Augmented Reality Scouting for Interactive 3D Reconstruction

Bernhard Reitinger*[1] Christopher Zach [2] Dieter Schmalstieg [1]

[1] Institute for Computer Graphics and Vision, Graz University of Technology, Austria
[2] VRVis Research Center, Austria

Abstract

This paper presents a first prototype of an interactive 3D reconstruction system for modeling urban scenes. An Augmented Reality scout is a person who is equipped with an ultra-mobile PC, an attached USB camera and a GPS receiver. The scout is exploring the urban environment and delivers a sequence of 2D images. These images are annotated with according GPS data and used iteratively as input for a 3D reconstruction engine which generates the 3D models on-the-fly. This turns modeling into an interactive and collaborative task.

Keywords: scouting, interactive 3d reconstruction, urban planning

1 Introduction and Related Work

Generating 3D models of outdoor scenes in urban environments is often a demanding task, but necessary for applications such as mobile Augmented Reality (AR) [3, 12], interactive visualization [2, 9], or model-based tracking [13]. Creating these models is usually done in an offline process, using conventional 3D modeling tools, and involves tedious hours of manual data preparation.

In contrast, many interesting applications demand that models must be created on-line and on-site. For example, urban planners like to spontaneously experiment with variations of their architectural designs when inspecting a planned construction site. This means that the 3D model generation must be performed interactively to give immediate feedback. In general, most applications that require digital reconstruction of architecture can benefit from immediate feedback that allows to verify the reconstruction process. Providing this interactivity is the aim of the work presented in this paper.

However, traditional reconstruction techniques are aimed towards a high-accuracy off-line work style, where data acquisition and data processing are strictly separated. The objective of such systems is to obtain the best scalability of the overall process by full automation of the acquisition and reconstruction phase. For example, Akbarzadeh et al. [1] use geo-registered video sequences captured by a multi-camera setup mounted on a vehicle. This data is then post-processed in a separate off-line stage and finally generates 3D models of the captured environment.

Another approach only uses aerial images for reconstruction urban scenes [8]. Area-based segmentation is used to cluster homogeneous photometric properties and calculate a dense map to obtain the reconstruction. This method can be used to reconstruct large-scale architectural scenes. However, high quality aerial images must be available. Another approach is presented by Wang et al. where the texture of facades is reconstructed based on a number of photographs [16].

A different approach called *Photo Tourism* was presented by Snavely et al. [14]. In this project, similar images of the same building are taken from an existing database and processed in order to generate a sparse 3D point cloud. The camera positions and orientations are reconstructed for each image and image-based rendering is provided. Since this system aims at lots of similar images, the processing time for dozens of images is beyond one hour. Once the 3D reconstruction is finished, the result can be observed in an interactive viewer.

A large body of work on reconstruction algorithms can be found in the robotics community but will not be discussed here. Robotics as well as all the reconstruction works mentioned above aim at automated 3D reconstruction; none of them brings the human into the reconstruction loop. We propose to employ a human *AR scout* who is able to spontaneously explore and reconstruct environments which are not yet known. The resulting models can immediately be inspected and refined by the scout or used by a broader remote audience through a wireless connection. This transforms the usually off-line modeling task into an interactive task where a group of people and the scout generate models on-the-fly.

2 System Overview

Our proposed interactive reconstruction system consists of two main sub-systems, the scout and the reconstruction server (see Figure 1):

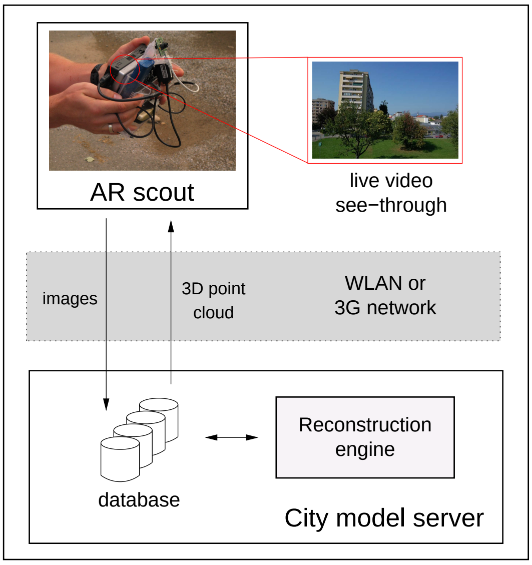


Figure 1: An overview of the interactive 3D reconstruction system. The AR scout stores current position and image data in the database. The reconstruction engine gets a notification and calculates the 3D model which is again stored in the database. Finally, the result can be visualized for a bigger audience on a projection screen.

The AR scout acquires geo-referenced image data with a handheld AR device and delivers it to a remote reconstruction server. The server is responsible for processing the individual images into

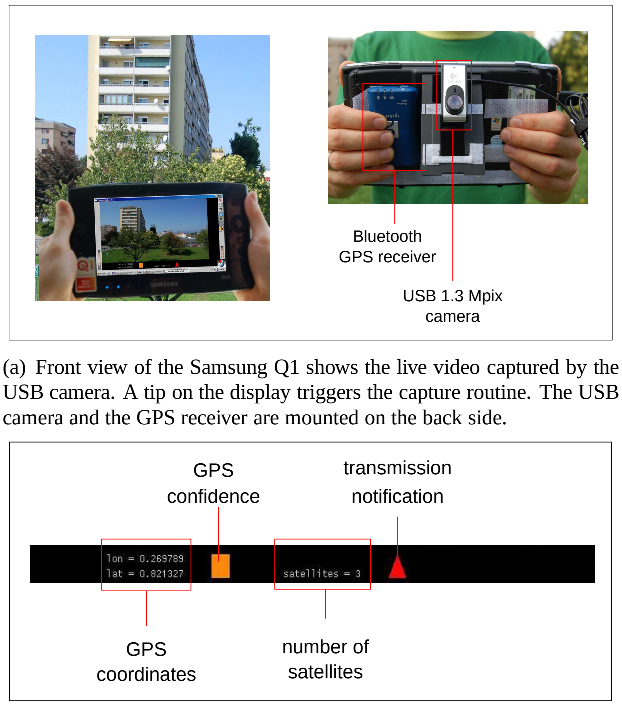
*e-mail: reitinger@tugraz.at

a 3D model (textured point cloud). Reconstruction is a very computationally intensive task and cannot be carried out with sufficient performance by a mobile computer. The server also stores the acquired and reconstructed data and makes it instantly available to remote users. The reconstructed model is returned to the scout for immediate 3D-registered display and inspection on the handheld AR device. If deficiencies are detected, the scout can use the AR device to acquire more data or prune erroneous reconstructions.

The AR scout is equipped with an ultra-mobile PC, an attached GPS receiver, and a USB camera. While exploring the environment, the scout takes several images for instance of a target building. These images are automatically annotated by current positioning data (taken from the GPS receiver). The enriched data is then transmitted to a custom database store [15]. This database is designed for multi-user data exchange based on the document object model.

Whenever a new image is stored in the database, the reconstruction engine gets a notification and triggers the reconstruction process (detailed in Section 3). The engine requires at least three different views in order to generate an initial 3D model. Each further image is added in an iterative way and updates the model accordingly within seconds. Again, the database server is used for storing the 3D model.

The AR scout equipment must be light-weight, connected to a network and equipped with sensors. Our setup consists of a handheld ultra-mobile PC (Samsung Q1) with a touch screen and a front-mounted camera. The AR user interface was developed using the *Studierstube* software framework [1]. Figure 2(a) shows the front and the back side of the handheld.



(a) Front view of the Samsung Q1 shows the live video captured by the USB camera. A tip on the display triggers the capture routine. The USB camera and the GPS receiver are mounted on the back side.

(b) The status bar of the capture application contains important feedback for the user such as current position or confidence of the signal.

Figure 2: The AR scout setup is used for capturing annotated image data in an urban environment.

A status bar (shown in Figure 2(b)) displays feedback on location, quality of the GPS signal, number of satellites, and a transmission notification. The user points the device at the target location like a digital camera, and triggers the image capture which are transmitted to the database together with corresponding GPS information using a WLAN or 3G connection. The GPS receiver with WAAS (wide area augmentation system) typically has a precision of 2-5 meters. Three or more images of a location in the database trigger the reconstruction procedure.

3 Reconstruction Engine

The reconstruction engine acts as a black box which takes 2D images and delivers 3D models. The main idea is that a sequence of 2D images (containing a sufficient overlap in image contents) is used to find correspondences between them. These correspondences can then be used to estimate the camera positions where the 2D image were taken. The mathematical framework to generate 3D geometry from multiple images is presented in [6]. Once the initial model is known, consecutive images can be related to each other, and a textured 3D point cloud can be computed by a dense matching approach. In the following, a brief overview of each individual task is given. The engine's pipeline is shown in Figure 3.

3.1 Camera Calibration

The reconstruction engine only works for calibrated cameras. For this reason the intrinsic camera parameters (focal length f, and the principal point (p_x, p_y)) are determined using a target calibration procedure in advance (e.g. [7]). In case of fixed lenses, the calibration procedure is needed to be performed only once. In addition to the camera intrinsic parameters f and (p_x, p_y) the utilized lens may have a significant distortion effect on the image, i.e. lines appear curved on the image. The parameters of this lens distortion can be e.g. determined using images of man-made objects containing straight lines. Computing the undistorted image from the original one is based on a look-up table obtained from the distortion parameters. Once the camera is calibrated, the information is passed on to the engine. Small deviations of the camera from the determined calibration results can be addressed later by the bundle-adjustment procedure (Section 3.3).

3.2 Feature Extraction

Since the input images contain too much redundant information for actual the reconstruction, the most relevant information required for finding correspondences must be extracted by using feature points. Feature extraction selects image points or regions which give significant structural information to be identified in other images showing the same objects of interest. We use Harris corners as feature points [5] which are well suited for sparse correspondence searches which is the case for urban scenes.

3.3 Correspondence and Pose Estimation

In order to relate a set of images geometrically it is necessary to find correspondences. For the task of calculating the relative orientation between images it is suitable to extract features with good point localization as provided by the feature extraction step (see above).

The relative orientation between two views taken from calibrated cameras can be calculated from five point correspondences. Hence a RANSAC-based approach is used for robust initial estimation of the relative pose between the first two views. We utilize an efficient procedure for relative pose estimation [11] in order to test many samples quickly. The result of this procedure is the relative orientation between these two views, but with unknown overall scale. The relative pose translates into known epipolar geometry, which represents the relationship of pixels in one view with images of the corresponding camera rays in the second view. With the knowledge of the relative poses between two views and corresponding point features visible in at least 3 images, the orientations of all views in the sequence can be upgraded to a common coordinate system

[1] http://www.studierstube.org

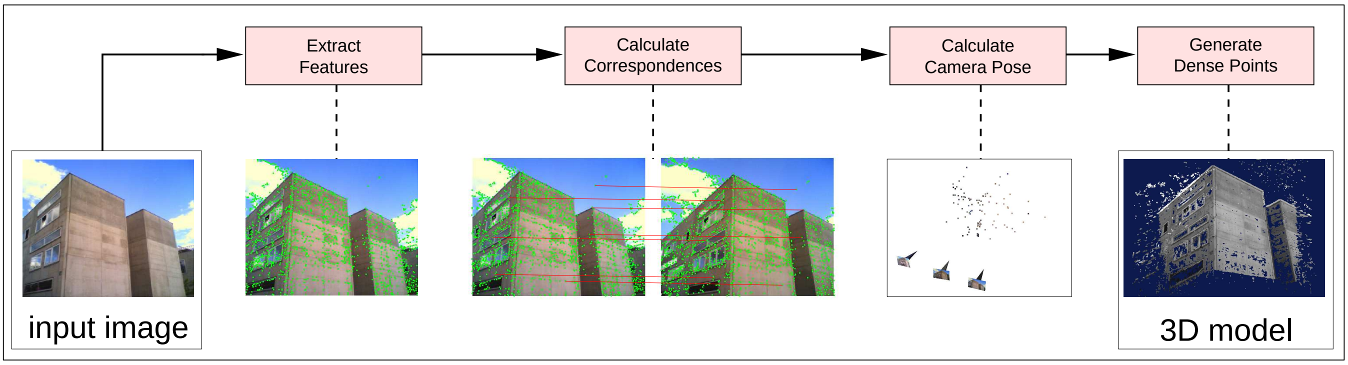


Figure 3: The reconstruction pipeline consists of four main components: feature extraction, correspondence search, camera pose estimation, and dense matching. Each captured image is passed through this pipeline in order to generate or enhance the 3D model.

by using an absolute pose procedure [4], again combined with a RANSAC approach to increase the robustness. The pose of every additional incoming image is calculated on the basis of 2D to 3D point correspondences, which is stronger than using the epipolar relationship between two views alone.

Purely image based reconstructions are located in a local coordinate system, which is upgraded to a world-reference system using the measured GPS locations of the camera positions.[2] The transformation from the local to the global system is a similarity transform in the case of calibrated cameras.

The camera poses and the sparse reconstruction consisting of 3D points triangulated from point correspondences are refined using a simple but efficient implementation of sparse bundle adjustment [10]. Our implementation allows the refinement of the camera intrinsic parameters and the integration of GPS data with estimated uncertainties as well. The output of this step are optimized camera orientations and intrinsic parameters in the first place. Additionally, sparse 3D points corresponding to the image features visible in several views are refined, too.

3.4 Dense Depth Estimation

With the knowledge of the camera parameters and the relative poses between the source views dense correspondences for all pixels of a particular key view can be estimated. Since the relative pose between the incorporated views is already known, this procedure is basically a one-dimensional search along the depth rays for every pixel. Triangulation of these correspondences results in a dense 3D model, which reflects the true surface geometry of the captured object in ideal settings.

We utilize a GPU-accelerated plane-sweep approach to generate the depth map for each source view [17, 18]. Plane sweep techniques in computer vision are simple and elegant approaches to image based reconstruction with multiple views, since a rectification procedure as needed in many traditional computational stereo methods is not required. The 3D space is iteratively traversed by parallel planes, which are usually aligned with a particular key view (Figure 4). The plane at a certain depth from the key view induces homographies for all other views, thus the reference images can be mapped onto this plane easily.

If the plane at a certain depth passes exactly through the surface of the object to be reconstructed, the color values from the key image and from the mapped references images should coincide at appropriate positions (assuming constant brightness conditions).

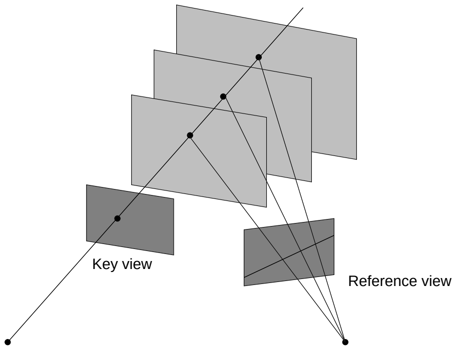


Figure 4: Plane sweeping principle. For different depths the homography between the reference plane and the reference view is varying. Consequently, the projected image of the reference view changes with the depth according to the epipolar geometry.

Hence, it is reasonable to assign the best matching depth value (according to some image correlation measure) to the pixels of the key view. By sweeping the plane through the 3D space (by varying the planes depth wrt. the key view) a 3D volume can be filled with image correlation values similar to the disparity space image (DSI) in traditional stereo. Therefore the dense depth map can be extracted using global optimization methods, if depth continuity or any other constraint on the depth map is required. We employ a simple winner-takes-all strategy to assign the final depth values for performance reasons.

3.5 Output

A depth map is generated as described above for every triplet of adjacent views, and the single depth images need to be fused into one common model. Currently, we employ a very simple technique: the depth maps are converted into colored point clouds (using the reference view for texturing), and these point sets are concatenated to obtain the combined model. This approach allows an easy incremental update of the displayed model after generation of a new depth map. Future work will address the creation of 3D surface meshes from the depth maps (e.g. [18]), which requires more complex methods to assign a texture to the resulting model.

4 Results

The first prototype was tested with multiple buildings at our campus. Additionally, we used it to reconstruct an ancient brick wall

[2] Since the GPS antenna and the projection center of the camera are very close, we ignore the resulting offset between them.

shown in Figure 5. Only some small clutter can be observed in the bottom of the resulting model.

The reconstruction time depends on the image resolution and the number of extracted features. For the above example, each pass of the pipeline takes less than one minute for uploading, feature extraction, finding correspondences, dense matching, and updating the 3D model.

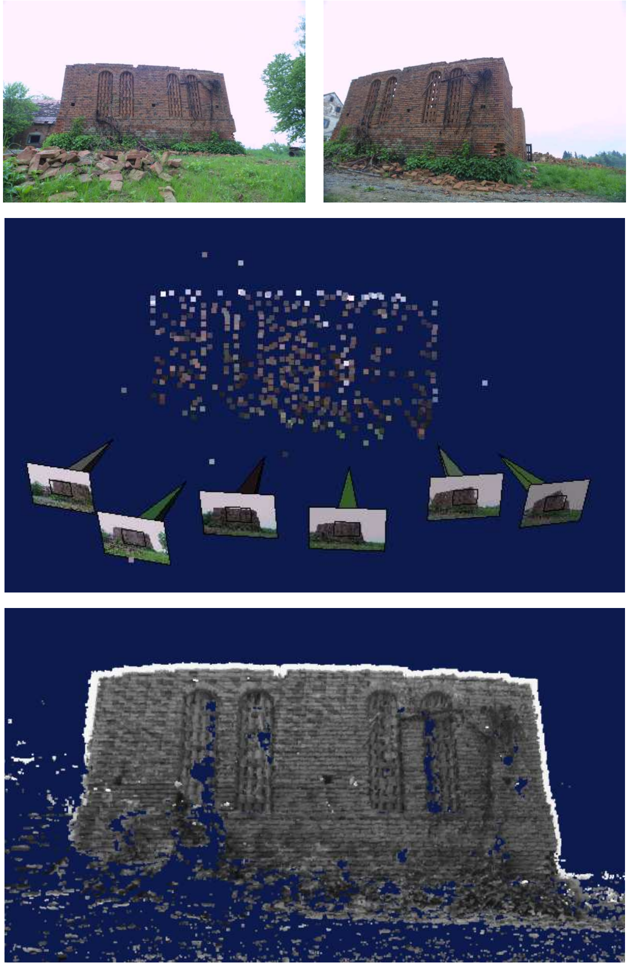


Figure 5: These screenshots show a test dataset of an old brick wall (with 6 input images). The image in the middle shows the reconstruction of the camera positions including sparse points. The image on the button shows a screenshot of the final 3D model.

5 Conclusion

AR scouting allows on-line generation of arbitrary 3D models in urban environments. The first prototype delivers promising results and works well with a handheld ultra-mobile PC. The resulting 3D models are currently represented by a textured 3D point cloud. Due to the GPS information, the reconstructed models are available in a global coordinate system and can be registered with available 3D geographic information systems.

In the near future we plan to replace the point cloud models with true surface meshes generated by a robust and incremental depth map integration technique. We also intend to test physically distributed collaborative 3D modeling with multiple scouts exploring the environment simultaneously and reconstructing larger areas. We also intend to perform a detailed quantitative analysis of the obtained models in terms of their reconstruction accuracy compared against conventional off-line reconstruction techniques.

Acknowledgements

This research is joint work between Graz University of Technology (sponsored partially by the European Union under contract FP6-2004-IST-4-27571 and the Austrian Science Fund FWF under contract Y193) and VRVis Research Center (funded by the Austrian government research program K-plus).

References

[1] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, and et al. Towards urban 3d reconstruction from video. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2006.

[2] H. Benko, E. Ishak, and S. Feiner. Collaborative mixed reality visualization of an archaeological excavation. In *Proc. of the International Symposium on Mixed and Augmented Reality (ISMAR 2004)*, 2004.

[3] S. Feiner, B. MacIntyre, T. Höllerer, , and T. Webster. A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment. In *Proc. of the ISWC '97 (First IEEE Int. Symp. on Wearable Computers)*, pages 208–217, 1997.

[4] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Analysis and solutions of the three point perspective pose estimation problem. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 592–598, 1991.

[5] C. Harris and M. Stephens. A combined corner and edge detector. *Proceedings 4th Alvey Visual Conference*, pages 189–192, 1988.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] J. Heikkilä. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(10):1066–1077, 2000.

[8] A. Huguet, R. Carceroni, and A. de A. Araújo. Towards automatic 3d reconstruction of urban scenes from low-altitude aerial images. In *Proc. of the 12th International Conference on Image Analysis and Processing (ICIAP'03)*, 2003.

[9] H. Ishii, J. Underkoffler, D. Chak, B. Piper, E. Ben-Joseph, L. Yeung, and Z. Kanji. Augmented urban planning workbench: Overlaying drawings, physical models and digital simulation. In *Proc. of the Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2002.

[10] M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, August 2004.

[11] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(6):756–770, 2004.

[12] W. Piekarski and B. Thomas. Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In *5th Int'l Symposium on Wearable Computers*, pages 31–38, Zurich, 2001.

[13] G. Reitmayr and T. Drummond. Going out: Robust model-based tracking for outdoor augmented reality. In *Proc. of the International Symposium on Mixed and Augmented Reality (ISMAR 2006)*, 2006.

[14] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, 25(3):835–846, 2006.

[15] D. Wagner and D. Schmalstieg. Muddleware for prototyping mixed reality multiuser games. In *Proc. of the IEEE Virtual Reality 2007*, 2007.

[16] X. Wang, S. Totaro, F. Taillandier, A. Hanson, and S. Teller. Recovering facade texture and microstructure from real-world images. In *Proc. 2nd International Workshop on Texture Analysis and Synthesis*, pages 145–149, 2002.

[17] R. Yang, G. Welch, and G. Bishop. Real-time consensus based scene reconstruction using commodity graphics hardware. In *Proceedings of Pacific Graphics*, pages 225–234, 2002.

[18] C. Zach, M. Sormann, and K. Karner. High-performance multi-view reconstruction. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2006.