Adaptive Enhancement of 3D Scenes using Hierarchical Registration of Texture-Mapped Models

Srikumar Ramalingam and Suresh K. Lodha
Department of Computer Science
University of California, Santa Cruz, CA 95064
{srikumar,lodha}@cse.ucsc.edu

Abstract

Adaptive fusion of new information in a 3D urban scene is an important goal to achieve in computer vision, graphics, and visualization. In this work we acquire new image pairs of a scene from closer distances and extract 3D models of successively higher resolutions. We present a new hierarchical approach to register these texture-mapped 3D models with a coarse 3D texture mapped model of an urban scene. First, we use the standard reconstruction algorithm to construct 3D models after establishing 1-1 correspondence between the feature points of two images at same resolution. Next, a subset of these feature points is used to register the higher resolution image with the lower resolution image using a scale-sensitive algorithm. Finally we register and consistently merge the 3D models at different resolutions. We present the results of our hierarchical algorithm for adaptive enhancement of a mural inside the UCSC Campus by registering data that differ in scale by a ratio of 1:15. Results indicate that the proposed hierarchical registration technique effectively utilizes the intermediate models to enable the smooth registration of the high resolution models on the coarser models.

Keywords: image registration, 3D model registration, scale-sensitive algorithms, 3D Reconstruction, hierarchical registration.

1 Introduction

Construction of 3D urban models is increasingly important in a variety of applications including urban planning, environmental monitoring, emergency relief scenarios, geospatial intelligent information visualization, virtual reality, and augmented reality. Many researchers have constructed 3D urban models from two or more images taken by hand-held cameras or a variety of sensors [5, 11, 2, 10, 20]. Typically, these models are constructed at a fixed resolution. In some cases, researchers have also constructed a highly detailed model of interior environments where dense set of images is acquired using mobile robots from various distances [18, 1]. A more likely scenario in outdoor 3D environment reconstruction is the acquisition of data using a variety of sensors during a short period of time to construct a model and then periodically update the model based on newly captured information using inexpensive hand-held sensors by mobile agents in outdoor environments.

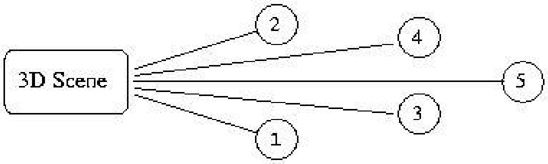

Figure 1: A schematic diagram showing the locations from where the images are captured. The image from location 5 is used to construct the coarsest model H-1. Images from locations 3 and 4 are used to construct the H2-Model. Additional image pairs are captured from successively closer locations.

In this paper we address one aspect of this adaptive fusion of new information – namely, the problem of enhancing the resolution of urban models by taking additional image sequences for specific sub-scenes. We acquire additional image sequences of a sub-scene from closer distances using a hand-held camera (see Figure 1) and extract 3D models at higher resolutions. The left diagram of Figure 2 shows that most of the existing registration approaches for 3D reconstruction of large scenes focus on collecting overlapping models at similar resolutions and mosaic them. The right diagram of Figure 2 shows our approach where the registration is needed at different scales. The challenge is to fuse the higher resolution images and models with the existing coarser level images and models. We achieve our goal through a hierarchical registration technique which utilizes the image properties of the intermediate data.

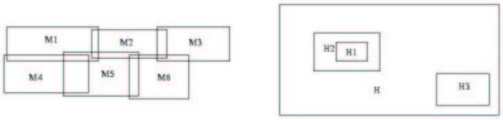

Figure 2: Left: most registration approaches use overlapping images for mosaicing. Right: adaptive registration at different scales used in this work.

We have constructed a coarse texture-mapped 3D urban model of a small region of UCSC campus using techniques similar to ones employed by other researchers [10, 20] More specifically, the elevation for the terrain is obtained using Digital Elevation Models (DEMs) freely available from USGS (United States Geological Survey). Elevation of the buildings is determined using LiDAR data, which we obtained using airborne sensors. High resolution aerial imagery obtained from local sources is texture mapped on the terrain. Finally, low resolution imagery for walls is obtained using a hand-held camera and texture mapped on the side walls using standard image processing operations involving image warping and mosaicing. The bottom diagram of Figure 12 shows the results of such a coarse texture-mapped 3D model of Thimann Labs area inside the UCSC campus. We then acquired additional image pairs of a mural embedded inside this model from closer distances using a hand-held camera (see Figure 1). The focus of this work is to describe an algorithm to hierarchically register these additional images and extracted 3D models with this coarse model.

2 Background and previous work

In this section we review previous work on outdoor environments reconstruction, 3D reconstruction from images, registration using image and geometry features including scale-sensitive registration.

2.1 Urban scene reconstruction

Large scale urban scene reconstruction has emerged as a very important problem in recent years [20]. Faugeras et al. [8] present a method of constructing 3D Euclidean models of urban scenes from a sequence of aerial images using uncalibrated cameras. Ribarsky et al. [20] describe a semi-automated urban construction approach using aerial or ground images and utilize shadow cues, counting floors and other factors. Fitzgerald et al. [10] have constructed 3D urban models using building footprints and LiDAR height data. Footprints are used to identify building locations and the LiDAR height information associated with points inside the building polygons are used to extrude buildings and assign heights to them. Images of the building walls are captured using digital camera and perspective distortions are corrected before pasting the images on the extruded building walls. As stated before, we have used a similar approach to construct the coarse level 3D model of a region inside the UCSC campus.

There are several efforts underway to construct large scale environments using a variety of sensors including GPS devices, omni-cameras, and range sensors. Bosse et al. [5] describe a system called *argus* which is a mobile platform fitted with high resolution digital camera, a GPS unit, orientation trackers and and is capable of capturing multiple georeferenced images of the urban scenes from various orientations and distances. Image registration and mosaicing is achieved by fusing complementary GPS, inertial sensors and odometry data through Kalman filtering. The authors also employ offline image-based registration techniques to refine pose estimates [14]. Allen et al. [2] describe an urban modeling system where the data is acquired using a GPS-equipped range sensor. Frueh et al. [11] also describe an urban modeling system where the data is acquired using a range sensor and path calibration is done using aerial imagery. Omni-directional sensors have been utilized to capture and create structure of complex indoor environments [18, 1]. None of these projects seem to have utilized scale-sensitive image or model registration in their work.

2.2 3D reconstruction from images

Extraction of depth information from images is a well known computer vision problem [21]. Figure 3 shows the standard pipeline for extracting 3D information from a pair of images using a calibrated camera. We have utilized this pipeline to generate 3D models from an image pair. We have used the work of Zhang [22] for camera calibration using a fixed pattern. The steps in 3D reconstruction are: (i) extraction of feature points using Harris corner detection algorithm in both the images, (ii) computation of correlation score for each pair of feature points, and categorizing each pair as a possible match if the score exceeds a certain threshold (this step may yield many-to-many correspondences), (iii) computation of match strength and utilization of relaxation strategies to remove false matches, (we have used *some winner take all stragety* for relaxation), (iv) using epipolar geometry along with a non-linear algorithm (least-median-of-squares) to remove outliers and select a set of robust matches, and compute camera motion parameters, (v) using back projection (also known as triangulation) to determine the corresponding 3D points, and (vi) finally, bundle adjustment to refine the 3D points and the camera motion parameters (translation and rotation) and improve the accuracy of the final 3D models. In our work we perform the bundle adjustment by computing the reprojection error (the error between the 2D feature points and the reprojected feature points from 3D points) and minimiz-

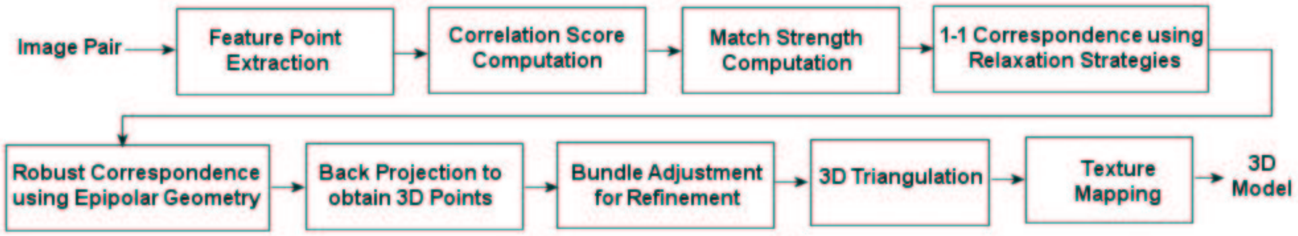
Image Pair
Feature Point Extraction
Correlation Score Computation
Match Strength Computation
1-1 Correspondence using Relaxation Strategies
Robust Correspondence using Epipolar Geometry
Back Projection to obtain 3D Points
Bundle Adjustment for Refinement
3D Triangulation
Texture Mapping
3D Model

Figure 3: 3D reconstruction from two images of similar resolution

ing it using the Levenberg-Marquardt iteration. These 3D points are then triangulated and texture-mapped to produce texture-mapped 3D models (see Figure 4). 3D reconstruction from multiple images has also been achieved [15].

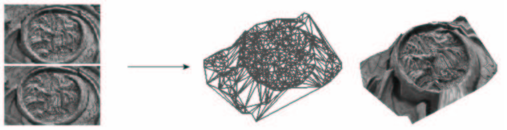

Figure 4: 3D texture-mapped high resolution model of a part of the mural.

Our work focuses on registering images and models at different resolutions. We now describe the previous work related to the two main approaches to register images and models – namely using image or geometry.

2.2.1 Registration using image features

The problem of matching two images where the viewpoints differ by small offsets in position, orientation, and viewing parameters such as focal length has been a well-studied problem for more than two decades. Several authors have also addressed the problem of matching two images gathered from two very different viewpoints but they did not consider a large change in resolution [9, 16, 6, 19]. The low-high resolution image matching task consists in finding a small region in the low resolution image that can be assigned to the high resolution image. Even an assumption of planar projective transformation between the two images leads to a very large search space and the associated non-linear minimization procedure has to deal with a four-parameter cost function [3]. Hansen and Morse [12] describe a method based on point-to-point correspondences where they are able to recover the scale factor between the two images but they cannot deal with camera motions. Our scale-sensitive image-based registration work (described later in Section 3) is perhaps most closely related to the recent work by Dufournaud et al. [7], where they are able to match images differing in scale by a factor of up to 5. They obtain point-to-point correspondences using a scale-sensitive Harris Corner's detection algorithm. Our work differs from theirs in at least two important ways. First, we introduce a scale-sensitive image registration algorithm, the motivation for which is described further below. Second, we match both images and models at different resolutions. We achieve this by ensuring that features points in 1-1 point correspondences obtained by matching images at different resolutions (say high-res image 1 with low-res image 1) are a subset of the feature points in the 1-1 point correspondences obtained by matching images at same resolutions (say high-res image 1 and 2 and low-res image 1 and 2). Therefore, instead of applying scale-sensitive Harris Corner detection algorithm, we start with already discovered robust 1-1 point correspondences between images at same resolutions (see Figure 5) and then look for a subset of these matches using a scale-sensitive image registration to match images at different resolutions. Due to our requirement of subset of matches, we are able to match images at different resolutions that differ by a factor only up to 2, but we are able to boot-strap this method to hierarchically register several layers of both images and models. In Section 4, we present an example where we register 7 layers so that the scale ratio between the lowest and highest resolution images is approximately 1:15. We are able to carry out this hierarchical registration for several layers because the number of robust final matches after subsetting seems to be much higher at ratios of 1:2 than at higher scale ratios such as 1:5.

2.2.2 Registration using geometry

An alternative to image-based registration between images and models is to first register 3D models at different resolutions and then try to match images (reverse process of what we are trying to do in this work). The most popular and successful geometry-based registration approach is the ICP (iterative closest point) algorithm introduced by Besl and Mckay [4]. Several efficient variants of this algorithm are

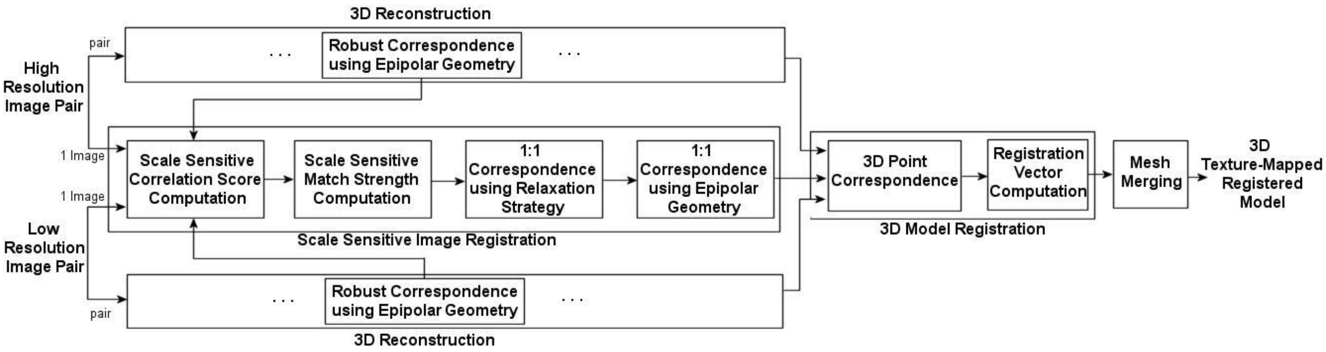

Figure 5: One step of the hierarchical registration of low-resolution images and models with high-resolution images and models

described by Rusinkiewicz et al. [17]. The variants are introduced by affecting one of the following six stages in the ICP algorithm: selection of the points in the data set, matching the points in one mesh with the points in other mesh, weighting of the corresponding pairs, rejection of certain pairs, assignment of error metric and finally the minimization of the error metric. The ICP algorithm and its variants has primarily been used in alignment of 3D information obtained directly from 3D scanners. It seems that most of the existing literature focuses on registration of 3D point clouds at similar resolution. One of the limitations of this technique is the computation of the initial state vector, which plays a major role in avoiding local minima problem. If the scale between two models is not known a-priori, it is not clear how to extend this algorithm to scale-sensitive model registration. Nevertheless, this approach has some merits and we have explored this venue to a limited extent as described in Section 3.2.

3 Algorithm

The proposed algorithm for hierarchical registration of texture-mapped models when data is acquired from image pairs at successively higher resolutions can be described in four major steps. The first step is the 3D model reconstruction from images at similar resolution using the standard techniques illustrated in Figure 3 and described briefly in Section 2.2. The second step is the scale-sensitive image registration algorithm described in Section 3.1 and illustrated in Figure 5. The third step is the registration of 3D models at different resolutions and mesh merging described in Section 3.2. The fourth and the final step is the hierarchical boot-strapping of the first three steps of this algorithm that allows us to register several layers of data on the initial coarse model. Results of applying the hierarchical registration are described in Section 4.

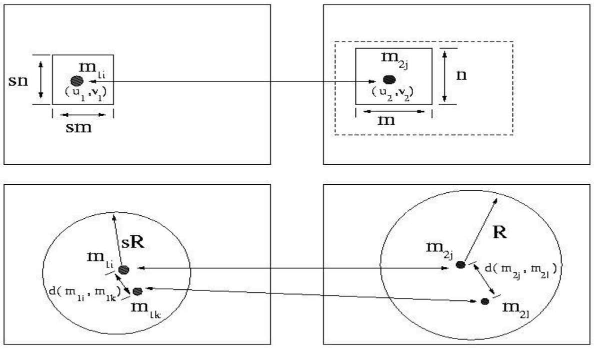

Figure 6: (top) Correlation score computation, and (bottom) match strength computation between feature point pairs in two images at different resolutions.

3.1 Scale-sensitive image registration algorithm

In order to register images at different resolutions, we have introduced a variation to the well known feature correspondence algorithm (presented in Figure 3 and described in Section 2.2) by introducing a scale-sensitive correlation score computation and scale-sensitive match strength computation that we describe below. As shown in Figure 5, the input to the correlation score computation algorithm is (i) feature points P_H found in one (say left) image of a high resolution image, (ii) feature points P_L found in the same (left) image of a low resolution image. Subsequent steps are performed only on these pairs of feature points.

Correlation score computation: Rather than using the correlation windows of same size, here we introduce a scaling factor s to correlate a window of size sm by sn centered at (u_1, v_1) with another window of size m by n centered at (u_2, v_2), as shown in the top diagram of Figure 6. Let m_{1i} and m_{2j} are the corresponding feature points located at (u_1, v_1) and (u_2, v_2) respectively. The scaling correction factor s, which is a fractional value between 0.0 to 1.0, is applied to the image at higher resolution.The modified expression for computing the correlation score $c_s(i,j)$with the scaling correction s is as follows:

$$\frac{\sum_{i=-n,-m}^{n,m}[I_1^s(u_1+si,v_1+sj)-\overline{I_1^s(u_1,v_1)}][I_2(u_2+i,v_2+j)-\overline{I_2(u_2,v_2)}]}{(2n+1)(2m+1)\sqrt{\sigma^2(I_1^s)\sigma^2(I_2)}}$$

where $\overline{I_1^s(u_1,v_1)}$ and $\sigma(I_1^s)$ are the average and standard deviation respectively at the point (u_1,v_1) in the neighborhood $(2sm+1)\times(2sn+1)$, while $\overline{I_2(u_2,v_2)}$ and $\sigma(I_2)$ are the average and standard deviation at the point (u_2,v_2) in the neighborhood $(2m+1)\times(2n+1)$. If the correlation score between the feature points m_{1i} and m_{2j} exceed a certain threshold C then (m_{1i},m_{2j}) is considered as a candidate match, otherwise not.

Match strength computation: The previous step of correlation score computation may yield many-many correspondences between feature proints. The idea of match strength computation is to select a 1-1 correspondence from the set of candidate matches. In order to deal with images at different resolutions, we scale one of the neighborhoods and modify the distance formula as described below. Consider a match (m_{1i},m_{2j}), where m_{1i} and m_{2j} are the feature points in first and second image respectively. We consider a neighborhood $N(m_{2j})$ of radius R around the feature point m_{2j} and a neighborhood $N^s(m_{1i})$ of radius sR around m_{1i}. Let n_{1k} and n_{2l} refer to the feature points belonging to the neighborhoods $N^s(m_{1i})$ and $N(m_{2j})$ respectively. Let $d(a,b)$ refers to the Euclidean distance between a and b respectively. The *strength of the match* $S(m_{1i},m_{2j})$ between m_{1i} and m_{2j} is as follows:

$$c_s(i,j)\sum_{n_{1k}\epsilon N^s(m_{1i})}[max_{n_{2l}\epsilon N(m_{2j})}\frac{c_s(k,l)\delta_{ijkl}}{1+d_{ijkl}}],$$

where $d_{ijkl}=[d(m_{1i},n_{1k})+sd(m_{2j},n_{2l})]/2$,

$r_{ijkl} = \frac{|d(m_{1i},n_{1k})-sd(m_{2j},n_{2l})|}{d_{ijkl}}$ represents the relative distance difference and

$$\delta_{ijkl}=\begin{cases} e^{\frac{-r_{ijkl}}{\epsilon_r}} & \text{if } (n_{1k},n_{2l}) \text{ is a match and } r<\epsilon_r \\ 0 & \text{otherwise} \end{cases},$$

where ϵ_r is the threshold on the relative distance difference.

The remaining steps of the algorithm – relaxation strategy, use of epipolar geometry along with a least-squares-of-median algorithm to remove outliers, obtain a robust set of matches and compute camera motion parameters – remain the same. In order to find the optimum scale ratio, we discretized the values of s from 0.5 to 1.0 with an increment of 0.1 and maximized the number of robust matches for appropriate correlation thresholds (see Figure 7). The top row of Figure 8 shows the final outcome of this stage of the algorithm.

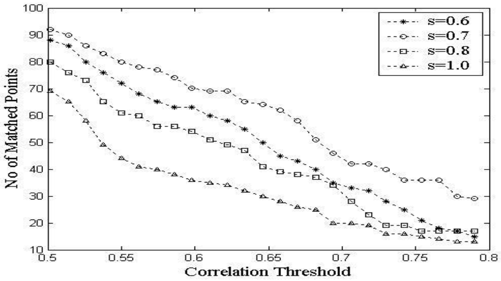

Figure 7: Number of matches selected for different values of *s* using different correlation thresholds in the scale-sensitive image matching algorithm. We select *s=0.7* for this case because it clearly outperforms other scale factors in terms of number of robust matches for different values of thresholds.

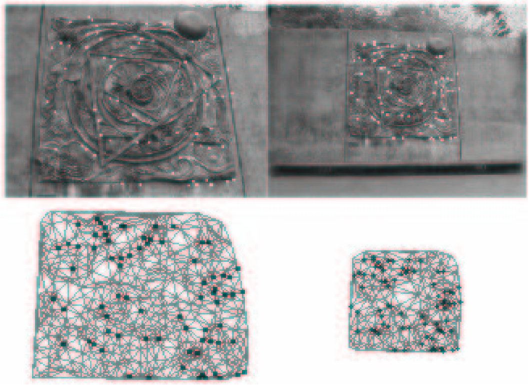

Figure 8: (top row) 73 robust matches between a high-res and a low-res image; (bottom row) feature points occurring in the scale-sensitive image registration algorithm (shown in drak) are a subset of feature points found during the 3D reconstruction of the respective models.

3.2 Scale-sensitive 3D Model Registration

The end result of the previous step is a set of robust 1-1 correspondences between 2D feature points of a high-resolution and a low-resolution image. Since these feature points are a subset of feature points used in 3D model construction, we use the standard back projection (also known as triangulation) to obtain the correspondences between the 3D points of high and low resolution models (see the bottom row of Figure 8).

We now need to find the scaling, rotation, and translation vectors in three dimensions to register the two models. We first compute scaling factor (S) between the two models as follows. For every possible pair (i,j), we compute the scale factor $S_{i,j} = \frac{d(H1_i,H1_j)}{d(H2_i,H2_j)}$. We first use this scale factor to remove the ourliers using a histogrambased approach. We then use the scale computation [13]: $S_{H,L} = (\frac{\sum_{i=1}^{n} d(\mu_L, L_i)^2}{\sum_{i=1}^{n} d(\mu_H, H_i)^2})^{\frac{1}{2}}$, where μ_L and μ_H refer to the centroids of the 3D points in low and high resolution models. Now we use the standard quaternion approach [13] to compute the rotation and translation vector to minimize the least squares error between the 3D points of the two models. We also attempted to improve the registration of 3D models at different resolutions further by applying the ICP (iterative closest point) algorithm. However, we learned that the image based registration was already very accurate and the improvements were very marginal. However, it is possible that ICP algorithm can improve results further in additional scenarios, and we plan to investigate them in future.

3.2.1 Mesh Merging

We now have a high resolution mesh registered and superimposed over the low resolution mesh. We now describe a mesh merging approach in which we merge these two meshes with overlapping region to obtain a single consistent mesh.

vertices inside convex hull	edges inside convex hull	edges intersecting convex hull	case number	required triangulation
3	3	0	3	not required
2	1	2	2	Tri-2
1	0	2 3	1a 1b	Tri-2 Tri-1
0	0	0 1 2 3	0a 0b 0c 0d	not required Tri-3 Tri-2,Tri-1 Tri-1

Table 1: Classification of intersections of a triangle from outer low-resolution mesh with the convex hull of the inner mesh

First, the convex hull of the high resolution mesh is computed and superimposed on the outer low resolution mesh (see the top left diagram of Figure 11). Next, (i) the triangles in the low resolution mesh which are completely inside the convex hull are eliminated, (ii) the triangles in the low resolution mesh which are completely outside the convex hull are preserved, (iii) the triangles in the high resolution mesh that do not share an edge with the convex hull are also preserved. The main task lies in splitting and retriangulating the triangles belonging to both the low-res and high-res mesh in the overlapping regions (see the top right diagram of Figure 11).

Possible intersections of a low-res triangle and the convex hull are summarized in Table 1. There are three possible scenarios – Tri-1, Tri-2, and Tri-3. The intersecting triangles are then re-triangulated in each of these cases as shown in Figure 10. This completes the processing of the low resolution mesh.

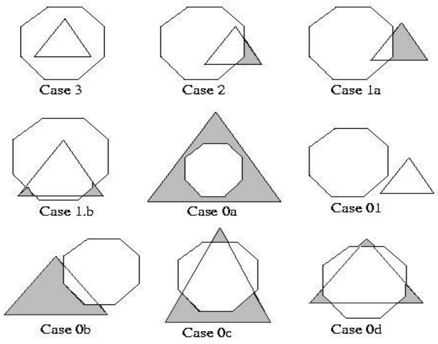

Figure 9: Possible intersections between a low-resolution triangle and the convex hull of high-resolution mesh.

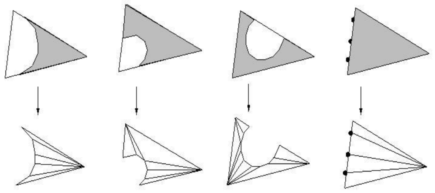

Figure 10: Retriangulations (from left to right): Tri-1 scenario, Tri-2 scenario, Tri-3 scenario, and Tri-4 scenario.

The above process introduces intersections points of the newly triangulated low resolution mesh with the edges of the convex hull. At these intersection points, we change the depth values of the high resolution mesh to the depth values of the low resolution mesh to avoid any holes in the finally merged single mesh. In addition, the triangles of the high resolution mesh sharing the boundary edge are also retriangulated (Tri-4 scenario shown in the right most diagram of Figure 10.)

4 Results

The proposed algorithm in this paper is tested on real data sets by taking images obtained using a Sony digital camcorder. We used Zhang's easy camera calibration tool [22] to calibrate the camcorder. We have used Matlab for implementing most of our algorithms. We have used OpenGL libraries and C++ programs to render and view the texture-mapped models.

We have conducted experiments with several data sets. Here we present the results for the mural data. We constructed seven models at different resolutions. The coarsest resolution model (H1), which is a part of the coarse 3D texture mapped 3D urban model, was built and registered using the techniques described in Section 1. All intermediate resolution models (H2 to H6) and the highest resolution model

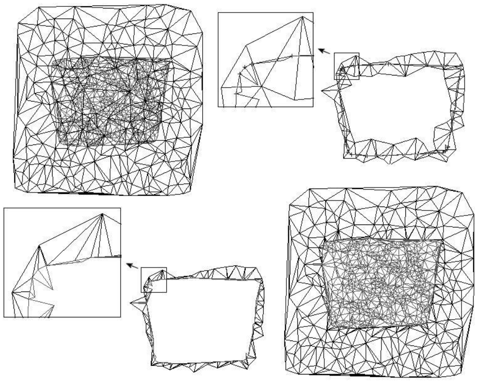

Figure 11: (top left): high and low resolution meshes superimposed, (top right): intersecting regions where the triangles need to be split and retriangulated, (bottom left): intersecting region after splitting and retriangulation to obtain a consistent mesh, (bottom right): consistently merged high and low resolution mesh.

(H7) were built using the algorithm described in Section 2.2 (see Figure 12). Statistics are summarized in Table 2. Although the initial reprojection error was very high for H5-model because the two images used in the 3D reconstruction differed by a very small rotation, which is not very good for the working of the 3D reconstruction algorithm using epipolar constraint. However there was a significant improvement after using bundle adjustment algorithm.

Statistics for registering a high resolution image and model with the next lower resolution image and model using the algorithm described in Section 3 are summarized in Table 3. In our experiments, we found that the image registration algorithm works well for scale ratios of up to .5:1. Although a more fine sampling of the scale ratio s may have an improvement in the number and robustness of the matches, we found that this improvement is not critical in registration vector computation for 3D model registration. Since the texture mapping is perfomed finally on the registered 3D models, it is the accuracy of the registration of 3D models that matter and not the initial estimate of the scale ratios between the images at different resolutions. We can also adaptively enhance or simplify the scene by registering or removing the intermediate models based on the viewpoint and the resolution of the scene. The mesh merging algorithm can jump across the hierarchy so that one can merge only the required models. For example, it is possible to merge and texture map H2, H5, H6 and H2 models only (see bottom row of Figure 13).

5 Conclusion

We have proposed and implemented a hierarchical algorithm for registering images and models at different resolu-

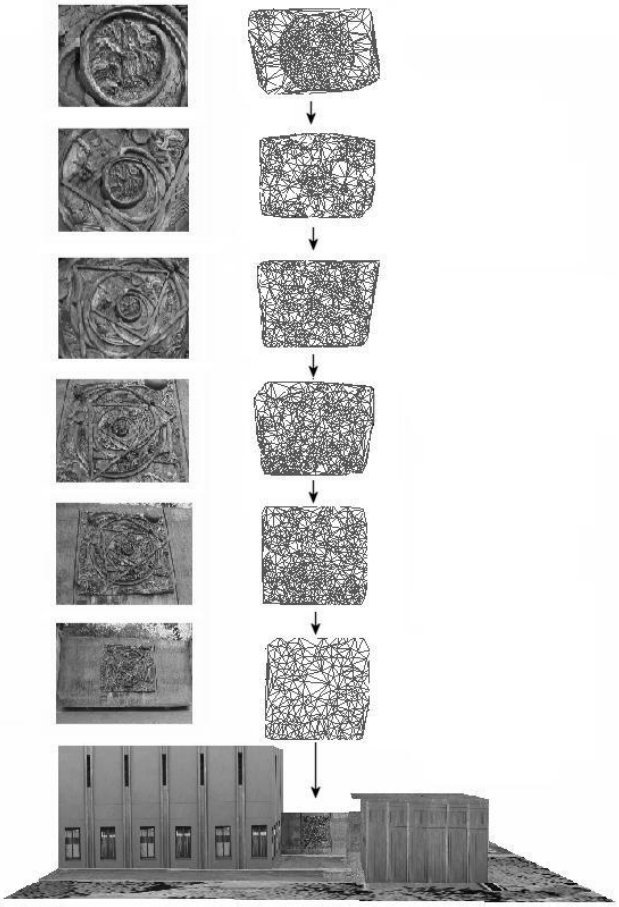

Figure 12: Images and corresponding 3D triangulated models at six levels of hierarchy to be registered on the coarse urban scene at the bottom.

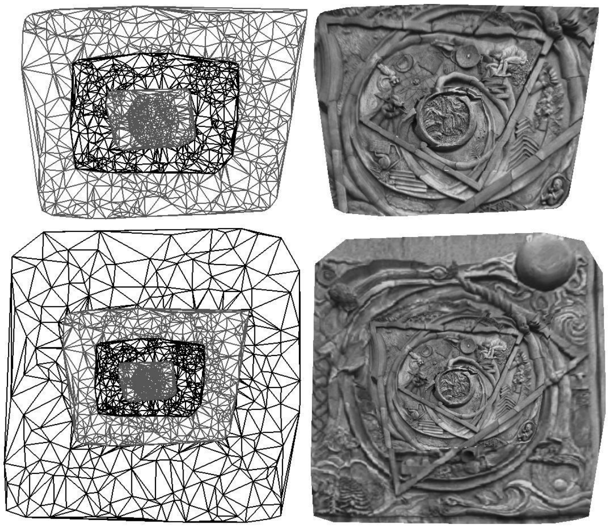

Figure 13: Registered, mesh-merged and texture-mapped models (top row) H-5, H-6, and H-7, (bottom row) H-2, H-5, H-6, H-7

I	FP	SM	RM	EM	IE	FE$\times 10^{-3}$
H2-1 H2-2	1027 961	879	366	252	0.8080	0.185
H3-1 H3-2	1650 1611	3175	682	583	0.6677	0.074
H4-1 H4-2	1292 1287	961	706	680	0.1299	0.13362
H5-1 H5-2	1098 1045	2009	656	607	5.7771	0.14353
H6-1 H6-2	978 957	2429	602	562	0.8929	0.1666
H7-1 H7-2	1526 1528	2166	765	674	0.7876	0.1841

Table 2: Construction of 3D Models (H-2,3,4,5,6,7) using image pairs: SM=matches after match strength computation, RM=matches after relaxation strategy, EM=robust matches after applying epipolar constraint, IE=Initial reprojection error, FE=final reprojection error after bundle adjustment.

I	FP	s	RM	EM	S	E
H1-1 H2-1	197 252	0.70	58	40	176.8824	0.0003
H2-1 H3-1	252 583	0.70	89	73	109.2049	0.0010
H3-1 H4-1	583 680	0.80	76	57	35.3214	0.0018
H4-1 H5-1	680 607	0.50	45	45	28.5310	0.0164
H5-1 H6-1	607 562	0.70	26	22	15.4088	0.0004
H6-1 H7-1	562 674	0.50	12	11	7.7527	0.0025

Table 3: Scale-sensitive image and model based registration for six levels of hierarchy: I=images, FP=feature points, s=scale ratio, RM=matches after relaxation strategy, EM=robust matches after applying epipolar constraint, S=scale for 3D registration, E=Least squares registration error

tions. For the case of mural, we have demonstrated that we are able to hierarchically register the data with an approximate scale ratio of 1:15. There is no inherent limitation in the proposed algorithm to carry this further with even higher scale ratios so long as the successive ratios between the two layers is approximately 1:2. We plan to investigate scale-sensitive algorithms further to remove this constraint.

References

[1] D. G. Aliaga, T. Funkhouser, D. Yanovsky, and Ingrid Carlborn. Sea of images. In *Proc. of IEEE Vis-2002*, pages 331–338, Oct 2002.

[2] Peter Allen, Ioannis Stamos, Atanas Gueorguiev, Ethan Gold, and Paul Blaer. Automated site modeling in urban environments. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, 28 May-1 June 2001.

[3] A.W.Gruen. Adaptative least squares correlation: a powerful image matching technique. *S.Afr.Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3):175–187, 1985.

[4] Paul J. Besl and Neil D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence-1992*, 2, Feb 1992.

[5] M. Bosse, D. De Couto, and S. Teller. Eyes of Argus: georeferenced imagery in urban environments. *GPS World*, pages 20–30, April 2000.

[6] D.Tell and S.Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. In *Proceedings of the 6th European Conference on Computer Vision*, pages 814–828, 2000.

[7] Yves Dufournaud, Cordelia Schmid, and Radu Horaud. Matching images with different resolutions. In *Proceedings of the Conference of Computer Vision and Pattern Recognition*, pages 612–618. IEEE Computer Society Press, June 2000.

[8] Olivier Faugeras, Stephane Laveau, Luc Robert, Gabriella Csurka, and Cyril Zeller. 3D reconstruction of urban scenes from sequence of images. *Research Report, Sophia Antipolis, Project Robotvis*, June 1995.

[9] F.Georgis, M.Petrou, and J.Kittler. On the correspondence problem for wide angular seperation of non-coplanar points. *Image and Vision Computing*, pages 16–35, 1998.

[10] Brian Fitzgerald. Virtual reality meets GIS: Urban 3D modeling in South Carolina. *Geospatial Solutions*, July 2002.

[11] Christian Fruh and Avideh Zakhor. 3D model generation for cities using aerial photographs and ground level laser scans. *CVPR-2001*, 2:31–38, december 2001.

[12] Bruce B. Hansen and Bryan S.Morse. Multiscale image registration using scale trace correlation. *Computer Vision and Pattern Recognition-Volume 2*, June 1999.

[13] Berthold K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society A*, 4(4):629–642, April 1987.

[14] J.P.Mellor. Geometry and texture from thousands of images. *3D Structure from Images-SMILE 2000,Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 170–186, July 2000.

[15] Marc Pollefeys. 3D modeling from images. *Tutorial organised in conjunction with ECCV 2000*, June 2000.

[16] P.Pritchett and A.Zisserman. Wide baseline stereo matching. In *Proceedings of the 6th International Conference on Computer Vision*, pages 754–760, 1998.

[17] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, 28 May-1 June 2001.

[18] Camillo J. Taylor. Videoplus: A method for capturing the structure and appearance of immersive environments. *3D Structure from Images-SMILE 2000,Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 187–203, July 2000.

[19] T.Tuytelaars, L. Van Gool, L.D'haene, and R.Koch. Matching of affinely invariant regions for visual servoing. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1601–1606, 1999.

[20] Tony Wasilewski William Ribarsky and Nickolas Faust. From urban terrain models to visible cities. *IEEE CG&A*, 22(4), 2002.

[21] Gang Xu and Zhengyou Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition-A Unified Approach*. Kluwer Academic Publishers, 1996.

[22] Zhengyou Zhang. A flexible new technique for camera calibration. *Technical Report, Microsoft REsearch, Microsoft Corporation, Redmond, WA 98052*, December 1998.