Semiautomatic generation of semantic building models from image series

Stefan Wirtz, Peter Decker, Dietrich Paulus

Active Vision Group, University of Koblenz-Landau,
Universitätsstr. 1, 56070 Koblenz, Germany

ABSTRACT

We present an approach to generate a 3D model of a building including semantic annotations from image series. In the recent years semantic based modeling, reconstruction of buildings and building recognition became more and more important. Semantic building models have more information than just the geometry, thus making them more suitable for recognition or simulation tasks. The time consuming generation of such models and annotations makes an automatism desirable. Therefore, we present a semiautomatic approach towards semantic model generation. This approach has been implemented as a plugin for the photostitching tool Hugin*. Our approach reduces the interaction with the system to a minimum. The resulting model contains semantic, geometric and appearance information and is represented in City Geography Markup Language (CityGML).

1. INTRODUCTION

It is widely accepted that knowledge of the object domain is needed to classify objects. Therefore, we need precise and detailed models. Geoapplications like Google Earth provide an increasing number of building models, which may contain semantic annotations like facades, windows, doors and so forth. Admittedly, models of geoapplications often contain few details. If they contain many details, usually the concentration in generating these models will lay on the appearance of the model and not on the correctness of the geometry. For that reason, the generation of accurate surface models combined with semantic annotations is still important.

In principle, there exists two kinds of systems to generate models (see section 2). On the one hand, full automatic systems which have to set limitations to the shape of the buildings to be modeled. They do usually not achieve the precision in modeling, which is needed to work with the resulting models in an adequate way. On the other hand, full manual systems which generates precise and detailed models, but this generation is time consuming.

An approach for the generation of detailed and precise models containing semantic with less effort is still missing. Therefore, we present a semiautomatic approach for 3D semantic model generation (see figure 3). The following steps are necessary: image acquisition, feature detection, pose estimation, 3D reconstruction, semantic interpretation/annotation and export in a suitable format. The resulting model contains semantic, geometric and appearance information and is represented in City Geography Markup Language (CityGML). CityGML is a common information model for the representation of sets of 3D urban objects. It includes semantic, geometric, topological and appearance information simultaneously; and because of its XML base, it is a good interchange format. CityGML use the OGC standard *Geography Markup Language (GML)*[†12] for geometry representation.

The paper is organized as follows. We introduce the related work in Section 2. Section 3 describes our approach of generating models semiautomatically in detail. We show and discuss our experiments and results in Section 4. A summary and ideas for future work can be found in Section 5.

Further author information: (Send correspondence to wirtzstefan@uni-koblenz.de)

*http://hugin.sourceforge.net

†http://www.opengeospatial.org/standards/gml

2. RELATED WORK

Several groups have worked with different research focuses on reconstruction of buildings and urban scenes from images during the last decade.

Debevec et al.[13] introduce a photograph-based approach for modeling and rendering architectural scenes. The modeling approach combines both geometry-based and image-based modeling techniques. Their system requires user interaction and provides typical geometric primitives applicable for architectural scenes. These primitives are use to generate the surface model. Additionally, the model is textured by mapping the photos on the surfaces.

Lee et al.[3] propose an effective 3D method incorporating user assistance for modeling complex buildings. This method utilizes the connectivity and similar structure information among unit blocks in a multi-component building structure, to enable the user to incrementally construct models of many types of buildings. The system attempts to minimize the time and the number of user interactions needed to assist an existing automatic system in this task. They use aerial images to construct models covering wide areas.

Additionally, Lee et al.[2] present a method for automatically integrating ground view images and 3D models to obtain high resolution facade textures for 3D architectural models. They propose a hybrid feature extraction method, which exploits a global line clustering based on the Gaussian sphere and a local line clustering based on a region-based segmentation approach.

Furthermore, Lee et al.[6] introduce a method for reconstructing the 3D building windows from a single calibrated ground view image. They extract windows automatically in the rectified image using a profile projection method, which exploits the regularity of the vertical and horizontal window placement and classify the extracted windows using the image texture information.

Dick et al.[10] describe a framework for automatically determining the structure and identifying a piece of architecture from a small number (2-6) of ground view images. The result is a labeled (window, door, column, roof, stairs, etc.) 3D surface model.

Mayer and Reznik[7] propose an approach for building facade interpretation ranging from uncalibrated images of an image sequence to the extraction of objects such as windows.

Rusu et al.[15] present a system for real-time 3D mapping using point cloud data from stereo. The locally acquired point cloud views are registered in a consistent global frame and transformed into a polygonal representation. Based on the geometric properties of each polygon, semantic annotations (Ground, Level, Vertical, Steps, Unknown) are added.

3. MODEL GENERATION

Our approach, the *SemanticModeler3D (SeMo)*, is able to generate a 3D model, similar to a CAD model, with reasonable semantic annotations. The method consists of four phases: (i) Image acquisition of the building; (ii)

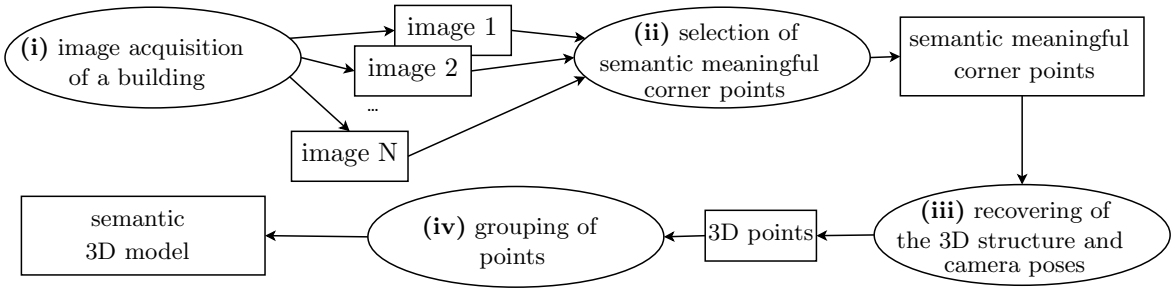
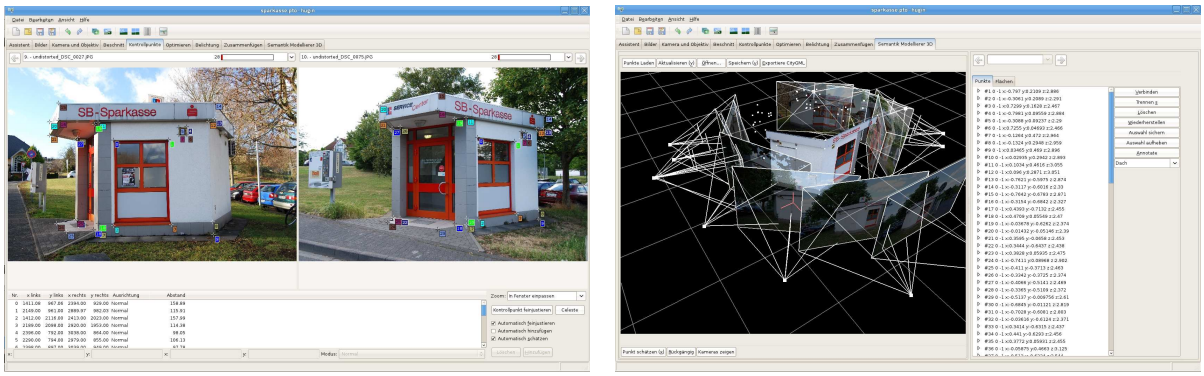


Figure 1. Data flow diagram of the procedure of SeMo.

Selection of semantically meaningful corner points, which are visible in at least two images; (iii) Recovering of the 3D structure and camera poses using structure from motion techniques; (iv) and the grouping of points, which form a ground-, facade-, roof-, door- or windowsurface. Figure 1 gives an overview of the procedure.



(a) (b)

Figure 2. Point correspondences in Hugin with automatic position refinement achieving subpixel accuracy (a) and annotations of semantic with our Hugin extension (b). Also the poses of the cameras with their corresponding image are visualized.

The basis of our approach is the photostitching tool Hugin. With Hugin it is possible to assemble a mosaic of photographs into a panorama and to stitch series of overlapping images. We use Hugin, because it offers comfortable visualization and editing functions, for example a SIFT-based mechanism for point annotations. After selecting a point in an image the SIFT-based annotation routine helps to find the corresponding point annotation in the other image. These point annotations achieve a subpixel accuracy, which improves the quality of the generated 3D models and the annotation speed. The approach works as follows:

At first we have to take overlapping pictures of the building which we want to reconstruct in 3D. Before we can use the images for the model generation, we undistort them. Therefore, we calculate the intrinsic camera parameters: the focal length, the pixel size on the sensor chip, the skew parameter, the position of the principal point in the image plane (in pixels) and the radial distortion coefficients. Using the radial distorion parameters, we compensate the distortion which results from the camera lens. The other parameters are encapsulated in the calibration matrix $\boldsymbol{K}$. An adequate Algorithm for intrinsic camera calibration is the algorithm of Zhang,[9] which uses a simple and planar calibration pattern with known geometry.

In Hugin we select point correspondences from the undistorted image series, which are important for the geometry of the building (see figure 2(a)). Next, we take the corresponding points from the first two images and determine the 3D world points using the eight point algorithm and triangulation[4] (see (i) and (ii) in figure 1).

The fundamental eight point algorithm delivers a fundamental matrix $\boldsymbol{F}$. Because the features were selected manually, we do not need to worry about outliers.

Using the previously acquired calibration matrix $\boldsymbol{K}$ for our camera, we can extract the essential matrix $\boldsymbol{E}$ from the fundamental matrix: $\boldsymbol{E} = \boldsymbol{K}^{\mathrm{T}}\,\boldsymbol{F}\,\boldsymbol{K}$. The essential matrix is then decomposed into a rotation and a translation up to a scalar factor: $\boldsymbol{E} = \lambda_E \cdot [\boldsymbol{t}]_\times\,\boldsymbol{R}$. The rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ describe the relative orientation and position of the second camera with respect to the first camera, assuming the first cameras projection matrix to be $\boldsymbol{P}_1 = \begin{pmatrix}\boldsymbol{I} & \boldsymbol{0}\end{pmatrix}$. The factorization of $\boldsymbol{E}$ is not unique but yields a translation vector with unknown sign, as well as four possible rotation matrices. Two rotation matrices can be discarded since their sign would lead to $det(\boldsymbol{R}) = -1$. From the remaining four possible solutions of $\boldsymbol{R}$ and $\boldsymbol{t}$ we have to choose the correct combination by determining the case in which a single triangulated point (see equation (1)) lies in front of both cameras.[11]

The resulting projection matrix of the second camera is: $\boldsymbol{P}_2 = \begin{pmatrix}\boldsymbol{R} & \boldsymbol{t}\end{pmatrix}$. Having obtained both projection matrices of the cameras this way, we can reconstruct the 3D coordinates of a world point $\tilde{\boldsymbol{p}}^{\mathrm{w}}$ from its projections $\tilde{\boldsymbol{p}}^{\mathrm{i}}$ and $\tilde{\boldsymbol{q}}^{\mathrm{i}}$ by doing a triangulation with minimal squared backprojection error:

$$\tilde{\boldsymbol{p}}^{\mathrm{w}} = \underset{\tilde{\boldsymbol{p}}^{\mathrm{w}}}{\operatorname{argmin}} \left\|\boldsymbol{P}_1\,\tilde{\boldsymbol{p}}^{\mathrm{w}} - \tilde{\boldsymbol{p}}^{\mathrm{i}}\right\|^2 + \left\|\boldsymbol{P}_2\,\tilde{\boldsymbol{p}}^{\mathrm{w}} - \tilde{\boldsymbol{q}}^{\mathrm{i}}\right\|^2 \tag{1}$$

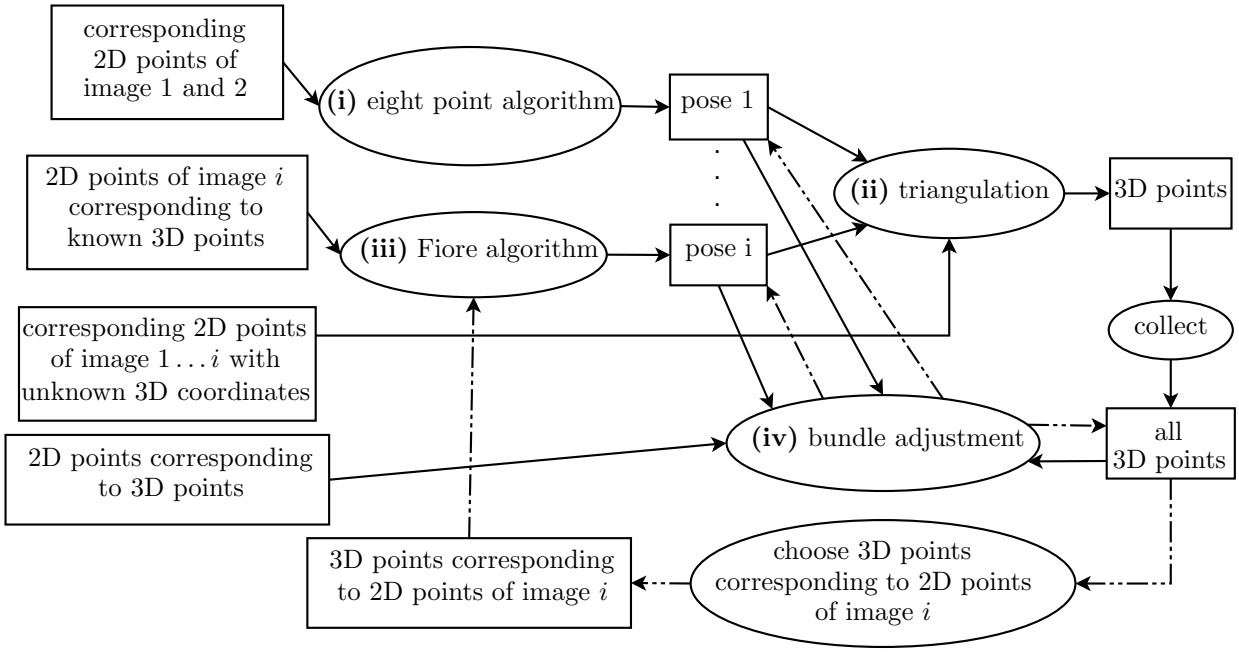


Figure 3. Data flow diagram of the 3D point generation.

A point in world coordinates is marked with a "w" in the exponent and a point in image coordinates is marked with an "i". The tilde denotes that the point is described in homogeneous coordinates. This equation can be linearized with respect to $\tilde{\boldsymbol{p}}^{\mathrm{w}}$ and solved in a single step using singular-value decomposition (SVD) to compute the optimal $\tilde{\boldsymbol{p}}^{\mathrm{w}}$.

The initial model is expanded iteratively using one new image at a time. Therefore, we estimate the camera pose of the next image using the the correspondences between 2D features in the new image and the already known 3D world points employing the approach of Fiore[8] (see (iii) in figure 3). Fiore assumes that we know the three-dimensional world points $\boldsymbol{p}_i^{\mathrm{w}}$ and, also, the corresponding camera image point $\boldsymbol{p}_i^{\mathrm{i}}$ locations. We seek the optimal translation vector $\boldsymbol{t}$, rotation matrix $\boldsymbol{R}$, scale λ, and projective parameters $\boldsymbol{l}_i$ that best satisfy

$$\boldsymbol{l}_i \tilde{\boldsymbol{p}}_i^{\mathrm{i}} = \lambda \boldsymbol{R}(\boldsymbol{p}_i^{\mathrm{w}} + \boldsymbol{t}), \quad i = 1, \dots, N. \tag{2}$$

For notational ease, we define the data matrices $\boldsymbol{D}_1 = [\tilde{\boldsymbol{p}}_1^{\mathrm{w}} \dots \tilde{\boldsymbol{p}}_N^{\mathrm{w}}]$. We can show that

$$\boldsymbol{l} = {\boldsymbol{D}_1}^{\mathrm{T}} \boldsymbol{\alpha}, \tag{3}$$

for an unknown 4×1 vector $\boldsymbol{\alpha}$. It requires $N \geq 6$ points to find a unique minimal $\boldsymbol{\alpha}$. Denoting the left-hand sides of equation (2) as $\boldsymbol{b_i} = \boldsymbol{l}\tilde{\boldsymbol{p}}^{\mathrm{i}}$, equation(2) becomes

$$\boldsymbol{b_i} = \lambda \boldsymbol{R}(\boldsymbol{p}_i^{\mathrm{w}} + \boldsymbol{t}), \quad i = 1, \dots, N. \tag{4}$$

Letting $\widehat{\boldsymbol{b}}_i$ and $\widehat{\boldsymbol{p}^{\mathrm{w}}}_i$ the points shifted to their centroids, we solve for the optimal least-squares scale λ using

$$\lambda = \frac{\sum_i^N \left\| \widehat{\boldsymbol{p}^{\mathrm{w}}}_i \right\| \left\| \widehat{\boldsymbol{b}}_i \right\|}{\sum_i^N \left\| \widehat{\boldsymbol{p}^{\mathrm{w}}}_i \right\|}. \tag{5}$$

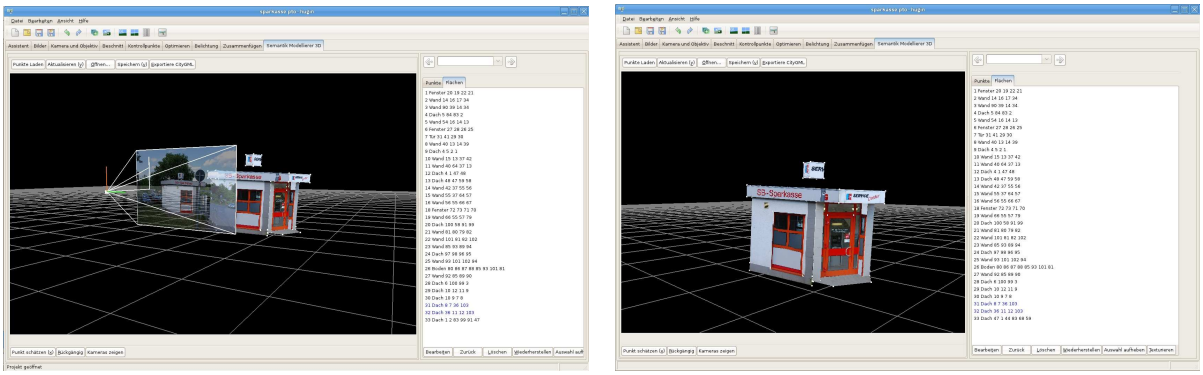
Let $\boldsymbol{B}$ be the $3 \times N$ matrix formed by stacking the points $\widehat{\boldsymbol{b}}_i$ side by side and $\boldsymbol{D_2}$ be the matrix formed by stacking the scaled points $\widehat{\boldsymbol{p}^{\mathrm{w}}}_i$ similarly. We would like to minimize $\sum_{i=1}^N \left\| \widehat{\boldsymbol{b}}_i - \lambda \boldsymbol{R} \widehat{\boldsymbol{p}^{\mathrm{w}}}_i \right\|_2^2 = \| \boldsymbol{B} - \boldsymbol{R}\boldsymbol{D_2} \|$, where the solution is known as

$$\boldsymbol{R} = \boldsymbol{V}_R {\boldsymbol{U}_R}^{\mathrm{T}}. \tag{6}$$

$\boldsymbol{U}_R$ and $\boldsymbol{V}_R$ are the left and right singular vectors of the SVD $\boldsymbol{U}_R\boldsymbol{S}_R\boldsymbol{V}_R$ of the matrix $\boldsymbol{D_2}\boldsymbol{B}^{\mathrm{T}}$.

The equations 3-6 provide the opportunity to calculate the camera pose. With the camera pose, given by the algorithm of Fiore, we are able to calculate new world points from the correspondences between the current and any other image with known position. In each iteration, we minimize the reprojection error by optimizing the position of the points in 3D space and the position and orientation of the observing cameras using bundle adjustment[14] (see (iv) in figure 3).

The result is a set of 3D points which contribute to the geometric structure of the building or have a semantic meaning. To generate a 3D model enriched with semantic annotations, we select groups of 3D points and assign a label to the group, like ground-, facade-, roof-, door- or windowsurface. Default labels which correspond to most of the semantic objects of CityGML are given. The annotation labels are free to choose, to exchange and to remove, so that it is possible to adapt the system to other objects as well. These labels have usually no correspondence to CityGML and are handled as an unspecified CityObject.



(a) (b)

Figure 4. Semantic annotations with textures embedded in our Hugin extension. It is possible to select cameras separately (a) and watch through the taken image from the camera position to simplify the annotation and check for correctness.

For each annotated surfaces the system choses an image, cuts the surfaces, calculates the homography and displays the texture on the surface (see figure 4).

After the annotation phase, we are able to export the semantic annotations and the associated surfaces along with their texture information to the exchange format CityGML, an established standard for building representation. Using the approaches of Falkowski et al.,[1,5] we are able to enrich the model with information about the relations between the semantic and geometric objects. Additionally, it is feasible to transform these information to COLLAborative Design Activity(COLLADA)[‡], Object-Oriented Graphics Rendering Engine (Ogre)[§], and to the STOR urban model[¶], another graph-based representation, which we use among others for building recognition. The whole system is integrated in Hugin, for that reason it is possible to make changes in any working step at any time without loosing any information.

4. EXPERIMENTAL EVALUATION

Without reliable data, the true deviation of the generated model to the original building can only be guessed. Especially, in the context of modeling buildings reliable 3D data is difficult to obtain. Therefore, we use a model house of scale 1:87, where we are able to measure all surfaces. With these measurements we manually generated a 3D model of the model house which serves as ground truth (see figure 5).

For the model generation with our system SeMo, we take 14 images of the model house and annotate overall 388 point correspondences. Figure 6 shows a subset of the image series used for the generation of the 3D model.

[‡] http://www.khronos.org/collada/
[§] http://www.ogre3d.org/
[¶] http://www.uni-koblenz-landau.de/koblenz/fb4/institute/uebergreifend/er

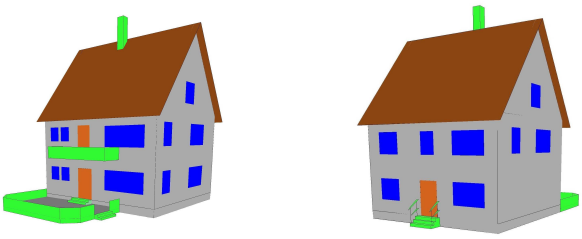
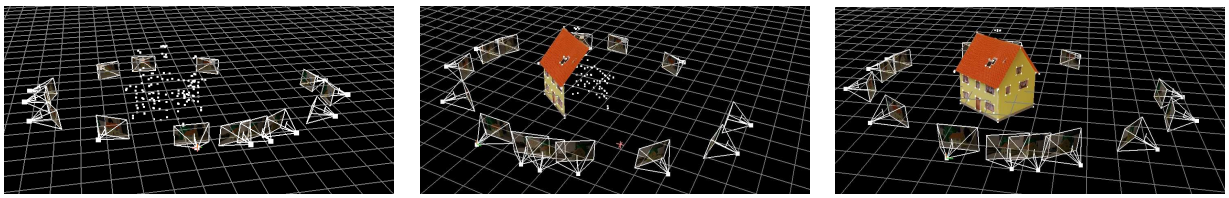


Figure 5. 3D Model of the 1:87 scale model of an one family house (see figure 6). The colors are not meant as texture but rather as differentiation between window-, door-, wall-, roof- and outer building installationsurfaces.

Figure 6. Image series of a 1:87 scale model of an one family house.

From the annotated point correspondences we obtain a set of 3D points which contribute to the geometric structure of the building or have a semantic meaning (see figure 7(a)). These points were then grouped to surfaces manually (see figure 7(b) and 7(c)).



(a) (b) (c)

Figure 7. 3D pointcloud generated of the annotated point correspondences. Additionally, the poses of the cameras are marked. If surfaces are annotated, they are visualized with their texture.

As evaluation, we compare the generated 3D model with the ground truth model. Therefore, we have to rotate, translate and scale the object coordinate system of the generated model to the object coordinate system of the ground truth model. The point-to-point error e is than given by:

$$e = \underset{\lambda \boldsymbol{R}\boldsymbol{t}}{\operatorname{argmin}} \sum_{i}^{N} \left\| \boldsymbol{q}^{\mathrm{ob}} - \lambda \boldsymbol{R}(\boldsymbol{p}^{\mathrm{ob}} - \boldsymbol{t}) \right\|. \tag{7}$$

In so doing, e denotes the point-to-point error between the models in cm, $\boldsymbol{q}^{\mathrm{ob}}$ are the 3D points of the ground truth model in object coordinates and $\boldsymbol{p}^{\mathrm{ob}}$ are the 3D points of the model generated with SeMo in the object coordinate system of this model which differs from the object coordinate system of the ground truth model. The average error $\overline{e} = \frac{1}{N}e$ amounts to 0.1269 cm per 3D point. This is an acceptable value, particularly, concerning that the measurements of the ground truth model have an accuracy in the range of millimeters, consequently the precision of the measurement can deviate abound one millimeter.

The evaluation of the pose estimation of the cameras can only be made quantitative in the way that a camera pose is approximately correct or not. In our case 100% of the camera poses are correct. Nevertheless, choosing point correspondences for the first two images needs some experiences to get a first correct pose estimation. The more images with correspondences we get the better becomes the quality of the 3D points and the pose estimations of the cameras.

It is also possible to use the system for generating 3D interior views of buildings (see figure 8). An imaginable application is the 3D visualisation of the room layout including annotations for electric sockets, windows, doors and walls for furnish planning.

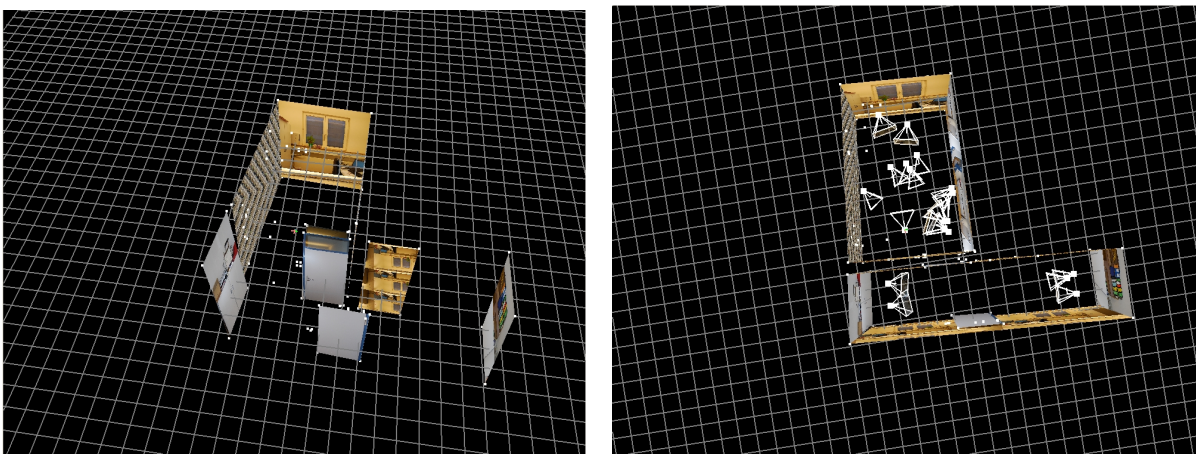


Figure 8. Room layout of an office with corridor.

The automatic choice of images for the texture works very well in the case of building modeling, because usually nearly the entire building is visible on the image. However, for interior views it is often impossible to find an image where the whole surface is visible (see texture error in figure 8), because most images show only small parts of the rooms.

5. CONCLUSION

We presented the approach SeMo3D embedded in the photostitching tool Hugin, which is able to generate a 3D model of a building with semantic annotations from an images series. Of course, it is possible to annotate any object, where reasonable points can be selected. Our approach reduces the interaction with the system to a minimum on the supposition that reliable models are needed. The average error is less than two millimeters for the tested model and the calculated camera positions are accurate. The more images and correspondences we get the better becomes the quality of the 3D points and the pose estimations of the cameras. It is also possible to use the system to generate 3D interior views.

One of the next steps will be the integration of existing detection approaches, like window detection, so that the effort to generate a model can be reduced even further. Furthermore, we will improve the texture selection by checking the images of occlusions. If no image exists where the whole surface is visible, we will merge image parts to a texture.

Acknowledgments

We thank the Deutsche Forschungsgemeinschaft(DFG) who partially funded this work under grant PA 599/10-1.

REFERENCES

[1] Falkowski, K. and Ebert, J., "A reference schema for interoperability between geo data and 3d models," in [*Geoinformatik 2011 - Geochange*], (2011). Accepted.

[2] Lee, S., Huertas, A., and Nevatia, R., "Modeling 3-d complex buildings with user assistance," in [*Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*], 170–177, IEEE (2000).

[3] Lee, S., Jung, S., and Nevatia, R., "Automatic integration of facade textures into 3d building models with a projective geometry based line clustering," in [*Computer Graphics Forum*], **21**(3), 511–519, Wiley Online Library (2002).

[4] Hartley, R. I., "In defense of the eight-point algorithm," *Pattern Analysis and Machine Intelligence* **19**(6), 580–593 (1997).

[5] Falkowski, K., Ebert, J., Decker, P., Wirtz, S., and Paulus, D., "Semi-automatic generation of full citygml models from images," in [*Geoinformatik 2009*], **35**, 101–110, Institut für Geoinformatik Westfälische Wilhelms-Universität Münster (2009).

[6] Lee, S. C. and Nevatia, R., "Extraction and integration of window in a 3d building model from ground view images," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* **2**, 113–120 (2004).

[7] Mayer, H. and Reznik, S., "Building façade interpretation from image sequences," in [*of ISPRS Workshop CMRT*], **XXXVI**, 55–60 (2005).

[8] Fiore, P. D., "Efficient linear solution of exterior orientation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **23**(2), 140–148 (2001).

[9] Zhang, Z., "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11), 1330–1334 (2000).

[10] Dick, A. R., Torr, P. H. S., and Cipolla, R., "Modelling and interpretation of architecture from several images," *Int. J. Comput. Vision* **60**(2), 111–134 (2004).

[11] Hartley, R. I. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Press, Cambridge, 2 ed. (2003).

[12] Cox, S., Daisey, P., Lake, R., Portele, C., and Whiteside, A., "Opengis geography markup language (gml) implementation specification," Tech. Rep. 3.1.1, Open Geospatial Consortium, Inc. (2001).

[13] Debevec, P. E., Taylor, and Malik, J., "Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach," in [*SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*], 11–20, ACM, New York, NY, USA (1996).

[14] Lourakis, M. and Argyros, A., "Sba: A software package for generic sparse bundle adjustment," *ACM Trans. Math. Software* **36**(1), 1–30 (2009).

[15] Rusu, R. B., Sundaresan, A., Morisset, B., Agrawal, M., and Beetz, M., "Leaving flatland: Realtime 3d stereo semantic reconstruction," in [*ICIRA '08: Proceedings of the First International Conference on Intelligent Robotics and Applications*], 921–932, Springer-Verlag, Berlin, Heidelberg (2008).