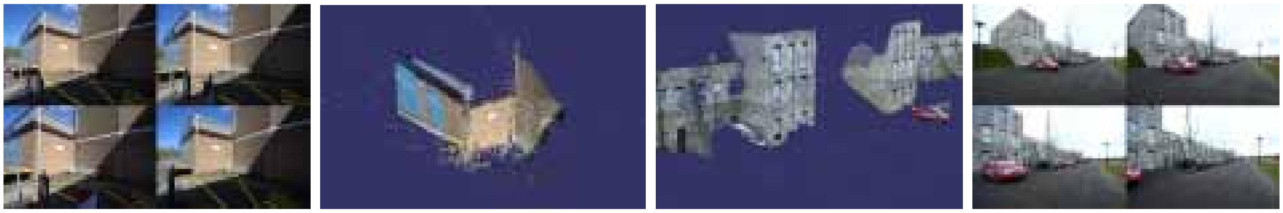
Automatic 3D Modeling of the Urban Landscape

Isaac Esteban[1,2] and Judith Dijk[2] and Frans Groen[1]



Abstract— **In this paper we present a fully automatic system for building 3D models of urban areas at the street level. We propose a novel approach for the accurate estimation of the scale consistent camera pose given two previous images. We employ a new method for global optimization and use a novel sampling technique for primitive fitting to obtain an efficient representation of the scene. We combine these techniques to obtain a 3D representation based on textured planar patches using a single handheld digital reflex camera (DSLR).**

I. Introduction and Motivation

Obtaining 3D models of the urban landscape has received much attention in the recent literature due to both the wide range of applications and the intriguing complexity of the problem. The classical approach to building 3D models is to use laser range scanners or active stereo due to their accuracy. These solutions are commonly considered the standard method for obtaining ground truth data. However, they are usually expensive and involve a complex and time demanding data acquisition process.

Recently, 3D modeling using passive techniques based on computer vision principles have gained popularity. Methods based on images taken at the ground level are specially interesting as they capture the scene from the user's point of view. In aerial images based techniques the facades are poorly reconstructed. The spectrum of current solutions in the field of ground image based 3D reconstruction range from user guided interfaces [1] to fully automated systems based on large collections of images or video streams [2] [3].

In this paper, we investigate the robust creation of 3D models of urban scenes using a single camera. Our system presents a number of advantages with respect to similar systems. It is a straight forward and low cost solution. It is user friendly since images are taken with an affordable off-the-shelf DSLR camera. And it provides an efficient representation of the scene based on small textured planar patches.

We present an approach that is a combination of very well know computer vision techniques and a set of novel approaches to the tasks of robust motion estimation and 3D modeling. Our system is based on three fundamental steps. Firstly, the motion between consecutive images is estimated using a robust method for monocular scale-consistent ego-motion. Second, the motion estimation is refined using an iterative procedure and a 3D point cloud is obtained. Finally, a set of small planar rectangular patches is fitted to the 3D points and the texture of the planes is determined.

This paper is organized as follows: section II briefly describes some of the most significant work in the field and establishes the relations to our methods. Section III presents the algorithms used for egomotion and details our novel approach for a robust estimation and solving for the scale problem. Section IV details our method for the iterative refinement. Section V reviews the modeling process in which an efficient representation of the scene is obtained. Section VI presents some of the results of our system for modeling urban landscapes and finally section VII discusses the results and draws some conclusions and points out future work.

II. Related Work

In this section we will discuss some of the latest work on ground-based computer vision systems for 3D modeling of urban or architectural scenes.

The spectrum of related scientific literature [4][1][2][5][6][7] ranges from user guided systems to fully automated techniques. User guided methods usually require only a set of unordered images that depict a single urban area, typically an architectural landmark. Sinha et. al. [1] present a system that combines computer vision techniques for motion estimation and a user friendly interface to obtain highly detailed 3D models of architectural structures. In their work, the user is responsible for selecting a set of planar polygons that represent planes in the real world. Earlier work from Cipolla et. al. [6] also construct a triangle based 3D model where the user has to select edges which are either perpendicular or parallel in the world. Debevec et. al. [7] also present a hybrid system in which the user draws primitives on some of the images and the model is fitted according to the rest of the photographs.

Intelligent Systems Laboratory Amsterdam[1] (ISLA), University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Electro-Optical Systems[2], TNO Defence, Security and Safety, Oude Waalsdorperweg 63, 2509 JG The Hague, The Netherlands, {isaac.esteban, judith.dijk}@tno.nl and groen@science.uva.nl - 978-1-4244-7286-4/10/$26.00 2010 IEEE

At the other end of the spectrum, Mordohai et. al. [2] present a fully automated system based on multi-stereo video streams coupled with GPS and INS information. They apply their techniques to mapping large portions of an urban scene to obtain a dense 3D texture mesh in real time. Fruh et. al. [8] combine 2D laser scanners with a camera mounted in a vehicle and aerial images to obtain a dense point cloud that they convert to a triangular mesh. Cornelis et. al. [5] present an approach in which they combine computer vision with object recognition to obtain a simple 3D model where cars are recognized and substituted by accurate 3D representations while facades and ground are represented as planar surfaces.

Werner and Zisserman [4] present a fully automated 3D reconstruction system for architectural scenes. They first generate plane based 3D models from sets of three images and then use those planes to guide the search for protrusions and indentations. Their system however requires that two orthogonal horizontal directions are visible on the scene, which is not always the case. Once the basic planar model is found, they refine the model by fitting rectangular block models for doors and windows and wedge models for dorm windows. Their approach is interesting but limited to three images and certain models for the refinement. They also have a very strong limitation on the visibility of the three principal directions. As they use the trifocal tensor to calculate the relative position of the images, the reconstructed point cloud is usually consistent.

Agarwal et. al. [9] present an approach based on the solution of a very large bundle adjustment problem using thousands of computer cores. It is worth noticing that our approach could be integrated with such a large system to obtain an initial solution that is closer to the real solution, reducing therefore the total computational cost.

Our approach to the problem of 3D urban modeling is significantly different from the systems in the related literature: we do not limit our system to three images, rather we present experimental results with up to 60 images. We do not consider any limitation on the scene that is depicted except the fact that it can be discretized by planar patches, which is a sound assumption for modeling urban areas. We employ a novel approach to the problem of scale adjustment and iterative refinement to obtain a consistent point cloud and motion estimation from a monocular handheld camera. We fit planar patches to the reconstructed point cloud with a robust RANSAC [10] approach (RANdom SAmple Consensus), also we do not need vanishing points or assumptions on the relative positions of the planes. Finally, we also employ a texturing technique to minimize the occlusion and reflexion in typical urban scenes.

III. Structure from Motion

The first step in the modeling process is the estimation of the urban sparse 3D structure that is depicted in the set of images. We accomplish this by first estimating the frame-to-frame motion, then computing the scale ratios and finally obtaining the 3D point cloud.

A. Motion Estimation

In order to compute the frame-to-frame motion we first extract a set of salient features in every frame. We employ SIFT [11] features and descriptors as they are the method of choice in the related literature and usually robust to camera motions in an urban environment. We then match the features across consecutive frames (see figure 1) using nearest neighbors and a minimum distance threshold in the SIFT descriptor space, obtaining matches between frames F_i and F_{i+1}. Given these image feature matches, we use the normalized 8-point algorithm to compute the frame-to-frame motion as described by Zisserman [12] due to its computational simplicity. Outliers are then rejected between frames F_i and F_{i+1} using RANSAC and the final motion is re-computed using only the set of inliers. This yields a fundamental matrix F that describes the camera motion. Given that we calibrated our camera beforehand, we know the calibration matrix K. We then obtain the essential matrix with the matrix product $K^{'}FK$. The frame-to-frame rotation and translation $[R|T]$ is finally obtained using the method from Horn [13]. This yields four possible solutions from which we select the one for which more inliers are in front of both cameras.

Camera motion is only estimated up to scale so only the direction of the translation T is available. This is the so called scale problem in monocular vision. Since we have multiple frames, the translation estimated between each pair of consecutive frames is only determined up to an unknown and different scale factor. Figure 1 shows the motion estimation for 3 frames. As only the direction of the translation is estimated, the distances $d_{(0,1)}$ and $d_{(1,2)}$ are not recovered. Before the 3D structure can be reconstructed, the ratio between these distances needs to be calculated.

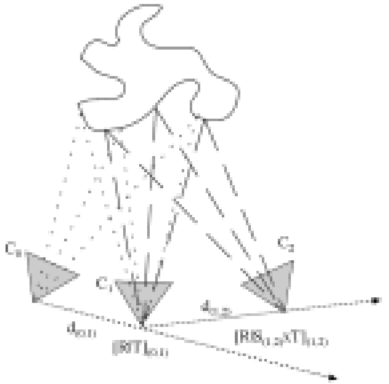


Fig. 1. Frame-to-frame motion estimation. In this graphic 3 cameras are shown (C_0, C_1 and C_2). A different set of image features corresponding to the 3D structure is used to compute the motion between consecutive frames. As the motion is only estimated up to scale, the distances $d_{(0,1)}$ and $d_{(1,2)}$ are arbitrary. The scales are adjusted based on the normalized distance $d_{(0,1)}$, therefore the ratio between both distances is defined by the scale parameter $S_{(1,2)}$.

B. Scale Adjustment

In order to obtain a scale consistent trajectory estimation of the camera motion, we need to calculate the scale ratios so only one global scale parameter remains unknown. This global scale defines the true size of the reconstructed 3D structure and can only be recovered if information about the real world is introduced, for instance the size of a specific portion or the structure or the distance between 2 reconstructed 3D points.

There are a number of solutions to the problem of estimating these scale ratios. A simple but expensive one is to use global optimization in which the scale is implicitly integrated in the optimized motion, for instance using Bundle Adjustment [14]. This is usually expensive as the initial guess might be far from the true scaled motion and cannot guarantee optimal results. Another very common alternative is solving the motion of the third camera using 2D-3D correspondences. A large number of solutions exist for this problem and methods can be found that use as little as 3 point correspondences. This methods however are either expensive as they are non linear and iterative or linear but yield a number of solutions from which one needs to chose. A linear method commonly used for reference is the DLT (Direct Linear Transform) algorithm where a set of linear equations is solved using SVD (Singular Value Decomposition). For the purpose of comparing our novel approach to the literature we use this algorithm as a reference, which we denote as P6P DLT [15]. Despite of not being the most advance approach it is comparable to our method as they are both linear approximations that do not require iterative steps.

P6P DLT assumes that 2D-3D correspondences exist between the world and the camera for which we wish to compute the pose. We estimate the motion between camera $i-2$ and $i-1$ ($[R|T]_{(i-2,i-1)}$) using the 8pt algorithm for its simplicity. Having obtained $[R|T]_{(i-2,i-1)}$ the 3D structure is recovered using a linear method for triangulation. The remaining step is therefore estimating the scaled pose of camera i. PnP methods [16] rely on 2D-3D correspondences and the pose is obtained optimally wrt some error measure (reprojection, reconstruction, etc). The P6P DLT approach uses the formal relation between 3D points and the 2D image features. Lets denote the points in space X_α and the points in the image x_α. Given a camera matrix $P_{(i-1,i)}$, they must comply with the relation:

$$x_\alpha = P_{(i-1,i)} X_\alpha \tag{1}$$

Where both x_α and X_α are expressed as homogeneous coordinates as:

$$x_\alpha = [s_\alpha u_\alpha, s_\alpha v_\alpha, s_\alpha]^T, X_\alpha = [a_{\alpha 1}, a_{\alpha 2}, a_{\alpha 3}, 1]^T \tag{2}$$

Note that we incorporate the scale in the image feature. Solving for s_α and substituting the system expressed by Eq. 1 we obtain 2 linear equations for each 2D-3D correspondence. 6 correspondences are needed to solve the system that can be expressed as $Aq = 0$ where q is the vector of entries of the camera matrix that we want to estimate. Applying SVD over the matrix A yields the LSQ (Least Squares) solution to the 2D-3D correspondences. This results in a camera matrix $P_{(i-1,i)}$ that integrates the rotation, translation and calibration. The translation is obtained as the right null space of $P_{(i-1,i)}$ and the rotation and calibration matrix using QR decomposition.

This approach has the disadvantage that feature matches across all three frames need to be found in order to compute the initial motion. If the camera is moving the number of matching features will decrease with the distance travelled. Additionally, the full motion is computed considering the accumulation of error of feature matches in all three images and the error in the triangulation of the 3D structure. In a camera moving along a street, we observed a typical reduction of 50% of feature matches between frames F_i and F_{i+2} with respect to the number of matches between F_i and F_{i+1}. As the number of matches is reduced, the quality in the motion estimation will also decrease.

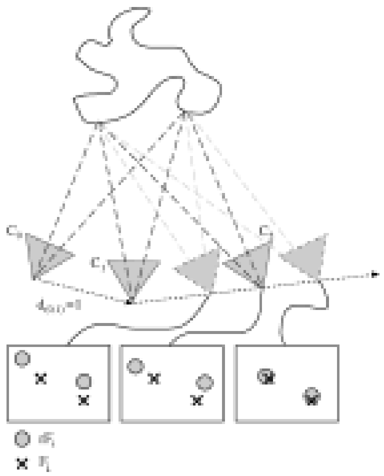


Fig. 2. Scale adjustment. The distance between cameras C_0 and C_1 is fixed to 1. The third camera C_2 can slide across the translation direction so that the reprojection error is minimized in a Least Square sense. The reprojection error is represented in the figure as the distance between the localized SIFT features F_i and the reprojected 3D points rF_i.

We employ a different approach exploiting the greater number of feature matches across consecutive frames. We calculate the motion between 3 consecutive frames using frame-to-frame feature matches. This provides two different motion estimations $[R|T]_{(i,i+1)}$ and $[R|T]_{(i+1,i+2)}$. The quality of this motion estimations is expected to be greater than motion $[R|T]_{(i,i+2)}$ due in part to the larger number of matches. These two motions are defined up to two different scale factors $S_{(i,i+1)}$ and $S_{(i+1,i+2)}$. We then explicitly calculate the ratio $s_{(i)} = \frac{S_{(i-2,i-1)}}{S_{(i-1,i)}}$ using feature matches across all 3 frames. We calculate this ratio using a linear system of equations in the same fashion as the P6P DLT algorithm with the difference that a single parameter needs to be estimated.

Given the scale free motion estimation of the second camera and the reconstructed 3D points of 3-frame matches, we can establish the relation:

$$K^{-1} x_\alpha = [R|s_i T] X_\alpha \tag{3}$$

where s_i is the scale ratio that relates the translation between cameras $i-2$ and $i-1$ and cameras $i-1$ and i. This system can be expressed as $As_i = b$ where A and b are vectors. The vector A contains one constraint per row: $[T_{\alpha x} - T_{\alpha z} u_\alpha]$ defined by one 2D-3D correspondence. The vector b is defined as: $[R_{1,1}a_{\alpha 1} + R_{1,2}a_{\alpha 2} + R_{1,3}a_{\alpha 3} - R_{3,1}a_{\alpha 1}u_\alpha - R_{3,2}a_{\alpha 2}u_\alpha - R_{3,3}a_{\alpha 3}u_\alpha]$.

We solve the system in the LSQ sense as:

$$s_i = \frac{A^T B}{A^T A} \tag{4}$$

We only need one 2D-3D correspondance to solve the scale parameter, though all available correspondences are used for robustnes.

This approach is significantly better than the P6P DLT linear solution as the error in the feature location in images $i-2$ and $i-1$ and the error in the 3D reconstruction is only propagated to the scale parameter and not to the direction of the translation nor the rotation. Experimental results are shown in Section VI.

IV. Iterative Refinement and Point Cloud Generation

Having obtained a set of scale consistent camera motions, we can reconstruct the 3D structure. We do so using a frame-to-frame approach reconstructing not only the inliers used for motion estimation but all matches whose distance is within certain threshold. We perform linear reconstruction using singular value decomposition (SVD) [12]. The reconstructed 3D structure can be noisy and not consistent, presenting duplicated structures due to the frame-to-frame reconstruction and the inaccuracy of the motion estimation (see 3). It is common in the relevant literature to perform Bundle Adjustment [14] (BA). There are two well known flavours of Bundle Adjustment: motion-only and motion-and-structure. The idea behind BA is to optimize the whole set of camera motions and alternatively also the reconstructed 3D structure with respect to a cost function. If only the motion is optimized, the camera matrices are adjusted to minimize the reprojection error of the reconstructed 3D structure given matches across frames. If both the motion and the structure are optimized, the number of parameters increases accordingly. BA is a global optimization technique that will only reach a local minimum, hence a good initial guess is usually required. Finally, the performance of BA is not only based on the initial estimation but also in the number of constraints. These constraints are defined as frame-to-frame matches. BA will perform better when feature matches are computed not only across successive frames but also with far away frames.

Since we are interested in 3D reconstruction of urban areas, the camera will typically move in the direction of a street capturing as much as possible of the urban scenery. This setup yields a smaller number of feature matches as the frame distance increases and it is usually difficult to match features with more than two frames in between. This is directly translated to the Bundle Adjustment as a lower number of constraints across frames. Due to this, BA will only reach a local minimum where the reconstructed walls are slightly separated (see Fig. 3). This separation does not have a large effect in the reprojection error due to the small angle between the rays and the fact that the features can only be matched across a few frames.

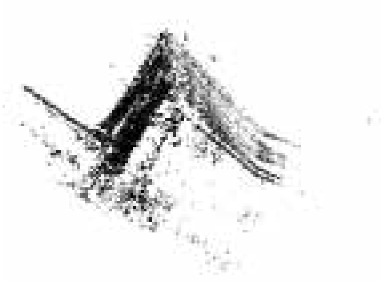


Fig. 3. Sample noisy reconstructed 3D point cloud BEFORE iterative refinement.

We employ a different approach for iterative refinement that aims at avoiding this situation. Since we assume that there exists feature matches across 3 consecutive frames, we employ an iterative refinement in which we minimize the distance between 2 corresponding point clouds (see Fig. 4).

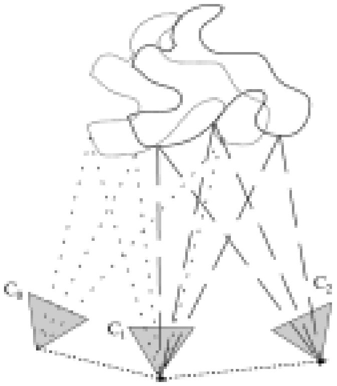


Fig. 4. We optimize the camera motions by minimizing the distance between the two reconstructed point clouds given 3-frame matches.

We need feature matches across 3 frames for computing the scale ratios, therefore, we can obtain 2 distinct point clouds for every triplet of frames. These two point clouds are the result of 3D linear triangulation using feature matches between frames F_i-F_{i+1} and frames F_{i+1}-F_{i+2}. Since this structure consist only of features matches across the three frames, we know the exact point-to-point correspondence between the two point clouds. This can be extended to the complete set of frames where we obtain 2 corresponding point clouds between every 3 consecutive frames. We use these 3D constraints to optimize the set of cameras so that the sum of distances between all corresponding 3D points is minimized. To avoid the inclusion of outliers in the minimization procedure, we only employ corresponding 3D points whose distance is below a certain threshold that we set to $1/10$th of the maximum distance between the 3D cloud and the camera center. This procedure for refinement does not require feature matches across more than 3 frames and performs well in scenes where the points lie in planes as opposed to the classical procedure where the reprojection error is minimized. In the classical approach with reprojection minimization, (all) points laying in a plane usually leads to a degenerative reconstruction. Our method does avoid this due to the direct 3D distance minimization. Figure 5 shows the refined point cloud from figure 3 after 10 iterations.

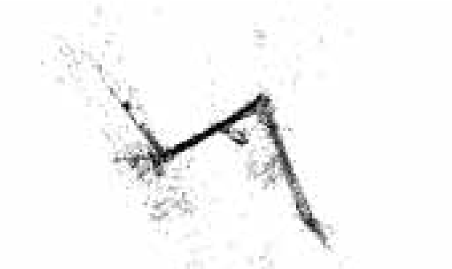


Fig. 5. Sample noisy reconstructed 3D point cloud AFTER iterative refinement.

V. 3D Modeling

Up to this point we have obtained a scale consistent and refined point cloud. We now need an efficient representation of the scene based on primitives. We chose to fit small planar patches since most urban or architectural scenes can be represented with planes. We define a planar patch as a plane limited by the convex hull over a set of points. Our goal now is to robustly fit a set of discrete planes to the point cloud while maintaining the 3D consistency and finally determine the most appropriate texture map for the planes.

A. *Robust Plane Fitting*

Werner and Zisserman [4] also employ plane fitting algorithms. First, they employ RANSAC to fit a set of planes that are consistent with the three principal directions, employing two of these directions as fixed points and using RANSAC to find the third last point. They employ this technique to find perpendicular planes present, for instance, in facades. Second, they employ the plane sweeping [17] algorithm for finding the ground plane and roofs.

Our approach is more generic because we do not make any assumptions about the scene except the fact that it can be represented by planes or planar patches. We also use RANSAC to iteratively fit planes to the point cloud. The novelty of our approach lies in the sampling method, the number of points for the plane fitting and the iterative reduction of the size of the 3D point cloud. Instead of using only three points to fit a plane and find inliers with RANSAC, we fit planes to a limited sample of the complete point cloud. We begin by randomly selecting a point in the computed 3D structure. Along with this point, the n closest points within a certain distance are also selected. Then RANSAC is used to reject outliers and fit a plane through the selected subset of points. If no plane can be fitted, the process begins by selecting a new random set. If a plane can be fitted, its convex hull is computed and the subset of selected points, including outliers, is eliminated from the point cloud. The complete procedure stops when either the number of remaining 3D points is smaller than n or a maximum number of iterations is reached. In order to improve the robustness of the algorithm, we set the minimum number of inliers to $n/2$ so planar patches of reasonable size are found. As planes are found, the 3D point cloud becomes smaller and more sparse. If we would set n too large, the procedure will only find the large planes in the structure. In order to avoid this, we employ a dynamic approach in which n is reduced by a factor 0.1 if a number of iterations pass where no plane was found. This reduction will continue until a new plane is found. This procedure compensates for the fact that the point cloud becomes more sparse as more planes are found. Setting a large n as an initial step will fit the large planes in the scene such as large portions of the facades. As the larger planes are found, the point cloud becomes more sparse and the number n is reduced, making it possible to fit patches to fewer points.

B. *Texture Mapping*

Having obtained a set of discrete planar patches that represent the reconstructed 3D structure, we improve the model by calculating the image projection into the planar patches. There exist several techniques for computing the texture map of a planar patch given a set of images and the camera poses. We employ a technique in which for every pixel in the plane, the angle between the normal of the patch and the ray between pixel and camera center is computed. We do this for every camera and the texture is obtained from the camera that yields a smaller angular difference. This produces the most frontal view of the model for every planar patch. As the objective of the final model is to freely navigate the scene in 3D, mapping the texture of the most frontal view yields a very homogeneous and natural 3D model. This technique is also robust against most occlusion situations in an urban area where the different shapes of the facades can create occlusion if the camera is looking diagonally at the facades and moving along a street. Also, reflexions are minimized due the most frontal view.

VI. Experiments and Results

In this section we present the results of our 3D modeling system in two different datasets. Both datasets were acquired with a Canon 350D using a wide angle lens (Canon EF-S 10-22mm). RAW format was used to record the images though only a quarter of the full resolution was used obtaining an image size of $1728x1151$ pixels. The images were recorded with no other aid than the camera itself. No particular attention was given to the distance between frames and only the overlap between 3 consecutive frames was considered. The camera was calibrated beforehand using Zhang [18] method and images of a black and white checkers board. Before obtaining the 3D model, images were compensated for radial distortion, which was minimal due to the quality of the lens.

A. *Scale Estimation*

In this section we present our experimental results with a fully simulated scene where 3 synthetic images are recorded. We built a virtual 3D scene where 30 3D points are created and reprojected to 3 images located at different positions. The cameras are pre-calibrated (perfect calibration) and only

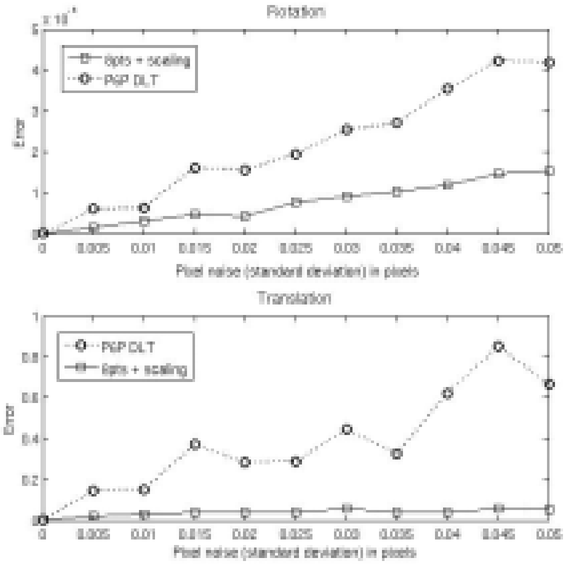


Fig. 6. Error on the estimation of rotation and translation of a third camera pose given 2 previous camera poses and the reconstructed structure using linear triangulation.

noise in the feature location is introduced. We performed 50 runs in which a different 3D scene is created and the cameras are located at a randomly selected position. For each run, we estimate the motion of camera 2 using DLT and our novel approach incrementing the noise in the feature locations from 0 to 0.05 pixels of standard deviation. For the DLT method we used 10 reconstructed features to simulate the lower number of feature matches across 3 frames. We then compare the estimated camera poses (rotation and translation) with the ground truth camera poses. The results are shown in figure 6. The plots clearly show that our method outperforms the DLT reference algorithm by a large factor. For these experiments no global optimization was used. The improvement is particularly significant in the estimation of the translation where our approach behaves more linearly than the alternative 2D-3D method which becomes almost unusable as the feature noise increases. Regarding the lower number of features for the DLT estimation, we also performed experiments in which we used the same number of features as with our approach and the results were similar to the ones presented here, only increasing the accuracy slightly due to the larger number of features. In order to reach a comparable accuracy, 4 times more 2D-3D correspondences are neccessary. This cannot be solved by just increasing the frame rate as the number of images required to obtain the same accuracy becomes too large and so does the computational costs.

B. Small Alley

This dataset consist of a collection of 8 images (see figure 7) recorded in the back of our TNO building. The recorded structure is a small alley consisting on a few intersecting planes. Part of the images are in the shadows which makes feature matching very challenging. This is a basic set meant to prove our modeling method in a small scale where multiple planes with varying orientations can be seen.

An average of 7000 image features were found on every image. Even though the visual overlap between the frames is sufficient, the large change in the viewing direction produced a considerable reduction in the number of feature matches across 3 consecutive frames. The average number of frame-to-frame matches was 2500 while the number of 3 frame matches was only 500.

Below are the parameters used for the modeling procedure:

Parameter	Value
Number of frames	8
Feature matching threshold	0.6
Ransac iter. for motion estimation	12000
Average feature matches 2 frames	2352
Average feature matches 3 frames	487
Average number of inliers	932
Max. iter. for scale finding	1000
Max. iter. for plane fitting	2500
Max. iter. for refinement	3000
Number of planes found	31
Size of point cloud	13452

Figure 7 shows a sample of the images of the dataset. The change in viewing direction is large as the camera was translating and rotating to keep the scene in the center of the image.

Figure 8 shows two views of the reconstructed 3D point cloud before (left) and after (right) iterative refinement. Our 3D minimization procedure works well even though the initial distance between corresponding 3D structures is large. No similar results were obtained with the classical reprojection error approach. This poincloud was used for obtaining the planar patches.

Figure 9 (TOP) is a representation of the planar patches that were found. For clarity, the planes have been colored in the image. It can be seen that there are 4 principal planes (4 walls and the ground). The three principal walls are reconstructed and placed in the correct position. The ground plane presents a bigger challenge since most of the surface was in the shadow which makes image feature matching very challenging and sometimes even impossible. Finally, the last wall (most right side) which is forming a slight angle with the contiguous one is also reconstructed though only the portion that is under direct sun light.

Finally, figure 9 (BOTTOM) shows a rendered image of the final planar patches based 3D model with the applied texture.

C. Suburban Area

In order to test our approach in a more challenging environment, we recorded a set of 60 images in a suburban area, covering a total distance of approximately 100 meters. The camera was directed in an angle towards the facades of the buildings while moving an approximate distance of 2-3 meter per recorded image along the direction of the street. Figure 10 shows a sample of 4 images of the complete set. The structure consist of 2 types of buildings where the main facade lays on a plane but with vertical structures with a triangular shape coming out of the buildings. These structures together with trees and cars pose a problem in texture mapping due to the occlusion problem.

An average of 10000 image features were found on every image. The overlap between images is only significant every 3 frames, posing a real challenge for standard reprojection error approaches in the iterative refinement step.

Below are the parameters used for the modeling procedure:

Parameter	Value
Number of frames	60
Feature matching threshold	0.4
Ransac iter. for motion estimation	12000
Average feature matches 2 frames	2934
Average feature matches 3 frames	785
Average number of inliers	1532
Max. iter. for scale finding	1000
Max. iter. for plane fitting	2500
Max. iter. for refinement	3000
Number of planes found	89
Size of point cloud	32318

Figure 12 (TOP) is a representation of the colored planar patches. The principal planes that represent the facades of the different buildings are correctly reconstructed. Also, the triangular structures that separate each house are also found, though due to the occlusion of cars and trees, the extent of the planar patches is not as good as in the facades. No patches are found on the ground due to the lack of texture and therefore feature matches. Figure 12 (BOTTOM) shows a rendered image of the final planar patches based 3D model with the applied texture. We show how the most orthogonal view projection model yields a natural texture map that approximates very well the true view of the street if one would look orthogonally at the facades. Also, most of the occlusions caused by the triangular shapes, the trees and the cars are eliminated and the facades can be clearly seen. In this dataset, no global iterative refinement was applied. The obtained model shows a very consistent structure were no drift can be appreciated.

VII. Discussion and Conclusion

In this paper we have demonstrated the results of our fully automated modeling system. We can summarize the novelty of our approach in three contributions. First, the scale consistent camera motion is computed with standard motion estimation algorithms where the scale is expliciteIy computed in closed form from as little as a single 2D-3D correspondence. We demonstrate how this approach offers significant advantages when compared to PnP techniques where at least 3 2D-3D correspondences are required. Second, we propose a global optimization step in which we minimize the distance of corresponding point clouds. This method performs significantly better when compared to standard reprojection minimization procedures specially when facing large point view changes or picturing planar scenes. Finally, we propose a modeling technique with an innovative sampling method in which small planar patches are fitted to the point cloud to obtain an efficient representation of the scene. Also, a natural projection is calculated to the obtained planes, reducing reflexions and minimizing the effect of occlussion caused by the shape of the facades, obtaining a very natural view of the 3D model. Our approach is more advatageous than others in literature in that it only relays on an off-the-shelf DSLR camera and the assumption that the scene can be discretized in planar patches. It also uses very fast linear algorithms that can be implemented in real time and a new global optimization technique that does not suffer from ambiguities such as the reprojection minimization approach. However, more accurate algorithms for frame-to-frame motion estimation could be explored while maintining the same approach for scale estimation, optimization and scene discretization. In our future work we plan to formally compare the error propagation in the scale consistent motion estimation between PnP linear methods and our explicit scale estimation approach[1].

References

[1] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys, "Interactive 3d architectural modeling from unordered photo collections," in *SIGGRAPH*, 2008.

[2] P. Mordohai, J. Frahm, A. Akbarzadeh, B.Clipp, C. Engels, D. Gallup, P. Merrell, C. Salmi, S.Sinha, B. Talton, L. Wang, Q. Yang, H. Stewnius, H. Towles, G. Welch, R. Yang, M. Pollefeys, and D. Nister, "Real-time video-based reconstruction of urban environments," in *ISPRS*, 2007.

[3] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collection in 3d," in *SIGGRAPH*, 2006.

[4] T. Werner and A. Zisserman, "New techniques for automated architecutral reconstruction from photographs," in *European Conference on Computer Vision*, 2002.

[5] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool, "3d urban scene modeling integrating recognition and reconstruction," in *International Journal of Computer Vision*, 2007.

[6] R. Cipolla and D. Robertson, "3d models of architectural scenes from uncalibrated images and vanishing points," in *International Conference on Image Analysis and Processing*, 1999.

[7] P. E. D. amd Camillo J. Taylor and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach," in *SIGGRAPH*, 1996.

[8] C. Fruh and A. Zakhor, "An automated method for large-scale, ground-based city model acquisition," in *International Journal of Computer Vision*, 2004.

[9] S. Agarwal, N. Snavely, I. Simons, S. M. Seitz, and R. Szeliski, "Building rome in a day," *ICRA*, 2009.

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," pp. 726–740, 1987.

[11] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *Int. J. of Computer Vision*, 2004.

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

[13] B. Horn, "Recovering baseline and orientation from essential matrix," 1990. [Online]. Available: citeseer.ist.psu.edu/horn90recovering.html

[14] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," 2000.

[15] Y. Abdel-Aziz and H. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry," *Proceedings of the Symposium on Close-Range Photogrammetry (pp. 1-18)*, 1971.

[16] M.-A. Ameller, B. Triggs, and L. Quan, "Camera pose revisited - new linear algorithms," *Rapport Interne - Equipe MOVI*, 2000. [Online]. Available: http://perception.inrialpes.fr/Publications/2000/ATQ00

[17] C. R., "A space-sweep approach to true multi-image matching," in *International Conference on Computer Vision and Pattern Recognition*, 1996.

[18] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *International Conference on Computer Vision*, 1999.

[1] All source code (and multimedia) is available online at *http://www.fit3d.info* from July 2010

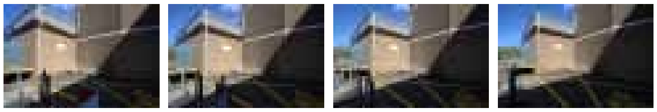


Fig. 7. Original images (4 out of 8).

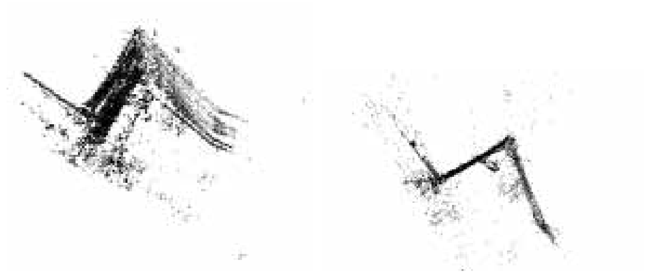


Fig. 8. Top view of the reconstructed 3D point cloud. Before refinement (LEFT) and after refinement (RIGHT).

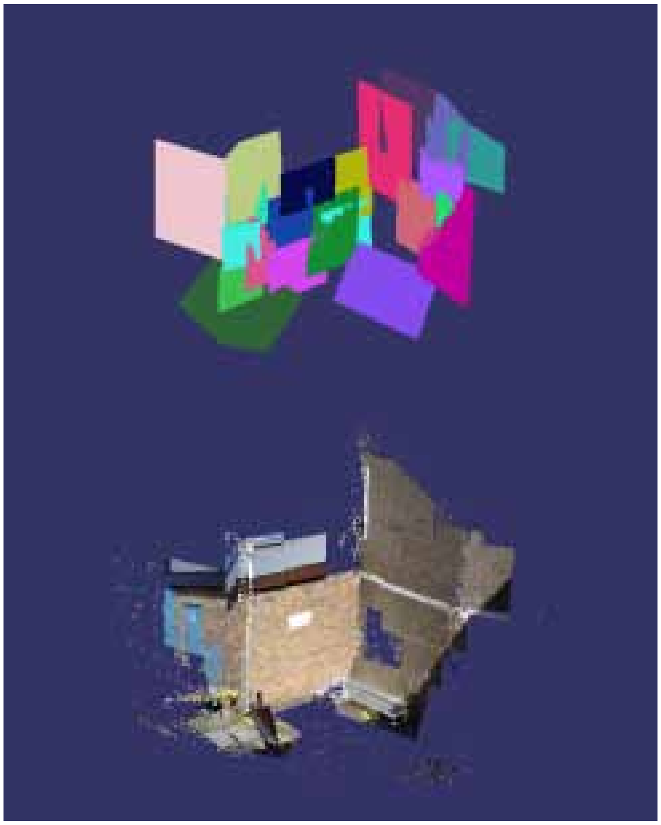


Fig. 9. TOP - Set of fitted planar patches (colored for clarity). BOTTOM - A rendered image of the textured 3D model together with the 3D point cloud.

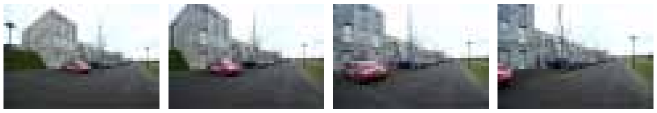


Fig. 10. Original images (4 out of 20).

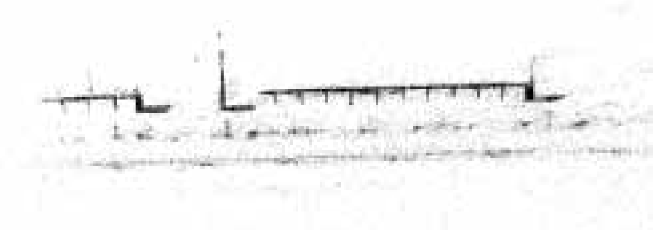


Fig. 11. Top view of the reconstructed 3D point cloud.

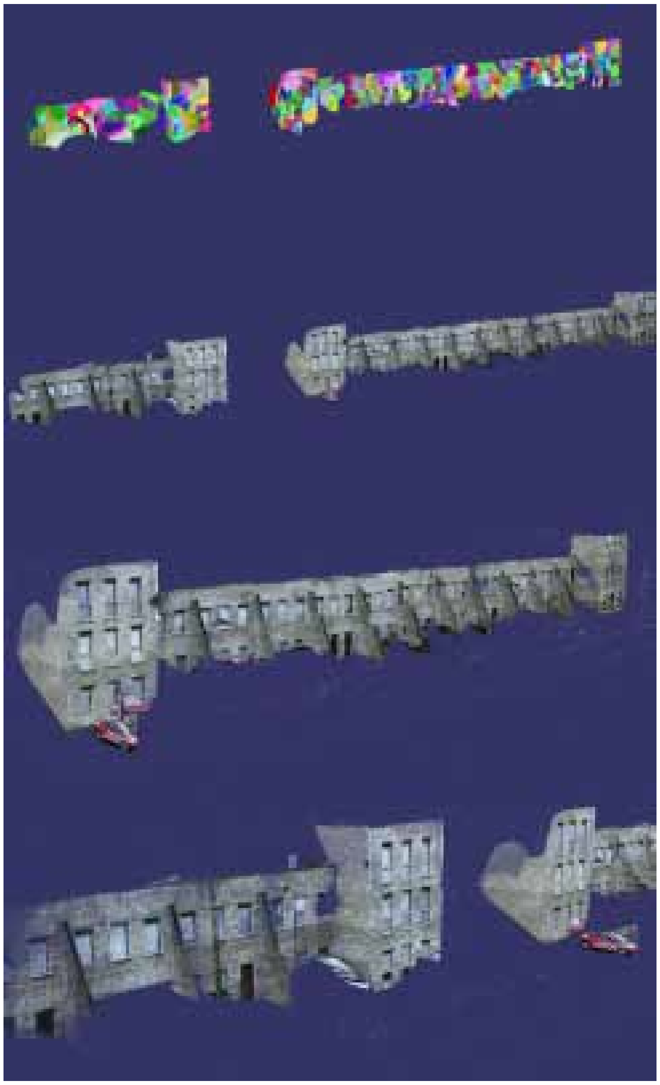


Fig. 12. TOP - Set of fitted planar patches (colored for clarity). BOTTOM - Two rendered images of the textured 3D model.