Towards in situ authoring of augmented reality content

Mike Wozniewski
Concordia University /
Hololabs Studio
Montréal, Québec, Canada

Paul Warne
Hololabs Studio
Montréal, Québec, Canada

ABSTRACT

We describe techniques for the authoring of 3D objects and scenes in a mobile augmented reality scenario. By capitalizing on human spatial awareness and two-handed interaction, we argue that complex modelling tasks can be effectively performed by non-expert users. We also confront the challenge of texture mapping as part of the 3D model creation pipeline, and demonstrate that the augmented reality camera view can offer a rich texturing environment, making use of real world imagery. The discussion is supported by references to the proof-of-concept application, called Farrago, which has served as a testbed for many of these theories. Ultimately, we show that not only can users create compelling 3D assets in situ, but can use the same tool for novel forms of authoring in the domains of photography, film making, and special effects.

1 INTRODUCTION

Augmented reality (AR) applications are typically divided into an authoring and presentation phase, requiring different tools, devices, and skills for each task. A typical pipeline for AR production requires a 3D designer, who uses sophisticated tools (3D Studio Max, Unity 3D, Maya) to create textured models. An interactive developer then uses an AR library or middleware (ARToolkit, Layar, Qualcomm's AR SDK) to create an application that loads the object and associates it with a fiduciary marker. The end user's role is most often that of a spectator, where he or she places a marker within the field of view, and sees the 3D form rendered on the screen. Interaction is often limited to changes of perspective, with perhaps the use of mouse clicks or touch events to trigger events.

We propose that touch-based pose manipulation, in addition with the ability to provide spatial frames of reference for interaction, can make AR a perfect tool for in situ 3D composition, while simultaneously providing a rich tool for photography and film making.

The recent support for AR on mobile devices provides an ideal platform for this type of creation. Devices easily fit into a one's hand, allowing the camera to become a tangible input device, and providing kinesthetic feedback to ultimately reduce the cognitive load required for accomplishing complex spatial tasks. After all, a human typically knows where their hands are in space without needing to look. As an example, try closing your eyes and touching your index fingers together. Then to contrast, try aligning two randomly located geometries with 3D Studio Max. In the latter case, you will likely need to switch between various orthographic camera views (front, side, top) and perform several independent translations in order to achieve the same result.

In addition to capitalizing on human spatial awareness, mobile authoring of AR content allows users to go out into the world and build creations in a particular place. It becomes much easier to get the right backdrop for a scene, because the camera view is an inherent feature of augmented reality. Moreover, real-world imagery can be used to facilitate texture creation and mapping, in situ. One can take a photo, erase the undesired parts of the image, and map it onto a geometry in a matter of moments.

Of course, there is more to the task of 3D authoring than moving around objects and applying simple textures, so it is important to find the right use cases for in situ authoring, and the type of creation that is possible without burdening the user. We focus on non-expert users, who have some experience in design and creation, or have an affinity to sandbox-style games and virtual worlds. That is, someone who may have created a simple website, posted a video on YouTube, created an avatar in Second Life, or plays games such as LittleBigPlanet and Minecraft. We are inspired by the recent surge of user-generated content coming from the {*mash-up artist / hacker / DIY / maker*} demographic, who make use of new social tools to create and share their ideas.

2 *Farrago*

As a proof of concept for in situ authoring, we have developed an iOS application called *Farrago*,[1] which provides users with a mobile tool for composing augmented reality scenes, without the need for a desktop-based authoring phase. While some AR apps support uploading 3D models (e.g., Layar, Junaio, Aurasma), there are only a few research projects [8, 5, 4] that have considered mobile creation and viewing of 3D assets directly in the AR camera view.

The name "Farrago" itself, which derives from a Latin word that means "mixture", speaks of the nature of the experience we wish to create. Users do not model in the traditional sense; rather, they combine, assemble, and arrange content into scenes that tell a story or convey a message. So in part, it is an app that creates 3D assets (which can be saved and shared), but it is also about capturing a moment in photo or video format. Thus, "authoring" with augmented reality can imply creation of works for different mediums. For the purpose of this discussion, we will primarily focus on the authoring of 3D assets, but it should be noted that these techniques can provide a fundamentally different and compelling method for the creation of 2D imagery as well.

Our research prototypes, and the official *Farrago* application published on Apple's App Store, use ARToolkit for tracking, and OpenSceneGraph for the underlying graphics engine and scene graph management.

3 MARKERS, DEVICES, AND DIVISION OF LABOUR

The authoring techniques described below assume that fiduciary markers are tracked with acceptable accuracy at arm's length. That is, the user holds the camera (mobile device) with one have and a marker in the other.

This *bimanual* (two-handed) arrangement has been shown to improve performance for complex tasks when the right considerations are made. For instance, Buxton and Myers [2] note that two hands for selection and positioning tasks greatly improve performance, while Leganchuk et al [9] note that compound (parallel) tasks can be performed more efficiently with two hands, without increasing the cognitive load imposed on the user.

So what considerations are required to provide effective bimanual interaction? The widely accepted paradigm of the *Kinematic*

[1] www.farragoapp.com

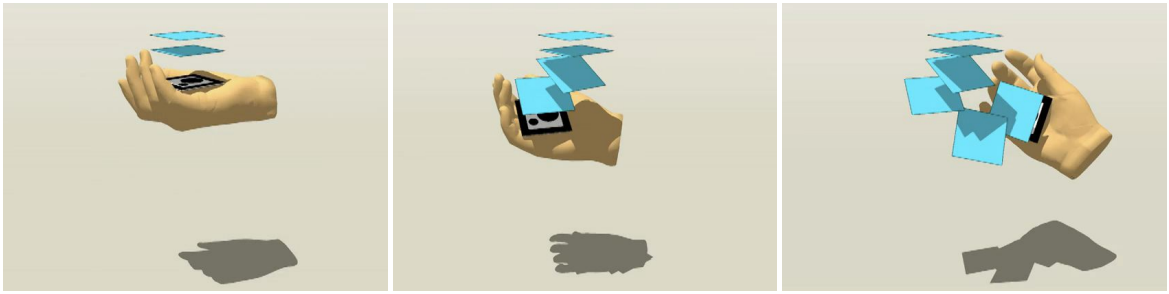

Figure 1: Illustration of creation using an *3D object brush*.

Chain Model proposed by Yves Guiard [3] suggests that tasks be designed with an asymmetry and division of labour between hands. He develops some rules that outline how two hands should be used:

- *Preferred-to-non-preferred reference:* the preferred hand finds its spatial references in the results of non-preferred hand motion.
- *Asymmetric scales of motion:* the preferred hand has higher temporal and spatial scales of motion. The non-preferred hand has more infrequent, coarse movements.
- *Non-preferred hand precedence:* the non-preferred hand moves first to set a frame of reference, then the preferred hand applies actions relative to that reference.

In an augmented reality scenario, such as that seen in Figure 2, we understand this to mean that that the non-preferred hand should hold the camera (providing the reference and moving first). Moving the marker in the preferred hand will typically involve more frequent, precise motions, and span a greater spatial scale, while adjustments of the camera position will be more coarse in nature. We will explore this idea further as we begin to look at how AR can simplify and optimize the task of in situ authoring.

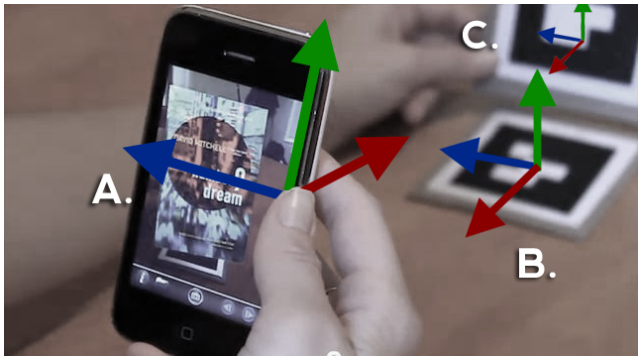

Figure 2: Bimanual interaction where the camera frame (A) is specified by one hand while an object's pose (C) is specified by a fiduciary marker in the other hand. The world coordinate system is also updated by a marker (B), providing a reference for the entire scene. (NOTE: the user is left-handed).

4 Techniques for in situ 3D modelling

Given that our aim is to support non-expert users, we start by considering the task of simplifying the pipeline for 3D creation, and describe techniques which work effectively in a mobile scenario.

First, we note that vertex-by-vertex generation of meshes is a fastidious process, and goes against the spirit of spontaneous content generation we would like to achieve. We look instead to higher-level metaphors and sketch-based paradigms. Google's *SketchUp* is an obvious example, which operates primarily by extruding 2D shapes to quickly create 3D volumes, then faces are pushed and pulled to fine-tune the geometry. Autodesk's *123D Sculpt* uses the metaphor of modelling clay, manipulated with multi-touch gestures. This produces mesh deformations that are more curved and natural in appearance. *123D Sculpt* also provides the ability to "rub" a portion of an image onto the mesh with your finger, a compelling way to solve the UVW texture mapping problem (i.e., specifying which part of an image is applied to a particular area of the mesh).

In terms of related work in mobile AR, Langlotz et al. [8] use a *SketchUp*-style technique where a 2D object footprint can be drawn on the marker plane and extruded to form a 3D primitive. Once that is done, the object may be translated, rotated or scaled by selecting an axis from a menu and gesturing on the touchscreen. Textures can be captured from the camera view by selecting rectangular areas on the marker plane (again using touch gestures), and resulting images can be mapped onto the primitives. The use of the marker plane as the site of both 3D object creation and texture extraction is compelling, yet user tests showed that too many independent actions were required. We believe that 3D manipulations are better suited to tangible bimanual gestures, where the user holds a camera in one hand and performs 3D manipulations with the other. Research in related fields has shown that this does not add significant difficulty to the interaction and in fact, speeds up spatial tasks [6, 9].

The use of bimanual gestures with mobile devices has been explored by Henrysson et al. [5],[2], who place the camera in the preferred hand to apply transformations to an object while the non-preferred hand manipulates a printed marker (controlling the global frame of reference for the scene). A big drawback of this technique is that small movements of the camera may result in amplified movements of the 3D form. Thus, it can be difficult to accurately place an object. Our arrangement of the bimanual task is the opposite, placing the camera in the non-preferred hand, and using the preferred hand to perform precise manipulations of an object.

4.1 3D object brush

An *object brush* is a 3D mesh that can be attached to a fiduciary marker and used to create copies or instances of the mesh in space. The marker is held in the user's preferred hand, providing a tangible tool for precise placement within the scene. Once an ideal pose is acquired, the user creates a copy of the object by tapping the screen with his/her thumb. Additional instances can be added, allowing the user to build complex 3D forms. Figure 1 illustrates this concept, using a simple 2D quad as the object brush, however any 3D object (with textures) may likewise be used as a brush, providing the building blocks for more complicated scenes.

While it is possible to just use the camera and one marker for this technique, we find it very useful to employ an additional marker

[2]It may be interesting to note that Henrysson et al. have also developed AR techniques for mesh deformations in the spirit of *123D Sculpt* [4].

that specifies the world reference frame (as seen in Figure 2). This helps to resolve occlusion issues, which occur when the scene fills up with objects. The user can simply move the reference frame (or camera) to the side to ensure that geometries are visible and line up correctly. If the reference frame is removed, the relative camera offset is maintained, and the effect is that the camera temporarily becomes the main reference frame. We have found from preliminary experiments, that one often wants to temporarily hide the reference frame when performing precision placement tasks. Furthermore, there results a constant play between setting a view, managing occlusions, and placing object instances. Which leads us to realize the impressive nature of this paradigm: that these manipulations can be accomplished in parallel, while managing three coordinate systems.

At the time of writing this, the object brush technique is still being prototyped, and has not yet deployed in the official App Store version of Farrago. Upcoming user tests will be used to fine tune this technique and should show good performance for spatial arrangement tasks.

4.2 Template objects

We have discussed positioning and creating instances of objects, but have not yet mentioned how the initial objects may be created. The quick answer is that we make use of a library of primitives, or rather *template objects* which have predefined shapes and texture mappings.

Figure 3 shows a small subset of almost 50 template objects that have been created for *Farrago*. The library ranges from simple cubes and spheres, to more complicated meshes that approximate entities such as faces and waving flags. UVW texture mapping is partially done in advance, removing one of the more complicated steps in the traditional 3D modelling pipeline, yet users are free to choose their own images and pan/zoom to fit them to the template.

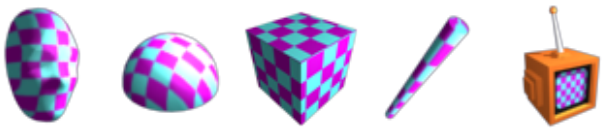

Figure 3: Examples of 3D template objects.

Looking more closely at the items in Figure 3, we see examples like template (e), which maps a texture onto a screen-like area of a more complex geometry, while template (a) provides a mesh deformed to simulate a human face. The collection of templates that we have provided should be able to produce a diverse assortment of creations from real world imagery in a matter of moments. Of course, there is a trade-off here, because custom mappings (e.g., different textures on size sides of a cube) cannot be accomplished by the user alone. However, it is always possible for the user to group geometries (e.g., six textured quads to form a cube) using an object brush to achieve the same result.

Farrago Object Creator: In contrast to the work by Langlotz et al. [8], we do not capture textures in real time from the AR camera view, in part due to the low resolution imposed by the real-time tracking algorithm. Instead, we provide a separate editor for image manipulation which provides users with the choice of taking a high-resolution photo of the current scene or choosing existing material from his or her device.

Once an image is chosen, the Object Creator is displayed and the user can construct a primitive. A template is chosen from a menu, and depending on the selection, a semi-transparent overlay will identify the part of the image to be mapped onto the geometry. Figure 4 shows an example of this process. A diamond-shaped template object has been selected, so the user must fit the image

Figure 4: Screenshot of the Farrago Object Creator, showing how textures are mapped onto template objects.

into the diamond shape that will be mapped onto geometry. Touch gestures are used to pan and zoom their images under this overlay.

Behind the scenes, a template object is like any other simple model, but contains a specially named placeholder texture. The user is effectively choosing a replacement texture, and saving a copy of the template to his or her library. We have created a set of roughly 50 templates, which is by no means exhaustive, but seems to provide the necessary building blocks for many types of scenes.

An additional, not to be overlooked feature of the Object Creator is the ability to add an alpha mask to the image using a touch-based erase/reveal tool. This feature can be used to create holes in a mesh, or "cut out" objects from photos taken in the current environment. Figure 5 shows how elements from a graffiti wall can become elements in the AR scene, fulfilling our goal of in situ texture mapping.

Figure 5: Example of in situ texture mapping from captured camera images. Note that the cartoon head was captured from the graffiti wall (top-right of image).

5 Touch Interface

While fiduciary markers provide the ability to specify a 6-DOF pose for an object a tangible manner, there are reasons to explore the touchscreen for these operations. One reason is that AR tracking technology is not perfectly stable and sometime fails to accurately report the pose of the marker every frame. Jjittery pose updates, false positives, and poor recognition under certain lighting conditions may warrant the use of the touchscreen as an alternative (or additional) means for arrangement.

There are several techniques, such as RNT [7], the Z-technique [10], etc., or each transformation (translation, rotation, scaling) can be performed independently. Due to fact that our task typically involves the 'lining up' geometries in space, we tend to use the latter approach of independent transformations. One-finger touches are used for planar translations relative to the camera's image plane, while two-finger touches have four possible modes chosen from a menu: X, Y, Z rotation or scaling. A special three-finger swipe is used to adjust the distance of the object relative to the camera.

The question of which object should be affected by the gestures is a challenge. The typical approach of "3D picking" projects a touch coordinate into the scene through the camera's perspective view, returns a list of intersected objects, and typically we choose the first (closest) object as the intended target. This is problematic, since it can be difficult to choose occluded objects, but more importantly, it results in the fact that the user's finger hides the geometry from view during interaction. This can perhaps be solved by two finger techniques [1], but we instead choose our objects using a menu, allowing the user to both select occluded objects, and to perform touch gestures anywhere on the screen.

5.1 Marker-relative transformations

It should be noted that the touch techniques mentioned above effectively replace the use of fiduciary markers. As soon as a marker returns to view, the object will snap to the marker's coordinate system and all transformations (except for scaling, which is actually a local transform) will be lost. However, a more compelling use these gestures is actually when they are used with a marker in view.

In this case, touch-based transformations become relative to the marker's coordinate system rather than the world. This allows the user to offset an object from the marker's local coordinate system. This may seem counter productive, but it allows for some interesting types of creation. For instance, by offsetting an object on the marker plane, the user can spin the marker around while 'tapping out' object instances to form a radial pattern around the marker.

Farrago also lets you load several objects onto one marker. An example of this occurs when the user wants to build a figurine by choosing independent body and head models from the library. A marker-relative transformation is required in order to properly place the head on top of the body.

6 Authoring of Photo & Video Content

To summarize, we have described AR tools and techniques for the creation of 3D assets and the arrangement of these elements into complex scenes. We have however ignored the fact that AR may likewise be used to author 2D content. The really compelling aspect of this technology is the fact that the camera image is constantly streaming real world video, and situating virtual content in reality. Taking photos and videos of this combination allows users to easily create augmentations of real scenarios. As such, it makes for a great authoring tool for photography and novice film making, where special effects can be added in real time rather than in a post production process (e.g., Figure 6). Even for professional cinematographers and movie makers, this approach can help in rapid prototyping. Location scouts can go to a physical location, import some models and simulate how the real special effects might look.

The fact that these tools are available on the spot, in situ, also provides an interesting arena of experimentation. By shifting the authoring phase of away from the desktop computer, creators are free to play with the boundary between real and virtual. Creative works may become more situated, context aware, and tell a more compelling narrative than those created in the confines of the studio.

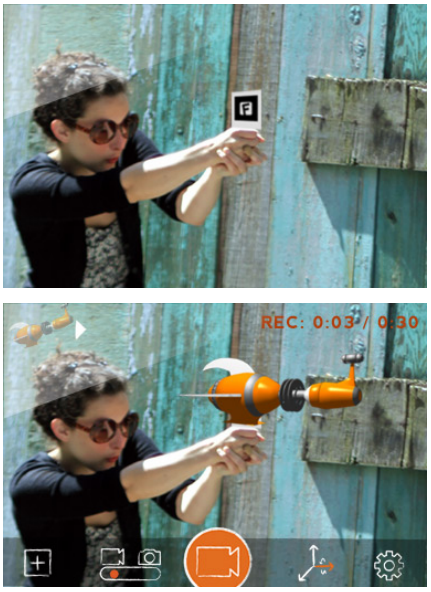

Figure 6: Example of film making and adding simple special effects.

References

[1] H. Benko, A. D. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 1263–1272, New York, NY, USA, 2006. ACM.

[2] W. Buxton and B. Myers. A study in two-handed input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, volume 27, pages 321–326, 1986.

[3] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a mode. *Journal of Motor Behavior*, 19(4):486–517, 1987.

[4] A. Henrysson and M. Billinghurst. Using a Mobile Phone for 6 DOF Mesh Editing. In *8th Annual Conference of the NZ ACM Special Interest Group on Human-Computer Interaction (CHINZ 2007)*, pages 9–16. ACM, 2007.

[5] A. Henrysson, M. Ollila, and M. Billinghurst. Mobile Phone Based AR Scene Assembly. In *4th International Conference on Mobile and Ubiquitous Multimedia (MUM 2005)*, pages 95–102. ACM, 2005.

[6] K. Hinckley, R. Pausch, D. Proffitt, and N. Kassell. Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(3):260–302, 1998.

[7] R. Kruger, S. Carpendale, S. D. Scott, and A. Tang. Fluid integration of rotation and translation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, pages 601–610, New York, NY, USA, 2005. ACM.

[8] T. Langlotz, S. Mooslechner, S. Zollmann, C. Degendorfer, G. Reitmayr, and D. Schmalstieg. Sketching up the world: in situ authoring for mobile augmented reality. *Personal and Ubiquitous Computing*, pages 1–8, 2011. 10.1007/s00779-011-0430-0.

[9] A. Leganchuk, S. Zhai, and W. Buxton. Manual and cognitive benefits of two-handed input: An experimental study. *ACM Transactions on Computer-Human Interaction*, 5(4):326–359, 1998.

[10] A. Martinet, G. Casiez, and L. Grisoni. The design and evaluation of 3D positioning techniques for multi-touch displays. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 115–118, Mar. 2010.