Marking up a World: Physical Markup for Virtual Content Creation

Eleanor G. Rieffel
FX Palo Alto Laboratory
3400 Hillview Ave.
Palo Alto, California
rieffel@fxpal.com

Sagar Gattepally
FX Palo Alto Laboratory
3400 Hillview Ave.
Palo Alto, California
sagar@fxpal.com

Don Kimber
FX Palo Alto Laboratory
3400 Hillview Ave.
Palo Alto, California
kimber@fxpal.com

Jun Shingu
FX Palo Alto Laboratory
3400 Hillview Ave.
Palo Alto, California
jun@fxpal.com

Jim Vaughan
FX Palo Alto Laboratory
3400 Hillview Ave.
Palo Alto, California
jimv@fxpal.com

John Doherty
FX Palo Alto Laboratory
3400 Hillview Ave.
Palo Alto, California
doherty@fxpal.com

ABSTRACT

The Pantheia system enables users to create virtual models by *'marking up'* the real world with pre-printed markers. The markers have predefined meanings that guide the system as it creates models. Pantheia takes as input user captured images or video of the marked up space. This video illustrates the workings of the system and shows it being used to create three models, one of a cabinet, one of a lab, and one of a conference room. As part of the Pantheia system, we also developed a 3D viewer that spatially integrates a model with images of the model.

Categories and Subject Descriptors

J.2 [**Computer Applications**]: Physical Sciences and Enginering; J.5 [**Computer Applications**]: Arts and Humanities

General Terms

algorithms, design

Keywords

image-based modeling, virtual environments

1. INTRODUCTION

The creation of virtual models is complex and cumbersome, even when the virtual model is based on a physical one. FXPAL's Pantheia system [6, 10] enables users to create virtual models by *'marking up'* a physical space with pre-printed visual markers and then taking images or video of the marked up space. The markers have associated meanings taken from a markup language that guides the system in creating models even from a relatively sparse set of markers.

Current state of the art model creation is done by artists who use measurement, photography, and sophisticated graphics tools to aid them in creating models. A number of research groups work on non-marker-based methods for constructing models from sets of images [8, 9, 5]. Their work advances progress toward the challenging goal of deducing geometric structure purely from image features. Instead, we make the problem simpler by placing easily detectable markers with associated meanings that greatly simplify the geometric deduction. Marker-based approaches have other advantages. They enable users to specify which parts of the scene should be included in the model. In this way, Pantheia handles clutter removal and certain occlusion issues easily since it renders what the markers indicate rather than what is seen. In addition, markers enable the specification of deviations from the physical scene on which the model is based; markers placed in the same relative positions on a rough cardboard construction, without doors or drawers, would create the same model of the cabinet we obtained by marking up an actual cabinet.

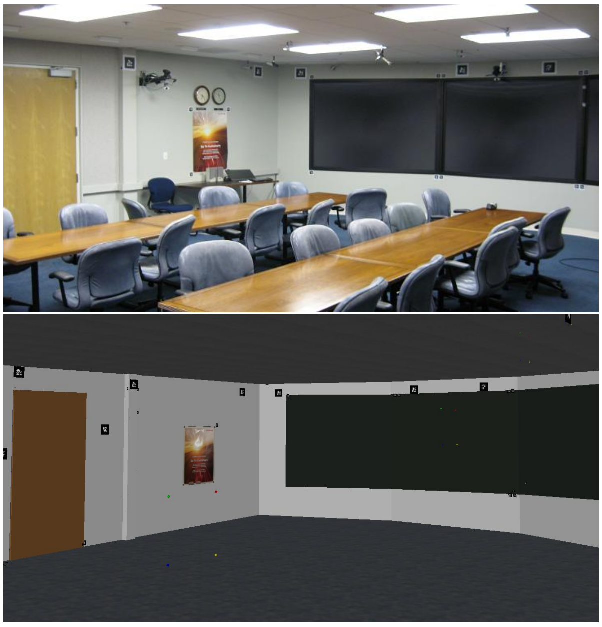

Figure 1: Photograph of FXPAL's main conference room with markup, and Pantheia generated model.

MM'09, October 19–24, 2009, Beijing, China.
ACM 978-1-60558-608-3/09/10.

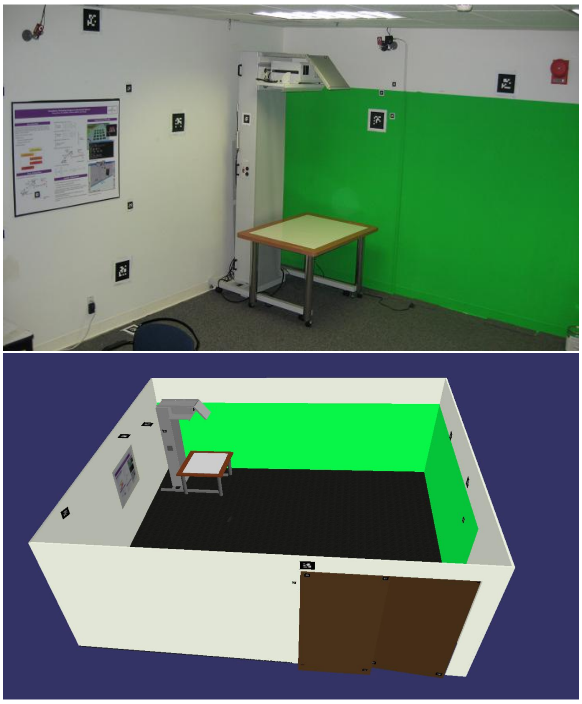

Figure 2: Photograph and 3D model of our lab.

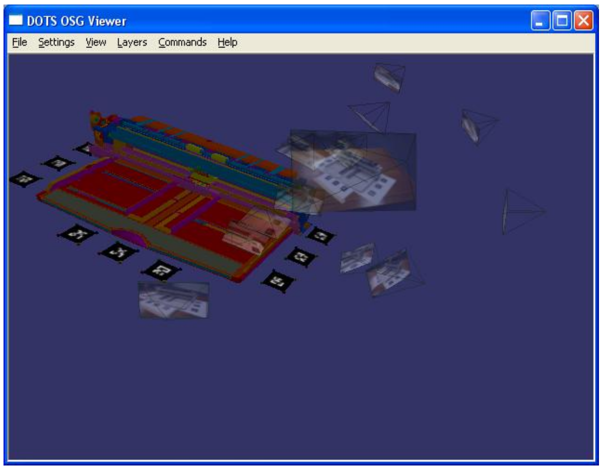

Figure 3: Model of a machine component and spatially placed thumbnails of all images showing one of the markers.

2. OVERVIEW OF THE PANTHEIA SYSTEM

To make use of Pantheia users must *'mark up'* the real world with uniquely identifiable visual markers and collect video or images of the marked up space. The markers have predefined meanings taken from a markup language [6, 10] that includes the following elements: planar (plane, wall, ceiling, floor, door); shape (parametric shape, extrusion); modifier (line, corner); appearance (color group, color sampler, texture group, texture sampler, image extractor); submodel insertion (object insertion, definition, reference, replication); dynamic (drawer, door, color switches). Typically markers of various shapes, sizes, and meanings, are placed around the space.

Pantheia takes as input the set of captured images, identifies markers in each image, and determines the relative pose of the marker to camera for each marker in each image. If the pose of a marker in the world is known then the position of the camera can be determined. Conversely, if the pose of a camera is known then the pose of any marker in that image can be computed. From a set of images that meets a few simple conditions, the pose of every marker and every image can be estimated. Pantheia obtains good estimates using ARToolKit [1] together with the sparse bundle adjustment package SBA [7] that globally optimizes the pose estimates for all markers and images. The marker poses and the meanings of the markers are then interpreted by a markup handler that creates the model. More details about the system can be found in [6, 10].

3. PANTHEIA DETECTION, ESTIMATION, AND MARKUP HANDLING

The video begins by showing views of models we made using Pantheia together with an image of the marked up physical space or object on which these models were based. The processes by which Pantheia creates a model are illustrated, using our lab as the example. Upon input of user captured images of a marked up space, Pantheia begins the process of marker detection and identification, pose estimation, and markup handling. If the user has a preferred coordinate system, such as the coordinate system used for a pre-existing model's, the user specifies the pose of one of the markers in this coordinate system. The video shows such a marker for our lab in relation to the rest of FXPAL's floor plan. The camera pose of any image containing this marker can then be estimated. Given the camera pose of an image, the pose of any marker detected in that image can be estimated. This process is iteratively repeated all possible marker and image poses have been determined. The meanings associated with the markers whose pose have been determined are then interpreted.

When sufficient information become available to determine an aspect of the model, that aspect becomes visible in the model. We first see the floor appear, then a wall, then the ceiling, and then another wall. As more aspects of the model are determined, previous parts of the model may be modified. For example, once the ceiling is determined the height of any already existing walls are adjusted. Similarly the length of a wall is determined by its intersection with other walls. From the door marker poses and associated meanings, the location of each door and its hinge are determined. A corner marker is used to define a subregion of a wall. The color sampler marker tells the system to sample the color inside the square region it outlines in each image containing the sampler marker and to use the average of those samples as the color for the region in the model. A associated marker in another region can indicate that that region also should be given that color. Markers can also indicate where a pre-existing model should be placed such as the piece of machinery in our lab.

4. PANTHEIA VIEWER

The final VRML model produced by Pantheia can be viewed in any VRML browser or in the Pantheia viewer we built using OpenSceneGraph [2]. The Pantheia viewer supports integrated 3D views of a model and its images. The model does not have to be one generated by Pantheia. In the video segment, a pre-existing CAD model for a machine part is shown together with images we took of the actual part. We surrounded the part with markers so that the pose of the images taken of the model could be determined. Knowing the pose of the images enables the system to place thumbnails of the images in appropriate positions around the model. The interface supports various views combining images and the model:

- **Cluster of images containing a marker:** If a user clicks on a marker, thumbnails of all images containing that marker are shown, positioned correctly with respect to the model. If the user clicks on one of these images, the view changes to show the model from the perspective of the pose of the camera when the image was taken. A slider enables the user to view a continuum of alpha blends of the image with the model.

- **Sequential tour of all images of the model:** The user can select an image tour in which all images are aligned with the model. The user can adjust the alpha value for the blend used by the tour. Furthermore, the user can choose between a mode with fixed view of the model, in which skewed images are shown aligned to the model, and a mode in which the view changes to that of the camera pose for each image.

5. INTERACTIVE COMPONENTS

Pantheia's marker based approach enables users to specify the behavior of interactive, dynamic components of a scene even if all of the images are of a static scene in one configuration. For example, drawer and door markers indicate the presence of hinge joints and sliders. Color change markers indicate where color control buttons should be placed in the model. These buttons enable a user to change the color of an object by clicking on the switch.

Our models can be viewed and manipulated in any VRML browser. We have also been able to export our models to Google SketchUp [4]. The interactive components work with the SketchyPhysics plugin which is built on top of a physics engine. In the VRML browser, the only interaction with a door comes from clicking on its knob, but in SketchyPhysics [3], users can partially open doors by pulling on them.

6. CONCLUSIONS

From a set of images of a marked up object or space, FXPAL's Pantheia system creates a model by detecting the markers in the images, determining their identity and pose, and using the associated marker meanings. Users may also specify interactive components using the markers. Pantheia renders the resulting model in such a way that end users can explore the model and interact with its dynamic elements. The Pantheia viewer provides integrated 3D views of a model and its images.

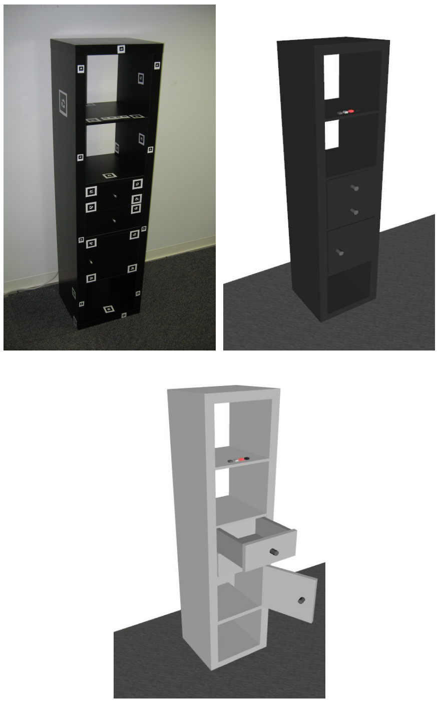

Figure 4: Pantheia generated model of an IKEA Bookcase cabinet. Users may interact with the model in a VRML browser by clicking on the drawers or door or clicking on the color control buttons.

7. REFERENCES

[1] ARToolkit. http://www.hitl.washington.edu/artoolkit/.

[2] OpenSceneGraph. http://OpenSceneGraph.org/.

[3] Physics plugin for Sketchup. http://code.google.com/p/sketchyphysics/.

[4] Sketchup. http://sketchup.google.com/.

[5] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Proceedings of CVPR'00*, 2000.

[6] D. Kimber, C. Chen, E. Rieffel, J. Shingu, and J. Vaughan. Marking up a world: Visual markup for creating and manipulating virtual models. In *To appear in the Proceedings of Immerscom09*, 2009.

[7] M. Lourakis and A. Argyros. sba: A generic sparse bundle adjustment C/C++ package based on the Levenberg-Marquardt algorithm. http://www.ics.forth.gr/ lourakis/sba/.

[8] M. Pollefeys. Visual 3D modeling from images. http://www.cs.unc.edu/ marc/tutorial.

[9] M. Pollefeys and L. V. Gool. Visual modeling: from images to images. *Journal of Visualization and Computer Animation*, 13:199–209, 2002.

[10] E. Rieffel, D. Kimber, J. Vaughan, S. Gattepally, and J. Shingu. Interactive models from images of a static scene. In *CGVR 2009, to appear*, 2009.