In Situ Image-based Modeling

Anton van den Hengel* Rhys Hill† Ben Ward‡ Anthony Dick§

School of Computer Science, University of Adelaide, Australia

ABSTRACT

We present an interactive image-based modelling method for generating 3D models within an augmented reality system. Applying real time camera tracking, and high-level automated image analysis, enables more powerful modelling interactions than have previously been possible. The result is an immersive modelling process which generates accurate three dimensional models of real objects efficiently and effectively. In demonstrating the modelling process on a range of indoor and outdoor scenes, we show the flexibility it offers in enabling augmented reality applications in previously unseen environments.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene Analysis—Shape

1 INTRODUCTION

This paper describes an in situ image-based modelling method for augmented reality, called Jiim (as Jiim is In-situ Image-based Modelling). The method uses information gained through automated analysis of video to empower an interactive 3D modelling process. The result is a flexible and efficient method for creating accurate 3D models of real objects in the scene. These models can be used within the AR system itself to enable real and synthetic objects to interact convincingly, or for non-AR purposes such as importing into Google Earth. The creation of the models and their use in AR can be interleaved, allowing "on demand" creation of the minimal 3D structure that is necessary for a particular application.

Jiim has benefits for both image based modelling and AR. The image-based modelling process benefits from the fact that the synthetic model and the real object can be compared in-situ while modelling is being carried out. This allows the user to see which parts of the model require further modelling, and to select camera viewpoints from which to capture the necessary footage.

The AR process benefits because Jiim provides an efficient means by which to inform the AR application about world geometry without the need for pre-existing models. The user may thus take the AR system into an unknown environment and quickly generate a 3D mesh representing the shape of the scene.This mesh can then be used to calculate the interaction between real and synthetic geometry. In Figure 1, for example, a user races a synthetic model of a car around a real environment, bumping into and jumping from real objects in the scene, after having created a model of the scene within the system itself (shown in Figure 2).

1.1 Image-based modelling

Image-based modelling is the process of creating a 3D model of an object on the basis of a set of images or a video sequence. Many approaches have been developed, from the fully automatic to the largely manual. For example, Photomodeler [6] and Facade [17] require the user to interactively specify shapes in the scene by marking corresponding primitives in multiple images. Based on image markup, they estimate the parameters of the cameras that took the images, and thence the 3D shape and texture of the scene.

More automated alternatives are made possible by using *camera tracking* software to estimate camera parameters and the 3D location of a sparse set of feature points. For example, Gallup et al. [7] use a non-interactive plane-sweep approach to model a scene, based on automatically estimated camera locations. This approach runs in real-time due to a GPU implementation, but not live.

In this paper, the image based modelling approach is based on VideoTrace [18], an interactive system by which a user can generate a 3D model of an object by simply tracing over it in an image. However, VideoTrace requires initialisation using offline camera tracking software. This can be a significant limitation in practice as it separates the steps of data capture and modelling, leading to several iterations of each in order to obtain video that can be verified to adequately cover the object.

Figure 1: A screen capture from Jiim showing a synthetic car model leaving the end a real ramp, and shadowing both the ramp and a toy ambulance. The geometry of the table, ramp, and ambulance required were modelled within Jiim in under a minute.

Figure 2: Part of the process of modelling the ambulance visible in Figure 1. One side of the ambulance has been traced out as a flat polygon, which can then be extruded.

1.2 Real time camera tracking

One way to improve the integration of data capture and modelling is to use camera tracking software that operates in real time. Methods for real-time camera tracking have evolved from fiducial marker-

*e-mail:Anton.vandenHengel@adelaide.edu.au
†e-mail:rhys.hill@adelaide.edu.au
‡e-mail:benjamin.ward@adelaide.edu.au
§e-mail:anthony.dick@adelaide.edu.au

IEEE International Symposium on Mixed and Augmented Reality 2009
Science and Technology Proceedings
19 -22 October, Orlando, Florida, USA
978-1-4244-5419-8/09/$25.00 ©2009 IEEE

based approaches, to those based on simultaneous localisation and mapping (SLAM) which are marker-less and require neither additional hardware nor a-priori knowledge of the camera or environment.

The MonoSLAM system [5] showed that SLAMcan recover, in real-time, the path of a single camera and a set of sparse 3D points (called a map) which describes the shape of the scene. A key limitation of MonoSLAM and similar systems is the sparsity of the 3D map typically maintained by these approaches, whose primary goal is to estimate camera pose relative to selected keypoints in the scene rather than to obtain a complete model of the scene's shape. The PTAM system of Klein and Murray [11], which combines real-time camera tracking with incremental bundle-adjustment, is able to build far denser maps containing over 10,000 point features. It does this by decoupling the camera tracking from map estimation, updating the map estimate using bundle adjustment while in parallel updating camera state using a faster, frame-rate process. By applying PTAM to the live video, we obtain an estimate of the current camera, relative to a fixed world coordinate system, and a map of 3D scene point locations which is dense enough to form the basis for interactive image based modelling software.

2 Modelling for Augmented Reality

The interaction between real objects and synthetic content is fundamental to AR. Many forms of interaction require a 3D model of the shape of the real objects involved. For example, to render a scene that combines real and synthetic objects, it is vital to know how they occlude each other. Klein et al in [10], tackle this problem for a tablet-based AR system, but require a pre-existing CAD model of any object in the scene if it is to occlude synthetic geometry.

The construction of a model which accurately reflects the shape of even a moderately complex environment can be a time consuming task. Another approach is to automatically estimate a simple scene model. Chekhlov et al.[4] and Klein et al.[11], for instance, detect planes in the scene in order to calculate how to render characters as they traverse the real environment. Although focussed on SLAM rather than AR, Castle et al. [3] show how planar objects can be recognised in real-time and added to a SLAM map as a single primitive rather than a collection of points. This technique is extended by Gee et al. [8] who show how planar areas can be identified within a scene without requiring prior knowledge of these objects. However, the goal of these systems is not to model the scene, but rather to simplify the map by collapsing groups of point primitives to a smaller number of higher level shapes.

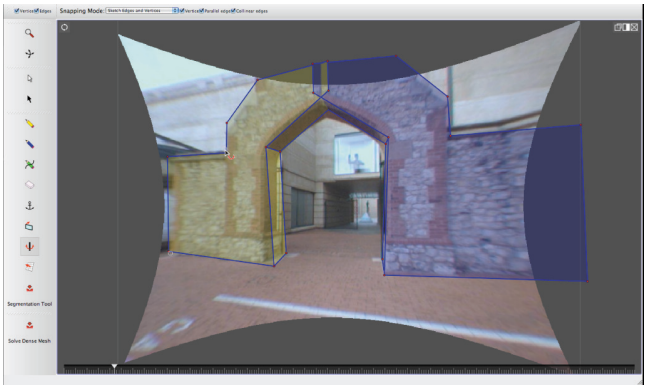

Figure 3: Part of the modelling process for the archway in Figure 6. A polygonal mesh is traced out over the corresponding structure in the image, and its 3D location is estimated using available image data. Each image is automatically undistorted.

Lepetit and Berger [14] present an approach to modelling general scene geometry which requires the user only to specify an object of interest in two (static) views. The authors make use of stereo geometry to estimate the motion of the camera and its uncertainty. The user-specified boundary is then projected into the current view and refined in order to inform the occlusion rendering process.This method produces good results and does not require a pre-existing model, but it does not run in real-time. Mooser et al. [15] have built upon Lepetit and Berger's approach but use optical flow to transfer an object boundary into a new image. The aim of this is to allow annotation of real objects rather than to estimate occlusion, Although it runs in real time it is entirely 2D and does not estimate a 3D model of the object.

2.1 Immersive modelling

Immersive modelling is a process whereby models are constructed using the Virtual Reality (VR) or AR system within which they will be used [12]. Many such modelling systems exist, and the advantages of the immersive approach have been well documented (see [12] for a survey). However, the modelling facilities in these systems are typically not designed to create models that accurately represent objects in the world, and using them for this purpose can be somewhat laborious. Examples include Piekarski et al.[16] which proposes a 3D constructive solid geometry approach to the construction of models within the Tinmith AR system, and Baillot et al. [1] a more CAD-like interface. Of note also is [13] which uses a contact probe to model a surface within an AR system. None of these systems perform any analysis of the image data, meaning that the modeller must fully specify all aspects of each object. Bunnun et al. in [2] propose a SLAM based and hence image assisted modelling process using a camera attached to a mouse, but this requires that each vertex in the model is individually specified in multiple images. Kim et al. in [9] describe an 'online 3D modelling' approach which uses satellite images to model outdoor structures for their AR system.

The distinction between these systems and Jiim is that Jiim facilitates user-assisted generation of accurate synthetic models of real objects based on analysis of video. Using PTAM and image based modelling methods, information is extracted automatically from the video which reduces the number of interactions required to construct a model which accurately reflects the shape of the real geometry.

2.2 Immersive image based modelling

The fact that the frame of reference for the modelling process and the use of the model are the same eliminates the potential for misalignment. Jiim thus uses no position information other than that recovered through the SLAM process. In-situ modelling also allows the user to make direct comparisons between the real geometry and its synthetic counterpart, greatly simplifying the verification process.

Another advantage of integrating the video capture and modelling processes is that it allows the user to identify instantly any of the image data required to generate the model which is missing, and capture it. This has the effect of ensuring that the user returns from the scene with a complete model, rather than a subset of the data required to create one.

3 Using Jiim

Jiim currently runs on a MacBook Pro (2.53GHz Intel Core 2 Duo, Nvidia 9600M GT) and a Unibrain Fire-i camera. No other sensors are used. Other interfaces could equally be used, such as a tablet-based computer with the camera attached or a head mounted display. The camera used has a wide-angle lens which, while highly advantageous for single camera SLAM, causes significant radial distortion in captured images. As many image-based modelling interactions involve specifying straight lines on images, undistorted versions of the images are used for modelling.

3.1 The Jiim process

The modelling interactions in Jiim are a superset of those of VideoTrace [18]. In its original form, VideoTrace supports a number of interactions, including:

- *Tracing polygons*, the primary modelling process whereby the user specifies the boundary of a polygon in an image and Jiim estimates its location and orientation in space on the basis of information gained through anaylsis of the image set;
- *Extrusion*, dragging a polygon creates a 3D volume from a 2D polygon;
- *Reflection*, specifying a mirror plane allow the replication of geometry and texture from one side of an object to another.

After each interaction, the model is immediately updated based on the specified shape and also by fitting the shape to the image data. This is done by a combination of cues, measuring the similarity of the projection of the shape into each image, the fit to image gradient data, and the fit to local 3D feature points estimated by the offline camera tracker. More details are available in [18].

Although the model is updated interactively, VideoTrace uses a batch optimisation procedure to initialise its processes for fitting sketched geometry to the underlying 3D point cloud supplied by the camera tracker. This means that all video data must be captured before modelling can begin. By contrast, Jiim is designed to build and update models from segments of video data as they are captured. To enable this, estimates of camera parameters and 3D point locations are calculated in real time using PTAM in place of an offline camera tracker. In addition, PTAM maintains an estimate of a dominant plane in the scene. This is used by Jiim as a ground plane, whose normal defines the vertical direction in the world.

Like VideoTrace, Jiim requires accurate user input to delineate polygons in an image. This would be difficult to achieve while performing the manipulations required to film the object. The Jiim interface therefore interleaves the capture and modelling operations. The user acquires footage in *capture mode*, and models in *model mode*. When in capture mode, video data is displayed and stored, along with camera parameters and 3D point locations estimated by PTAM. The use of PTAM allows diverse AR based applications to be run while in this mode, as shown in Section 4. When in model mode, previously captured frames of video are displayed and can be sketched on to create or update 3D models of objects they depict, as shown in Figure 3 for example. Upon switching from model mode to capture mode, the user sees the latest version of the model overlaid on the live footage. The two modes are thus tightly integrated within Jiim, but with interfaces which reflect their different purposes.

Jiim introduces a number of interactions designed to support immersive modelling:

- *Plane snapping* allows the user to specify that new geometry should be confined to the same plane as existing polygons or edges, which is useful for constraining geometry to rest on the ground plane;
- *Texturing*, the user can specify which images are to be used in order to calculate textures for polygons (individually);
- *Texture painting*, to remove occluding foreground objects from the texture, the user may paint on polygons with pixels from unoccluded images.

The disadvantage of the alternating modelling process is that PTAM needs to re-localise on every transition from modelling to capture mode. The modelling process typically involves holding the platform at an angle that is convenient for modelling, but unsuitable for PTAM. Re-localisation is thus typically required even when PTAM runs continuously.

The Jiim process is immersive and in-situ in that modelling and AR are carried out within the same software, and coordinate system, and the user can update the model from within the AR system at any time. The interactions are mouse / pen based rather than being carried out through the camera itself, and are informed by both PTAM and automated analysis of the video as it is captured. The fact that user is able to select any frame of the video upon which to model, rather than only the frame most recently captured, may be seen as limiting the degree of immersion. Recall, however, that the only video used in this process is that captured by the AR system itself. Limiting the modelling process such that it could only access the most recently captured frame would significantly limit the practicality of the system as the user would need to carry out the interactions while holding the camera (and possibly attached computer) in precisely the position required to achieve the desired view.

Figure 4: The car model partly occluded by a (real) guitar on the basis of a model generated using Jiim.

4 Results

Jiim produces low polygon count texture-mapped models which are of suitable quality for a variety of purposes, including importing into Google Earth as shown in Figure 5. In order to test the suitability of the models for AR purposes three AR applications have been implemented. In the first the user throws balls into the scene. The balls emerge from the current location of the cursor, and interact with modelled geometry. Figure 6 shows a ball being hurled, and then bouncing off a part of the modelled scene geometry. The balls are synthetic elements, but through the immersive image-based modelling process presented, are able to bounce off, shadow and be occluded by real geometry, thus demonstrating the interaction of real and synthetic geometry. In all applications, to better integrate synthetic objects with the live video, a shader is used to apply Gaussian blur and slight desaturation to the rendered geometry.

Figure 5: A Jiim model imported into Google Earth.

The second demonstration application is a racing game. The user generates a model of the environment around which they wish to race using Jiim and then is able to race a model of a car through the space. The car model bounces off real obstacles, and jumps over real ramps as the user drives. A frame from the racing game is shown in Figure 1.

Finally, a game titled 'Bowl-A-Mole' is shown in Figure 7. In this game, the users attempts to bowl moles over using balls

Figure 6: A frame from the Ball Game showing (synthetic) balls bouncing off, casting shadows upon, and occluded by the (real) geometry. See video accompanying submission.

Figure 7: A frame from the Bowl-A-Mole game in which the user throws balls at moles emerging from modelled geometry. Note the mole on the left is occluded by the table. See video for more results.

launched from the camera position. The moles are located at random positions in 3D upon the model the user has created.

4.1 Limitations and further work

The quality of the result produced by Jiim is dependent upon the quality of the camera tracking and point location information provide by PTAM. Although effective, PTAM is not perfect, and often loses track, particularly in outdoor environments, as it was not intended for this purpose. PTAM also has a number of limitations, such as requiring a wide–angle camera and losing track under certain circumstances [11]. SLAM technology is developing, however, and Jiim can easily be adapted to another SLAM system. The modelling interactions require a level of accuracy which demands the user's full attention. A more robust set of interactions may be easier to use if a significant amount of modelling was to be carried out. The bottom right of Figure 6 shows a failure case in which a portion of the white synthetic ball is not correctly occluded by a real car due to the front of the car being omitted from the model.

We aim to incorporate a GPS into the SLAM system in order to facilitate geo-referencing of models, and add stability to the SLAM process. Following that, we aim to incorporate the capacity to import and edit models from other sources.

5 Conclusion

Jiim combines interactive image-based modelling empowered by PTAM and automated image analysis, with the benefits of immersive AR. The result is an efficient and effective method for generating accurate 3D texture–mapped models of real objects within an AR system. The benefits this offers in applying AR in situations where pre-existing models are not available have been shown, along with the quality of the models which may be produced for other purposes.

References

[1] Y. Baillot, D. Brown, and S. Julier. Authoring of physical models using mobile computers. In *Proc. 5th IEEE International Symposium on Wearable Computers*, 2001.

[2] P. Bunnun and W. Mayol-Cuevas. Outlinar: an assisted interactive model building system with reduced computational effort. In *7th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, September 2008.

[3] R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray. Video-rate recognition and localization for wearable cameras. In *Proc. British Machine Vision Conference (BMVC'07, Warwick)*, 2007.

[4] D. Chekhlov, A. P. Gee, A. Calway, and W. Mayol-Cuevas. Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In *ISMAR '07: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–4, Washington, DC, USA, 2007. IEEE Computer Society.

[5] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Tranactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[6] Eos Systems. Photomodeler: A commercial photogrammetry product http://www.photomodeler.com, 2005.

[7] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[8] A. P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas. Discovering higher level structure in visual SLAM. *IEEE Transactions on Robotics*, 24(5):980–990, Oct 2008.

[9] S. Kim, S. DiVerdi, J. S. Chang, T. Kang, R. Iltis, and T. Höllerer. Implicit 3d modeling and tracking for anywhere augmentation. In *Proc. ACM symposium on Virtual reality software and technology*, 2007.

[10] G. Klein and T. Drummond. Sensor fusion and occlusion refinement for tablet-based AR. In *Proc. Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2004.

[11] G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. International Symposium on Mixed and Augmented Reality (ISMAR'07, Nara)*, 2007.

[12] G. A. Lee, G. J. Kim, and M. Billinghurst. Immersive authoring: What you experience is what you get (wyxiwyg). *Commun. ACM*, 48(7):76–81, 2005.

[13] J. Lee, G. Hirota, and A. State. Modeling real objects using video see-through augmented reality. *Presence: Teleoper. Virtual Environ.*, 11(2):144–157, 2002.

[14] V. Lepetit and M.-O. Berger. A semi automatic method for resolving occlusions in augmented reality. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR '00, Hilton Head Island)*, 2000.

[15] J. Mooser, S. You, and U. Neumann. Real-time object tracking for augmented reality combining graph cuts and optical flow. In *Proc. International Symposium on Mixed and Augmented Reality*, 2007.

[16] W. Piekarski and B. H. Thomas. Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In *Proc. 5th IEEE International Symposium on Wearable Computers*, 2001.

[17] C. Taylor, P. Debevec, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. *ACM SIGGraph, Computer Graphics*, pages 11–20, 1996.

[18] A. van den Hengel, A. R. Dick, T. Thormählen, B. Ward, and P. H. S. Torr. VideoTrace: Rapid interactive scene modelling from video. *ACM Transactions on Graphics*, 26(3), 2007.