama based vision tracking combined with sensors for tracking the orientation.

To combine the accuracy and robustness of vision tracking with the absolute orientation of the inertial and magnetic sensors, the results of the panorama-based tracking and the sensors are fused in a Kalman filter-based framework.

As this approach only tracks rotational movements we use the GPS-sensor of current generation smartphones to estimate the current position

5 In-situ Modeling

Existing authoring solutions address a specific user group. Most of them target professional content creators and media artists. The few targeting an inexperienced audience are desktop-only applications like BuildAR[5]. In this system we target an inexperienced audience to create content for mobile Augmented Reality. The minimal requirement is that they can handle their phones (taking pictures, sending short messages). Therefore we limited the functionality to basic, yet powerful functions to create 3D content. The functions comprise the generation of 3D primitives, such as cubes and cylinders, but also polygonal geometry, the geometrical modification of the generated primitives, such as scaling and moving and changing the texture of these objects. Additionally, 2D annotations registered to the ground plane can be created.

5.1 Generating registered 3D primitives

To generate 3D primitives the user starts to draw the 2D shape according to the footprint of the primitive by touching the screen. The 2D footprint of the object is generated on the ground plane.

In the case of a small tracked environment the ground plane is the marker plane formed by the planar target used for natural feature tracking. To determine the 3D coordinates of the footprint, the intersection point p_I between the line given by the touched spot on the screen p_S and the viewing direction $\vec{v}$ and the marker plane Π is calculated (see Figure 2 left).

In case of the large working volumes the ground plane is the physical ground. In this case the intersection between a line given by the touched spot on the screen p_S and the viewing direction and the physical ground is determined by using the cylindrical representation of the panoramic map used for tracking. Therefore the 2D point p_S is mapped from the screen space to a 3D point on the cylinder p_C (see Figure 2 right). The intersection line l_V is then given by p_C and the coordinate of the origin of the camera. Intersecting l_V and the ground plane Π determines the 3D intersection point p_I.

The calculated intersection points are then used as input to generate the footprint of the object in the 3-dimensional world space. After defining the 2D footprint on the ground plane the user can extrude this shape to form a 3D object. Hence the supported 3D models range from cubes, tubes and spheres to objects with arbitrary polygonal ground plane.

---

[5] http://www.buildar.co.nz

Figure 3: Example showing created augmented 3D content and interface using the system configured for small working environments, Left: Augmented scene with augmented duplicates of real objects (book and candle) , Middle: Augmented scene showing different object shapes with different applied textures, Right: Augmented scene showing different objetcs with different colors and interface for selecting colors.

5.2 Modifying registered 3D primitives

After creating different 3D objects, the user may want to manipulate the object. Therefore we integrate different manipulation tools. The object can be translated, scaled and rotated by selecting the axis of transformation and by sliding in the corresponding direction over the touchscreen.

5.3 Applying textures to the registered 3D objects

To enable the creation of more realistic models, we provide the functionality to assign different colors to the objects as well as to texture the object (see Figure 3). To assign a new color to the object, the user can select from a color palette as shown in Figure 3 (right). To assign a new texture, the user can select from a set of predefined textures, which can be mapped to the object (Figure 3 middle). Furthermore, it is possible to create new textures by selecting a region of the current camera image, which can later be used as texture and assigned to the objects (Figure 3 left). This is especially useful if the goal is to create virtual duplicates of real objects (see Figure 4). In this case it can be also convenient to immediately use the corresponding physical

texture of the object. This can be achieved by back projecting the created 3D object into the camera image and using the projected area of the camera image as texture. This can be comfortable if objects have to be created fast, because no texture cutout is required.

5.4 Generating registered 2D objects

In addition to the creation of 3D objects, the system also supports a variety of 2D tools and objects to draw or annotate the environment. Similar to basic paint programs the user can choose between pencils, brushes or a graffiti spray tool. In order to create registered 2D objects, the objects are not located in the camera space, but in the world space. Therefore the intersection point with the ground plane is determined analog to the creation of 3D objects.

5.5 Interacting with the mobile device

The creation and manipulation of the content is done directly on the mobile device. Since more and more mobile phones have no physical keyboard, all manipulations and authoring functions are controlled using a touch screen. This raises the problem how to

Figure 4: Example showing the created augmented 3D content using the system configured for large working environments, Left: Real scene as displayed in the camera view of the smartphone, Middle: Augmented scene showing the created virtual object (highlighted in green) during placement operation, Right: Final scene showing a created duplicate of an existing building augmented left to the real building.

accurately operate the device while pointing the camera to the designated spot.

We solved this problem by implementing a freeze mode. Once the user is in the position where she wants to start the authoring task she can freeze the view by pressing a button on the touch screen. This simulates a fixed position by freezing the current camera frame and allows moving the device with no implication regarding the tracking. After a successful completion of the selected task, the user can unfreeze the view.

The whole system running on the smartphone is implemented using the C/C++ based Studierstube ES as underlying framework. This allows real-time performance for the tracking as well as rendering during the content creation process. While Studierstube ES is a cross platform framework this work was mainly tested on an Asus P565 (600Mhz) and a HTC HD2 (1Ghz) both running Windows Mobile.

6   Content providing

The idea of AR 2.0 raises the question how the created content once created is published to all target devices. Assuming millions of target devices and massive amount of content, we have to select the displayed items wisely to minimize visual clutter (e. g., showing too much content on a small screen) and to preserve bandwidth (e. g., transmitting irrelevant content to the client).

To address these goals, we use a server infrastructure as backbone, which shares many techniques and features with a server hosting a Web 2.0 application. This server for hosting content for AR supports among others user accounts, tagging with meta information, geo-referencing of content with GPS coordinates, referencing content to a ground truth in 6 *Degrees of Freedom* (DOF) (marker, natural feature tracking database) and allows to query that stored content using this information. The server application is implemented using standard web frameworks such as Apache Tomcat together with a MySQL database. The content is made available via a web services as well as via a programming interface, which can be accessed using C/C++.

This system can easily be adapted to only download content of known friends (e. g., by using Facebook's friends list) or updating friends (e.g. via Twitter or Facebook) if new content is published.

We used this server to hosts more complex content types, which was created with our prototype for 3D in-situ authoring described in section 5. In this system the generated AR content and all related files are submitted to the server and stored in a zip container as this content is more complex and can consist of many files like 3D models, textures, text information and a XML markup file to describe the content and add meta information. We developed our own XML markup language schema to express all relevant information and call it *Augmented Reality Markup Language* (ARML). The resulting XML markup file is also placed in the container and later used for indexing the content and inferring the relations concerning place, time, user etc. (see Figure 5).
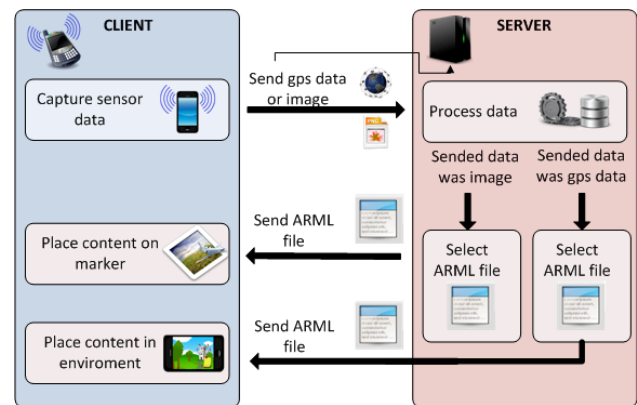

Figure 5: Illustrating the dataflow for content providing using an remote server and a client with AR capabilities.

Beside the possibility of sharing the generated content with other users over a remote content server, each user has a private inventory. This allows to place created objects into the local inventory on the device for later use. Hence the user is able to create an object (e.g. annotated with a texture of the current environment), pick it up and insert it into a scene at a different location.

7 Conclusion and Future Work

We have presented a system for mobile AR authoring, which can be used for user content creation in an AR 2.0 system. We showed how users could create content in place by utilizing two different approaches for tracking the device and combining it with simple, but powerful authoring functions.

This allows the user to create new 3D content as well as to replicate existing 3D objects, which can be used for various AR applications like phantom rendering, games, art installations and social AR applications like AR browsers. Furthermore we allow the user to sketch in AR using augmented

graffiti's allowing to color the environment or place annotations. Embedding a client-server approach to support content sharing between multiple users completes the system, which fulfills all aspect of an AR 2.0 environment.

During preliminary user tests we identified the need for an improved interface, which we try to realize in future version. While most of the users were able to handle the content creation process they stated that they have to make too many actions to create the content. Hence we'll revise the interface to decrease the numbers of buttons and user actions. One approach is to allow the picking of the object instead of switching through the objects to select the current active one.

Other future works include a user study to evaluate this system in term of usability as well as to integrate approaches for an image based automatic reconstruction in case of existing geometry. We are also working on a desktop-based interface for accessing the content server to manipulate and improve content, which was created while in place. Furthermore we are interested in cross-linking with other conventional online resources such as Wikipedia to create mash-ups of content, which is displayed using Augmented Reality as interface.

8 Acknowledgements

References

[1] R. Azuma, A Survey of Augmented Reality, In Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), 1995, pp. 355-385.

[2] Y. Baillot, D. Brown, and S. Julier, "Authoring of Physical Models Using Mobile Computers," Proceedings of the 5th IEEE International Symposium on Wearable Computers, 2001, p. 39.

[3] P. Bunnun and W.W. Mayol-Cuevas, "OutlinAR: an assisted interactive model building system with reduced computational effort," 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, Ieee, 2008, pp. 61-64.

[4] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment," Personal Technologies, vol. 1, 1997, pp. 208-217.

[5] A.v. Hengel, R. Hill, B. Ward, and A. Dick, "In situ image-based modeling," Symposium on Mixed and Augmented Reality, 2009, pp. 107-110.

[6] A. Henrysson, M. Ollila, and M. Billinghurst, "Mobile phone based AR scene assembly," ACM International Conference Proceeding Series; Vol. 154, 2005, p. 95.

[7] A. Henrysson and M. Billinghurst, "Using a mobile phone for 6 DOF mesh editing," CHINZ; Vol. 254, 2007, p. 9.

[8] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," Symposium on Mixed and Augmented Reality, 2007, pp. 1-10.

[9] B. MacIntyre, M. Gandy, S. Dow, and J.D. Bolter, "DART: a toolkit for rapid design exploration of augmented reality experiences," Symposium on User Interface Software and Technology, 2004.

[10] W. Piekarski and B. Thomas, "Tinmith-Metro: new outdoor techniques for creating city models with an augmented reality wearable computer," Proceedings Fifth International Symposium on Wearable Computers, IEEE Comput. Soc, 2001, pp. 31-38.

[11] D. Schmalstieg, T. Langlotz, and M. Billinghurst, "Augmented Reality 2.0," Dagstuhl seminar series, 2010.

[12] G. Simon, U. Henri, and P. Inria, "In-Situ 3D Sketching Using a Video Camera as an Interaction and Tracking Device," Proceeding of Eurograhics 2010, 2010.

[13] J.C. Spohrer, "Information in places," IBM SYSTEMS JOURNAL, vol. 38, 1999, pp. 602-628.

[14] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, "Real-time panoramic mapping and tracking on mobile phones," 2010 IEEE Virtual Reality Conference (VR), 2010, pp. 211-218.

[15] G. Schall, D. Wagner, G. Reitmayr, E. Taichmann, M. Wieser, D. Schmalstieg, and B. Hofmann-Wellenhof, "Global pose estimation using multi-sensor fusion for outdoor Augmented Reality," Symposium on Mixed and Augmented Reality, 2009, pp. 153-162.

[16] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose Tracking from Natural Features on Mobile Phones," Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, 2008, pp. 125-134.

[17] Y. Wang, T. Langlotz, M. Billinghurst, and T. Bell, "An Authoring Tool for Mobile Phone AR Environments," Proceedings of New Zealand Computer Science Research Student Conference 09 (NZCSRSC '09), 2009, pp. 1-4.