

# BUILDING RECONSTRUCTION FROM N UNCALIBRATED VIEWS

S. Cornou<sup>1,2,\*</sup>, M. Dhome<sup>1</sup> and P. Sayd<sup>2</sup>.<sup>1</sup>LASMEA, UMR 6602 du CNRS – Université Blaise Pascal – Clermont-Ferrand – FRANCE<sup>2</sup>CEA Saclay – DRT/LIST/DTSI/SLA/LCEI – Gif-Sur-Yvette – FRANCE

Commission V, WG V/4

**KEY WORDS:** Building reconstruction, bundle adjustment, uncalibrated views, constrained modeling**ABSTRACT:**

We present a supervised approach to recover 3D models of buildings from multiple uncalibrated views. With this method the user matches 3D vertices in the images and defines the 3D model of the building with the help of elementary and intuitive geometric constraints. At the same time, a graph describing relationships between vertices is built. Then, unknown parameters of this graph are estimated non-linearly through a bundle adjustment to recover the building model and the camera parameters. This method asserts that geometric rules are perfectly respected. This approach is used to recover independently 3D parts of the building with suitable images. Then all these independent 3D models are merged to obtain a full multi-scale model of the building. An example on real images is given.

## 1. INTRODUCTION

In this work, our objective is to present a semi-automatic building reconstruction method. Our main contribution is an easy-to-use modeling method based on the definition of intuitive geometric constraints. The method yields a 3D model with the smallest description in term of the number of parameters, the absolute certainty to respect the geometric rules defined by the user, and the possibility to merge easily several models to obtain a full and multi-scale global 3D building model.

3D modeling of buildings from images is an active area in computer vision. Many methods already exist and can be divided in three categories: the low-level features approaches, the primitives-based approaches and the hybrid ones that combine the two others.

The low-level features approaches describe buildings as a low-level features set. Most of the time, an automatic features matching is performed between images of a sequence to calibrate cameras and to compute the 3D structure. Features' matching has widely been studied (especially in the case of points) and is particularly efficient in the case of small baseline between images (cf. [10] [12] [13]). This method is used by M. Pollefeys [15] on video sequences, and T. Werner and A. Zisserman [20] on static images. F. Schaffalitzky and A. Zisserman address in [17] the case of widely separated and non-ordering views. Nevertheless, the model surface needs to be defined from the resulting 3D clouds of low-level features to perform a photo-realistic rendering or measurements with these approaches. Several strategies exist, Werner and Zisserman [20] automatically obtain a planar segmentation, Pollefeys [15] meshes the object with the help of a dense matching and Morris and Kanade [16] exploit image information to determine a triangulation.

In the primitives-based approach, the user provides parametric primitives to model the building. These 3D primitives are located in images. The non-linear minimisation of the distance between the primitive detected in images and the back-projected model is performed to estimate the structure and the motion of

the scene. This approach is used in [5] [9]. An advantage of such an approach is that simple geometric rules are implicit in the definition of primitives (e.g. orthogonality and length equality in a cube...). Nevertheless, this method is limited by the number of 3D elements available in the library.

Lastly, hybrid approaches try to merge the advantages of these two approaches. Cipolla and Robertson [2] present a method based on statistical estimators. Bartoli and Sturm [1] suggest a strategy in the case of multi-coplanarity constraints and, Grossman and Santos-Victor [7] implicitly describe constraints to estimate the model as an unconstrained optimisation problem. Our approach is close to the Grossman's approach except that we define a larger set of constraints, we use a bundle adjustment method that does not require an initialisation of camera motions (previously published by the authors in [3]), and our method can easily merge 3D models with different scales (e.g. an accurate window model added to a low resolution building model).

In this paper, we describe the modelization of the building. Then, the bundle adjustment algorithm is described. Finally, a real sequence is used to recover the 3D model of a castle.

## 2. CONSTRAINED MODELING

In this section, we explain our method to implement constraints in the building model. First, we give an overview of the elementary geometric constraints used. Then, we describe the full data structure with relationships between vertices and geometric constraints, and we discuss its drawbacks and its advantages. Finally, we explain how to simply merge two 3D models to obtain a more complex 3D model (with multiple scale levels for example).

### 2.1 Elementary geometric constraints

In this work, the building model is described as a cloud of 3D points. Elementary geometric rules are defined to organise and to structure this 3D cloud. This is a major difference with feature-based algorithms that recover 3D cloud from low-level

\* Corresponding author: Sébastien CORNOU, [sebastien.cornou@cea.fr](mailto:sebastien.cornou@cea.fr)

features matching without a global organisation. In our method, each new point of the model is defined with a geometric constraint in relation to already existing points (named *antecedents*). Geometric constraints and their associated equations are described in Table 1. This table indicates the equations checked for each rule and the degrees of freedom left to the new vertex. These degrees of freedom correspond to parameters that need to be estimated non-linearly with the help of 3D-images matching. The total number of unknown parameters, describing the building model, is the sum of the degrees of freedom associated to each vertex.

Name	Antecedents	Equations (constraints)	Degrees of freedom
Free point	None	$\underline{A}(x,y,z)$	3 (coordinates)
Vectorial equality	A,B,C	$\underline{CD}=\underline{AB}$	0
Parallelism	A,B,C	$\underline{CD}=\lambda\underline{AB}$	1 (1 length)
Single orthogonality	A,B	$(\underline{AC}) \perp (\underline{AB})$	2 (1 length 1 angle)
Double orthogonality	A,B,C	$(\underline{AD}) \perp (\underline{AB})$ and $(\underline{AD}) \perp (\underline{AC})$	1 (1 length)
Planarity	A,B,C	$\underline{AD}=\alpha\underline{AB}+\beta\underline{AC}$	2 (coordinates)
Distance equality	A,B,C	$\ \underline{CD}\ =\ \underline{AB}\ $	2 (2 angles)

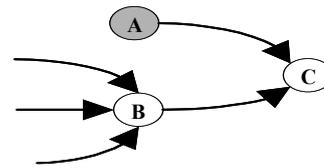
**Tableau 1.** The set of geometric constraints. The degrees of freedom indicate the number of parameters that needs to be estimated with the help of 3D-image matching to completely position 3D vertices. (Couples of bold letters correspond to vectors, the underlined letters correspond to the constrained points)

### 2.2 The global model structure

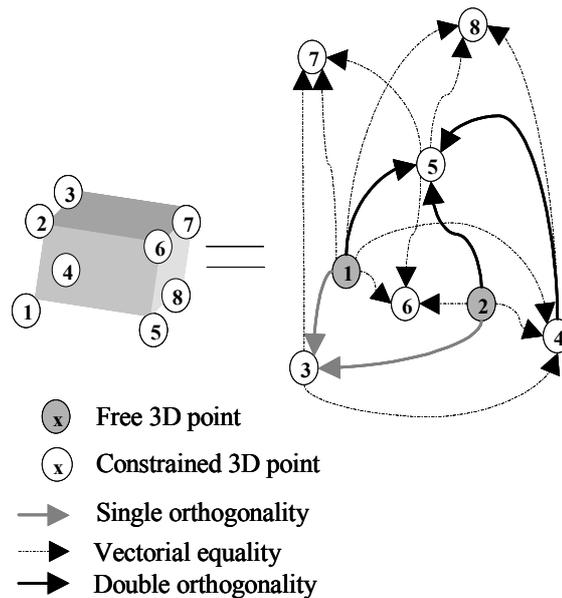
The model structure used in our method is an oriented graph. In this oriented graph, a node represents a 3D vertex, and each set of branches arriving to this node represents the geometric rules used to define this 3D vertex. The branches origins are the *antecedents* described in the previous section. One can notice that free points do not have *antecedents*, and their 3D position can directly be computed without knowing the positions of the other vertices. Figure 1 describes the relationships between vertices. If the 3D positions of A and B are known, we use the geometric rule linking C to A and B to compute the 3D position of C. The position of C depends on its associated geometric rule and on the 3D positions of its *antecedents* (A and B). Recursively, the position of B depends on the positions of its own *antecedents*... Our method yields a constraint model described with a minimum of parameters, but the 3D position of a vertex depends on the position of many others (except for free points such as A).

In figure 2, we give an example of a graph representing a rectangular parallelepiped. This graph is not unique and depends on the user description of the model. Two free points (1 and 2) are the seed of this model, and the parallelepiped model has only 9 degrees of freedom (6 for the parallelepiped poses in 3D space and 3 for the internal lengths of the model).

This graph is a description of a parametric object and is close to the primitive-based approaches.



**Figure 1.** A graph describing the links between C and its two *antecedents* A and B. B has also three *antecedents*.



**Figure 2.** Graph description of a rectangular parallelepiped. The numbers indicate the vertex references. The vertices n°1 and n°2 are the seed of this graph, each of them can be evaluated without knowledge of the others.

### 2.3 Merging of different model

In the previous section, we defined a graph description of 3D objects. To build a realistic model of a building, it is easier to design independently several models such as doors, windows, building body... Once all these models have been defined, each part has to be located in a unique framework to constitute a complete and coherent final model. Imagine that we want to complete a model A with a model B. Our goal is to define the scale factor between A and B and the rigid transformation between the A framework and the B one. This is obtained by defining three points of the model B into the model A. First, the user defines three new points in the model A. These new points are matched in images and reconstructed as any other points. Then, these 3 points are matched with vertices in the model B. This is enough to compute the scale factor and the rigid transformation between A and B. This merging approach keeps the accuracy of each model (up to a scale factor), and the model B position is geometrically constrained due to the use of geometric constraints to define the three common points (the

user can impose windows corners to be in the frontage of the building).

### 3. BUNDLE ADJUSTMENT

#### 3.1 General case

We address the case of a cloud of 3D points without constraints. Knowing some 2D projection of these 3D points in the images we try to recover the positions of the 3D points and the camera parameters (poses and intrinsic parameters). A classical answer is a non-linear optimisation method called bundle adjustment ([6] [8] [11] [18] [19] [21]). Bundle adjustment aims to minimize distance between 2D points detected in images and projection of the associated 3D points. More precisely, the criterion  $C$  to minimise is ( $i$ : 3D points index,  $j$ : cameras index,  $\text{Pr}$ : projection matrix,  $\text{int}$ : intrinsic parameters,  $f$ : projection function,  $(T_x, T_y, T_z)$ : translation vector,  $(\alpha, \beta, \gamma)$ : angle of Euler,  $\delta_{ij}$ : 0 when the 3D point  $i$  is not visible in the image  $j$ , 1 otherwise):

$$C = \min_{\substack{\text{int}_j \\ (T_x, T_y, T_z, \alpha, \beta, \gamma)_j \\ (x, y, z)_i}} \sum_{i,j} \delta_{ij} \left\| \vec{p}_{i,j}^{2D} - \text{Pr}^j(\vec{P}_i^{3D}) \right\|$$

$$\text{Pr}^j = f(\text{int}_j, (T_x, T_y, T_z, \alpha, \beta, \gamma)_j)$$

With this approach, the 3D scene is described by extrinsic ( $6N$ ) and intrinsic ( $kN$ ) parameters of  $N$  cameras and by  $P$  parameters associated to the 3D model.  $6N+kN+P$  parameters are estimated with this method.

In fact, the knowledge of the position of the 3D points and the knowledge of the intrinsic camera parameters are enough to calculate camera poses. From this observation, we suggest a new algorithm for bundle adjustment (already published in [XXX]), which hides parameters of camera poses and does not require any initialization of these parameters. Furthermore, this approach requires evaluating  $kN+P$  parameters, offers a larger convergence area, and is faster than the classical approach (previous results).

With this algorithm, we minimise the criterion  $C$ : ( $i$ : index of cameras,  $j$ : index of 3D points,  $f_2$ : camera poses estimation function,  $E$ : extrinsic matrix estimated with a pose estimation algorithm  $f_2$ ,  $f_1$ : function that expresses the intrinsic matrix in function of the intrinsic parameters,  $I$ : intrinsic matrix estimated with the function  $f_1$ ,  $\text{foc}$ : focal length,  $(x, y, z)$  3D coordinates of vertices  $P^{3D}$ ,  $\delta_{ij}$ : 0 when the 3D point  $i$  is not visible in the image  $j$ , 1 otherwise):

$$C = \min_{\substack{\text{int}_j \\ (x, y, z)_i}} \sum_{i,j} \delta_{ij} \left\| \vec{p}_{i,j}^{2D} - I^j E^j(\vec{P}_i^{3D}) \right\|$$

$$E^j = f_1(\text{int}_j, (x, y, z))$$

$$I^j = f_2(\text{int}_j)$$

In practice we use the Levenberg-Marquard algorithm to lead the non-linear optimisation and the Dementhon algorithm [4] (function  $f_2$ ) to estimate camera poses.

Some experiments has been led to evaluate performance of this approach. To compare the classic method and this new approach we have used synthetic data. The 3D scene is a cloud of free 3D points. These points have been chosen in a sphere of one distance unit radius and 5 (more for the second experiment) images have been taken around this scene. The camera model

has been reduce to a unique focal length. The principal point is supposed to be at the centre of the image and the skew parameter is equal to zero. They are no distortion.

The convergence rate and the duration of the convergence in function of the number of images have been studied for the classical approach and our new one.

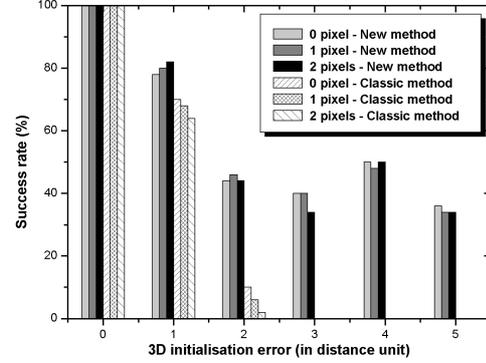


Figure 3. Convergence rate in function of the error on the 3D initialisation.

The figure 3 gives the convergence rate in function of the initial error on the 3D cloud of points (it explain how far from the solution the initialisation of the 3D shape is done). The Dementhon algorithm is used to define initial image poses. We can notice that close to the solution the 2 methods converge, then when the noise increase (close to the sphere radius value) the convergence rate tumble down to zero for the classical approach and fall to 40% for the new approach. We can give a first but incomplete explanation it is that the new approach evaluate a new pose for each image at each iteration while the classic method conditioned the next pose by the previous one (has rotation and translation are non-linearly estimated).

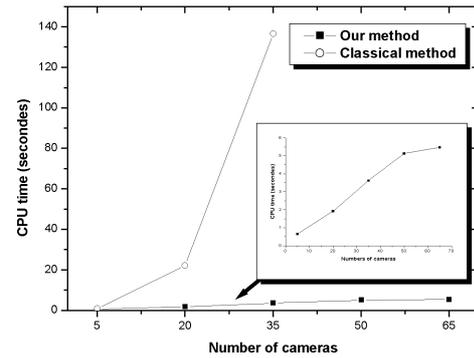


Figure 4. Computing time in function of the number of images.

We have show previously that our approach hide the pose parameters. The figure 4 compares the computing time in function of the number of image. The benefit is evident!

### 3.2 Constrained model estimation

In the previous section, we gave an overview of our bundle adjustment algorithm in the case of unconstrained cloud of 3D points. In the specific case of constrained models, the bundle adjustment is more complicated. The measurements used are the projection of 3D vertices located in images. The parameters estimated are the degrees of freedom existing in the building model. There are no direct relationships between the unknown parameters and the measurements such that a modification of one parameter can modify many measurements. This is due to the constraints introduced in the model, and the consequence in the optimisation process is a non-sparse and non-diagonal Jacobian matrix (a contrario, in the case of unconstrained cloud of 3D points the Jacobian matrix is sparse and closed to a diagonal form).

Nevertheless, the optimisation process searches the vector of parameters that minimised the sum of distances between back-projected vertices and their 2D projections in images. With our modelization, the geometric constraints are always perfectly respected, this yields a reduction of the number of parameters associated to the building model but this introduction of hard constraints modifies the underlying topology of the non-linear criterion (defined section 3.1) and can generate local minima.

Experimentally, this method converges to a good solution when the model is well balanced. Nevertheless, if we try to reconstruct a multi-scale model (e.g. the body of a building and a window), we meet difficulties because some parts heavily weight due to the large number of measurements available (window), while others are neglected (frontage (only 4 corners)). Our solution is to build each element independently, using adapted images, and to merge them in a final model.

## 4. BUILDING RECONSTRUCTION FROM A REAL SEQUENCE

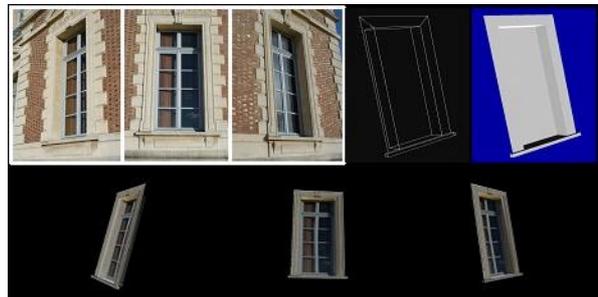
The “Château de Sceaux” has been chosen as an example. It is a French XVII century castle. Some specific elements have been selected. 22 photographs of these elements have been taken with a digital camera and 2 focal lengths 28 mm and 135 mm have been used. The image resolution is 2000x3008 pixels, the distortions have not been corrected and the focal lengths are unknown. The photographs have been taken from the ground and no information on the camera poses is available. If we look in detail at the castle elements, it appears a wide variety of details (windows, gutters, bas relief...). The user has to choose the details he wants to reconstruct (in function of the needs), because it would be too costly to reconstruct each detail of the building. Nevertheless, it is always possible to complete the model later by adding new part of the building reconstructed with a new set of images (section 2.3).



**Figure 5.** Images of the sequences describing each selected elements of the castle.

### 4.1 The reconstruction process and the results

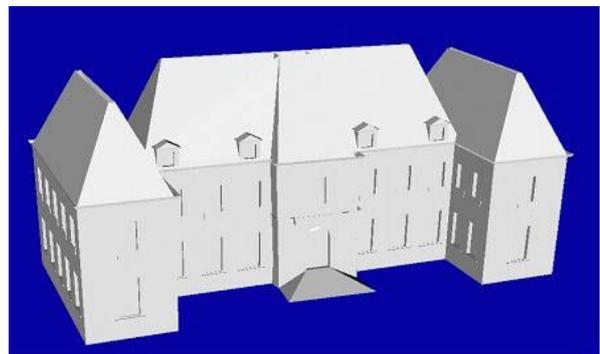
Here, we present the reconstruction of the castle body, the downstairs and upstairs windows, one sort of dormers windows, and the central advance of the frontage (figure 5).



**Figure 6.** Example of the downstairs window. The 3 initial images are on the top-left. On the top-right, there are the wire-frame and the surface model. On the bottom, the final model of the windows with accurate texture.

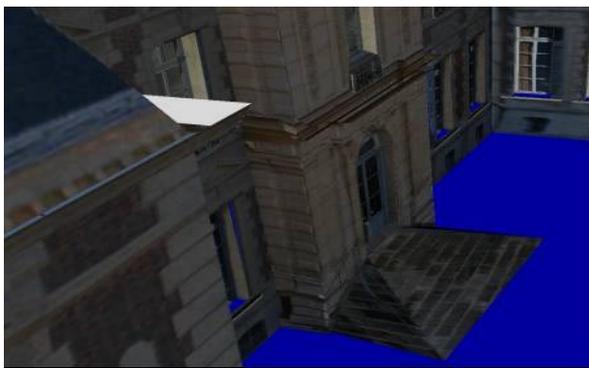
Each element has independently been reconstructed from adequate images. The figure 6 presents the example of a downstairs window. With only three images available for downstairs windows, we define a constrained model using our graph description (section 2) and we apply our bundle adjustment method (section 3) to recover the 3D structure and the camera parameters. After a manual surface definition (see [14] for an automatic extraction algorithm), textures have been extracted from the images. This process has been applied to each element.

Then, all the elements have been merged together to obtain a full model of the castle. For example, to locate the downstairs windows, window corners have been defined in the frontage with geometric constraints and reconstructed with the help of images (the body ones). These windows are located up to the castle body resolution, but their proportion and their texture have previously been defined with close-range images. For repetitive elements, unique models have been used. This merging process has been applied to each detail and the results are shown in figures 7 and 8.



**Figure 7.** Global model without texture. All the specific elements have been merged to obtain a multi-scale model.

Two days were necessary to obtain these results and the 2D RMS error was around 8 pixels. Difficulties to detect the buildings corners (hidden by gutters...) and the non correction of distortion may explain this RMS error. Nevertheless, with such a stratified approach, we have been able to recover a complete model while conserving the accuracy of each element (up to a scale factor) and extracting the texture from the better images available. Furthermore, we can improve the model with new elements, if needed, and we can adapt the reconstruction time and the accuracy to the need.



**Figure 8.** Final textured model. We can notice the high quality texture on windows and the balustrate in front of the upstairs windows.

#### 4.2 Texture extraction

The extraction of texture from images to obtain a realistic 3D model has been obtained with a simple algorithm. As we know a surfaced 3D model of the building and the position of each image we extract the real color of part of the building while taking into account the occultation. This ray-tracing texturing process is time consuming as it needs to detect each occultation. Nevertheless, the resulting texture is as good as possible for a given set of images. The texture of each part of the building are extracted for dedicated images and the result is a building model with multiscale resolution for length and texture. Such an approach allows to introduce data in a full 3D model. The user can reconstruct the interesting part of the building with high accuracy and a very detailed texture (in the case of the Sceaux castle we can see the nail in the wood of the windows) while obtaining a low resolution model of the other parts.

#### 5. CONCLUSION

In this study, a new approach for building reconstruction has been presented here. We suggest a method using constraints on 3D points features with simple and intuitive geometric rules. This result is an easy-to-use tool that offers the flexibility of low-level features approaches and the modularity of primitive-based methods. Moreover, a new bundle adjustment approach (without camera poses initialisation) has been used to estimate these models. Finally, this method has successfully been applied to a real sequence, and a multi-scale model of the "Château de Sceaux" obtained.

Further work will increase the quality of the camera calibration with the help of an automatic interest features matching based on this initial reconstruction. The merging step will also be updated to obtain a better accuracy.

#### REFERENCES

- [1] A. Bartoli and P. Sturm. *Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene*, International Journal of Computer Vision 52(1), 45-64, 2003.
- [2] R. Cipolla and D.P. Robertson. *3D Models of architectural scenes from uncalibrated images and vanishing points*. In Proc. IAPR 10th International Conference on Image Analysis and Processing, pages 824--829, Venice, 1999.
- [3] S. Cornou, M. Dhome and P. Sayd. Bundle adjustment: a fast method with weak initialization. BMVC 2002, Cardiff.
- [4] L.S. Davis and D.F. Dementhon. *Model-based object pose in 25 lines of code*. International Journal of Computer Vision, 15(2):123-141, 1995.
- [5] P. E. Debevec, C. J. Taylor, and J. Malik. *Modeling and Rendering Architecture from Photographs: A Hybrid Geometry and Image-Based Approach*. In SIGGRAPH 96, August 1996, New Orleans.
- [6] O. Faugeras, Q-T. Luong, *The geometry of Multiple Images: the laws that govern the formation of multiple images of a scene and some of their applications*. ISBN 0-262-06220-8, The MIT Press, 2001.
- [7] E. Grossmann and J. Santos-Victor. *Maximum likelihood 3D reconstruction from one or more images under geometric constraints*. BMVC 2002, Cardiff.
- [8] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge university press, 2000.
- [9] F. Lang, and W. Förstner. *3D-city modeling with a digital one-eye stereo system*, In Proceedings of the XVIII ISPRS-Congress, 1996, Vienna, Austria.
- [10] M. Brown and D. G. Lowe, *Invariant features from interest point groups*, British Machine Vision Conference, BMVC 2002, pp. 656-665, Cardiff.
- [11] E. Malis, A. Bartoli. *Euclidean Bundle Adjustment Independent on Camera Intrinsic Parameters*. Research report n°4377. INRIA – ISSN 0249-6399 - December 2001.
- [12] J. Matas, O. Chum, M. Urban and T. Pajdla, *Robust Wide Baseline Stereo from Maximally Stable Extremal Regions*, British Machine Vision Conference, BMVC 2002, Cardiff.
- [13] K. Mikolajczyk and C. Schmid. *An affine invariant interest point detector*. In European Conference on Computer Vision, vol. 1, 128--142, 2002.
- [14] D.D. Morris and T. Kanade. *Image-consistent surface triangulation*. Computer Vision and Pattern Recognition (CVPR 2000), IEEE Computer Society, Vol. 1, June, 2000, pp. 332-338.
- [15] M. Pollefeys. *Tutorial on 3D modelling from images*, in conjunction with ECCV 2000, 26 June 2000, Ireland, Dublin.
- [16] C. Rother, *Multi-View Reconstruction and Camera Recovery using a Real or Virtual Reference Plane*, Doctoral Dissertation, ISBN 91-7283-422-6, KTH, Stockholm, Sweden, Januari, 2003.

[17] F. Schaffälitzky and A. Zisserman. *Multi-view matching for unordered image sets, or "How do I organise my holiday snaps?"*, 7th ECCV, 2002, Copenhagen.

[18] Steedly, Essa. *Propagation of Innovative Information in Non-Linear Least-Squares Structure from Motion*, In Proceedings of International Conference on Computer Vision (ICCV 2001), Vancouver BC, Canada, July 2001.

[19] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon. *Bundle adjustment – A modern synthesis*. In W. Triggs, A. Zisserman and R. Szeliski, editors, *Vision algorithms: Theory and practice*, LNCS, pages 298-375. Springer Verlag, 2000.

[20] T. Werner and A. Zisserman. *New Techniques for Automated Architecture Reconstruction from Photographs*. Proc. 7th European Conference on Computer Vision, Copenhagen, Denmark.

[21] Z. Zhang and Y. Shan. *Incremental Motion Estimation Through Local Bundle Adjustment*. Technical Report MSR-TR-01-54, Microsoft Research, Microsoft Corporation One Microsoft Way, Redmond, WA 98052, May 2001.