Automatic Registration of Oblique Aerial Images with Cadastral Maps

Martin Habbecke and Leif Kobbelt

Computer Graphics Group, RWTH Aachen University, Germany
http://www.graphics.rwth-aachen.de

Abstract. In recent years, oblique aerial images of urban regions have become increasingly popular for 3D city modeling, texturing, and various cadastral applications. In contrast to images taken vertically to the ground, they provide information on building heights, appearance of facades, and terrain elevation. Despite their widespread availability for many cities, the processing pipeline for oblique images is not fully automatic yet. Especially the process of precisely registering oblique images with map vector data can be a tedious manual process. We address this problem with a registration approach for oblique aerial images that is fully automatic and robust against discrepancies between map and image data. As input, it merely requires a cadastral map and an arbitrary number of oblique images. Besides rough initial registrations usually available from GPS/INS measurements, no further information is required, in particular no information about the terrain elevation.

1 Introduction

Aerial images of urban regions have been in wide-spread use for various applications for more than a century, with a strong focus on images taken vertically to the ground (i.e. nadir images). In contrast to vertical images, aerial images taken at an oblique angle with respect to the ground have the important advantage of providing information on building heights, appearance of facades, and terrain elevation. Thus, they are not only more intuitive for untrained viewers [1] but enable new kinds of applications like 3D city modeling [2–4], texturing [5–7], dense stereo matching [8], or photo augmentation [9], which are not possible in this form with vertical images. In recent years oblique aerial images have been created in large-scale projects even for medium-sized cities [1] and have become widely available e.g. as "bird's-eye view" in Microsoft's internet map service [10]. The combination of oblique images with cadastral maps is of special interest since it not only simplifies standard cadastral applications [1] but has the potential of strongly improving 3D city reconstruction techniques [2–4] in terms of automation and speed. However, the established standard tools for vertical aerial images cannot easily be applied to oblique imagery due to the varying scale of pixels across an image caused by perspective foreshortening, the strongly changing appearance between different views, and the inevitable (self-)occlusion of buildings. While the registration of oblique aerial images with

Fig. 1. Problem statement: Given a set of oblique aerial images (a) and a cadastral map (b), we compute the registration of the images with the map as shown in (c). Besides rough initial registrations, no further information is required. In particular, the cadastral map does not contain terrain elevation or building height information.

vertical images [11] and with LiDAR data [6, 7] has been studied before, the precise registration with cadastral maps and the process of *conflation* [12] (i.e., the removal of misalignment between images and map vector data) is still a challenging problem for oblique aerial images that has not been automated yet [13]. This problem is amplified by the fact that, instead of a single vertical image, at least four oblique views from different directions are required to fully cover individual objects. Thus, there is a strong need for a fully automated processing pipeline that includes a robust and precise geo-registration.

In this paper, we address the problem of registering oblique aerial images (cf. Fig. 1a) with digital cadastral maps containing the footprints of buildings (cf. Fig. 1b). The set of images is assumed to be sparse with the viewing directions being just the four cardinal directions since images of this kind are widely available. To allow for a robust registration, neighboring images are required to overlap by about 30-40%. While the resulting registrations (cf. Fig. 1c) can be used for various purposes, our main target application is the reconstruction and texturing of 3D city models.

We assume that rough initial estimates of the per-image registrations are known, as they can usually be acquired using in-flight GPS and orientation measurements. No further information is required, in particular no information about the terrain elevation. In contrast to previous approaches, our system is fully automatic without the need for user interaction. For each input image, the registration is recovered as parameters of a perspective projection that aligns the map with the image. If the intrinsic calibration of the input images is not known, it is recovered during the registration process in addition to the extrinsic calibration. While the recovery of radial distortion parameters could seamlessly be integrated as well, this has not been necessary for the images used in our experiments. Due to different creation times and measurement errors during map generation, a certain level of discrepancy between the digital map and the input images is inevitable. We employ robust sampling techniques to cope with such cases.

1.1 Method Overview

The registration process performs the following steps. Similarly to [6], for each individual image our algorithm first detects the vanishing point that corresponds to the vertical scene direction (cf. Section 2.1). This vanishing point reduces the degrees of freedom of the extrinsic calibration from 6 to 4, thereby effectively simplifying the later search for camera parameters. For each image, the algorithm then detects line segments that correspond to vertical scene edges, i.e., line segments that pass through the respective vanishing point.

In the second step, our method estimates the extrinsic and, if not provided, intrinsic calibration of each image (cf. Section 2.2). This process is based on corresponding pairs of map corner vertices and image line segments detected in the previous step. Since these correspondences are unknown, we generate a large set of candidates and employ the RANSAC [14] approach to find a valid subset. Distance measurements using the Mahalanobis distance and an integrated approximation of the per-image terrain elevation yield a robust procedure. This step already results in very good alignments of the oblique images with the map.

Due to the usage of vertex-to-line constraints, however, there is still an unknown height offset between pairs of images left. Furthermore, due to slight inaccuracies in the detected vanishing points, the offset usually is not constant for an image but varies according to an unknown linear height function. To compensate for both effects, in a final step, we detect horizontal (in scene space) edges on building facades, robustly match them across pairs of images, and solve a bundle-adjustment-like global optimization problem over all camera parameters (cf. Section 2.3). This results in precise and compatible registrations of all oblique images with the cadastral map.

The paper continues with a discussion of related work. The steps of our processing pipeline are presented in detail in Section 2. Results are presented in Section 3 and we conclude with a discussion of our method in Section 4. Please see the accompanying video for an extended overview of our approach.

1.2 Related Work

Geo-registration, the alignment of overlapping images, and conflation are well-understood problems for vertical aerial images and a variety of established techniques exists [15, 16]. While these processes can often be automated for vertical images, the same approaches cannot easily be transferred to oblique images due to perspective foreshortening, occlusion of ground points and buildings, and the strongly varying appearance of e.g. facades for different vantage points. Gerke and Nyaruhuma [17] explicitly address the calibration of the extrinsic and intrinsic parameters of oblique aerial images. They present a method based on manually specified points, horizontal or vertical lines, and right angles, and compare their approach to several commercial products. It was shown that for the case of oblique images, commercially available solutions are still inferior compared to an approach tailored to the specific properties of these images. Frueh et al. [5] present a system that automatically registers oblique aerial images with a 3D

city model with the goal of texture generation. With the same goal, Ding et al. [6] and Wang and Neumann [7] register 3D LiDAR models with oblique aerial images. All three approaches are based on matching line segments between the 3D model and the images. [5] matches lines directly, [6] and [7] combine individual line segments to more complex descriptors for improved matching robustness. While these methods yield very good registration results, they cannot easily be transferred to our setting since cadastral maps do not provide a sufficient number of edge candidates for matching. Furthermore, cadastral maps do not provide information about building heights, roof shapes, and terrain elevation, all of which is contained in LiDAR / 3D model data and which is crucial for the above methods to work. The lack of this information makes the problem of registration with cadastral maps more challenging.

Läbe and Förstner [18] have demonstrated the feasibility of a general structure-from-motion approach for the recovery of camera parameters of oblique images. However, since structure from motion requires a sufficiently large set of features matched across the images, this approach only works for densely sampled image sequences. Due to the strong appearance changes in sparse sets of oblique images as we use them, automatic feature matching is not feasible. Sheikh et al. [11] present a technique to register perspective oblique images to a geo-referenced orthographic vertical image mapped onto a digital elevation model (DEM). While this works well for images taken at high altitudes such that the DEM can be considered to be a smooth surface, it cannot be applied to images taken at lower altitudes where buildings result in considerable relative height differences. Mishra et al. [13] detect inconsistencies in vector data, especially street data, by projection into oblique images. Their approach is able to detect errors in the vector data as well as in the calibration. It is, however, not able to correct the calibration.

An alternative to the traditional approach of geo-registration in a post-process (i.e., off-line) is the *direct geo-registration*. Here the position and orientation of the camera is measured during flight. To achieve a sufficient level of registration precision, this approach requires specialized, expensive GPS/INS equipment and a large manual calibration effort to compensate for the different poses of the measurement devices and the camera. Such systems have been shown to achieve registration precisions of below 1m for vertical [19] and for oblique aerial images [20]. However, in the same work Grenzdörfer et al. [20] also report that the fully automatic texturing of an existing 3D model has not been possible due to too large registration errors of about 1-3 meters. Similarly, the texturing efforts by Stilla et al. [21], the evaluation of oblique aerial images for cadastral applications by Lemmens et al. [1], and the texturing approaches [6, 7] have shown that the precision of direct geo-registration solutions is often not sufficient without further processing. Furthermore, as discussed by Gerke and Nyaruhuma in [17], the traditional approach of off-line determination of camera poses cannot be replaced by direct geo-registration for several reasons: this technology is not applicable to unmanned airborn vehicles (UAVs) with limited loading weight, it has a high burden of precise calibration that has to be redone

every time the system is modified, and the registration information might not be available at all depending on the source of the images. We hence believe that a combination of direct and automated off-line geo-referencing is the simplest, most robust, and most effective approach.

2 Image Registration Pipeline

As outlined in the introduction, our registration approach consists of three main steps. These steps will now be discussed in detail.

2.1 Vanishing Point and Vertical Edge Detection

Vanishing points corresponding to the scene's vertical direction are among the few entities that can easily be computed in oblique aerial images without further scene knowledge. Even for images with strong occlusion caused by tall buildings, usually a large number of vertical building edges is visible. Furthermore, although oblique images are most often captured with long focal distances, there is still enough variation in the orientation of projected vertical edges to allow for a stable detection of this particular vanishing point. Following [6], we exploit these points to fix two degrees of freedom of the extrinsic camera orientation, thereby stabilizing the estimation of initial registrations in the next step.

The detection of vanishing points is accomplished by a very simple yet effective procedure. We compute edge-pixels using the Canny-operator [22] and then extract straight line segments by least-squares line fitting. We then employ a simple RANSAC-based procedure that randomly picks two line segments, computes their intersection as hypothesis of the vanishing point, and evaluates its support using the remaining segments. By exploiting a-priori knowledge about the position of the vanishing point, this approach has proven to be extremely robust in our experiments: Since we can safely assume that the vertical vanishing point lies way below the image, only hypotheses with a y-coordinate of at least two times the image height are considered for further evaluation. The winning hypothesis is refined by an MLE procedure [23] with all inlying line segments.

The camera parameter optimizations in the second and third step are based on correspondences between map corner vertices and image line segments that agree with the vanishing points. While the inlying line segments of the previous step could well be used for this purpose, we found that additional segments can be detected by a slightly modified second detection pass. For each pixel, we compute the derivative along the direction perpendicular to the line connecting the vanishing point and the pixel's position. Applying the Canny-operator (non-maximum suppression and thresholding) to the directional derivatives effectively suppresses pixels with strong but wrongly oriented gradients. A low threshold then yields many small connected components that can easily be discarded, but also preserves line segments distorted by noise or with smaller gradient magnitude. The final line segments are again obtained as ML estimates constrained to pass through the vanishing point.

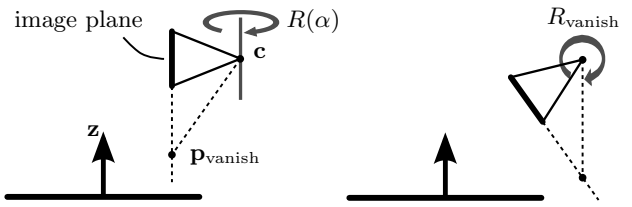

Fig. 2. Parameterization of the extrinsic camera calibration. $\mathbf{z}$ denotes the scene's vertical direction and $\mathbf{p}_{\text{vanish}}$ denotes the vanishing point in image space. $R(\alpha)$ rotates around $\mathbf{z}$, R_{vanish} aligns the vanishing direction induced by $\mathbf{p}_{\text{vanish}}$ with $\mathbf{z}$.

2.2 Estimation of Initial Registrations

The central goal of this step is the recovery of good estimates of the registration parameters for each individual image in the form of perspective pin-hole projections [24] with 6 extrinsic (rotation and camera center) and 5 intrinsic parameters, respectively. Due to the known vanishing points, we need to recover 4 extrinsic parameters only: the vertical vanishing point of an image determines the orientation of the camera relative to the scene's vertical direction. We therefore only need to recover a single orientation parameter α, yielding an extrinsic orientation parameterized as

$$T(\alpha, \mathbf{c}) := R_{\text{vanish}} R(\alpha)(I| - \mathbf{c}) \ \in \mathbb{R}^{3\times 4} \tag{1}$$

where $\mathbf{c}$ is the camera center, $R(\alpha) \in \mathbb{R}^{3\times 3}$ is a rotation around the scene's vertical axis, and $R_{\text{vanish}} \in \mathbb{R}^{3\times 3}$ aligns this axis with the vanishing direction induced by the vanishing point (cf. Fig. 2). In contrast to [6] and [7], we do not assume a fixed camera center $\mathbf{c}$ in this step to be able to handle cases where the initial registrations are not provided by GPS measurements and are hence less precise. We assume that a rough estimate of the focal distance is known at this point and set the remaining intrinsic parameters to their canonical values (aspect ratio 1, zero skew, principal point in the image center). A full optimization of all intrinsic parameters is done in the last step (cf. Section 2.3).

The parameter computation is based on correspondences between line segments $\mathbf{l}$ in image space as detected in the previous step and corner vertices $\mathbf{v}$ of the given map. For a set of corresponding lines and map vertices $M := \{(\mathbf{l}_i, \mathbf{v}_i)\}$, we find the optimal projection parameters by minimizing

$$E(\alpha, \mathbf{c}) := \sum_i \text{dist}_2(\mathbf{l}_i, KT(\alpha, \mathbf{c})\mathbf{v}_i)^2 \tag{2}$$

with respect to α, $\mathbf{c}$. Here $K \in \mathbb{R}^{3\times 3}$ is the intrinsic calibration matrix, $KT\mathbf{v}$ denotes the perspective projection of a map corner vertex $\mathbf{v}$ into image space and $\text{dist}_2(\cdot, \cdot)$ denotes the Euclidean distance between a 2D point and the supporting line of an image space line segment. The varying parameters are optimized using the Levenberg-Marquardt method. Notice that, if only lines $\mathbf{l}$ passing through

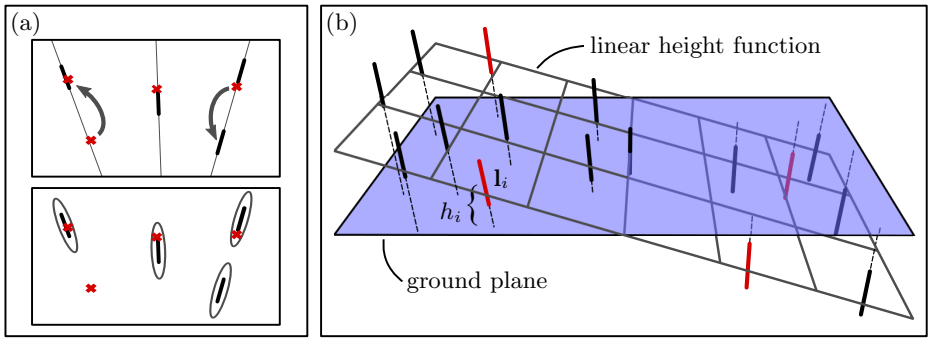

Fig. 3. (a) Inlier determination with Euclidean distance to the supporting line (top) and with Mahalanobis distance (bottom). The latter case effectively prevents false positive inliers denoted by arrows in the top figure. (b) Illustration of a linear height function computed for a random set of vertical lines $\mathbf{l}_i$ (shown in red) in the RANSAC procedure that finds initial per-image registration parameters. This approach relaxes the assumption of a horizontally flat terrain to a planar but arbitrarily oriented terrain.

the vanishing point are used in (2) as assumed so far, the solution would degenerate to a state where the projections of all map vertices collapse into the vanishing point. In other words, the recovered camera would be moved up extremely high above the map. To prevent this, we construct an additional line constraint perpendicular to the first line. More precisely, for the first constraint $(\mathbf{l}_0, \mathbf{v}_0)$ we add a constraint $(\tilde{\mathbf{l}}_0, \mathbf{v}_0)$ with $\tilde{\mathbf{l}}_0$ being perpendicular to $\mathbf{l}_0$ and passing through $\mathbf{l}_0$'s center.

Since it is not known which are the valid correspondences, we employ RANSAC to find them. If a rough estimate of the focal distance is known, the size of each sampling set is 3 to determine the 4 unknown extrinsic parameters, due to the additional constraint for the first correspondence. Candidate correspondences are constructed by first determining a set of visible (from the initially provided rough camera perspective) map vertices $\mathbf{v}$, projecting them into image space, and finding all nearby line segments $\mathbf{l}$. The search radius in image space has to be chosen according to the discrepancy between the initially provided registration and the correct solution. That is, the search space has to be large enough such that the correct matches are contained in the set of candidate correspondences, and as small as possible to speed up the RANSAC process. In our experiments, we have found that usually a search radius of 80 to 130 pixels (i.e., about 12 to 20 meters in world space) is sufficient even for only rough initial registrations. The RANSAC procedure then works in the usual way by picking random correspondences, solving for optimal parameters by minimizing (2), and counting all inlying correspondences.

Depending on the radius of the candidate search space, the number of false positive inliers can become very large. Here false positives are map vertices $\mathbf{v}$ that project close to the supporting line of a segment $\mathbf{l}$, but do not actually belong to the respective segment (cf. Fig. 3a). To counter this problem, the Euclidean

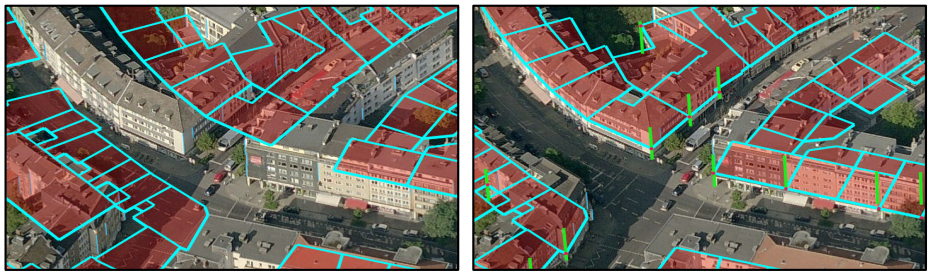

Fig. 4. Result of the initial registration process. Starting from a rough estimate of the registration parameters (left), our system automatically recovers good initial registrations for each individual image (right). Vertical line constraints are shown in green.

distance to a segment's supporting line is replaced by an elliptical Mahalanobis distance during inlier determination. As a consequence, by keeping the stretch of the ellipses along the line segment directions small, it is implicitly assumed that the underlying terrain is horizontally flat, since only line segments slightly above or below the projection of the map yield a sufficiently small Mahalanobis distance. We relax this assumption by approximating the fraction of the terrain visible in a single image by a plane with arbitrary slope. This is implemented by computing a linear height field for each random set of matching candidates. More precisely, after the optimization of (2), a height value h_i is computed for each random match $(\mathbf{l}_i, \mathbf{v}_i)$. The least-squares plane of all height values then yields the linear height function (cf. Fig. 3b). During the determination of inlying correspondences, all map vertices $\mathbf{v}$ are shifted up or down according to the height function before projection into the image. In our experiments we have found that both the Mahalanobis distance and the linear height functions introduce little extra computational effort, but effectively reduce the number of false positive inliers. Fig. 4 shows an example of the alignment before and after the initial registration process.

2.3 Global Optimization

Up to now, we have considered the separate registration of individual images only. Due to the additional, arbitrarily chosen height constraints $(\tilde{\mathbf{l}}_0, \mathbf{v}_0)$ introduced in the previous step, the registration is not yet globally consistent across all images. In an ideal setting, the only step missing for a consistent registration of all images would be a height adjustment of each image with respect to a common reference, i.e., a translation of all but one cameras along the scene's vertical direction. Unfortunately, as shown in Fig. 5, this is not sufficient most of the time, since the necessary height offset to align pairs of images is not constant but rather varies over the images.

An analysis of this problem shows that the offset variations are caused by slight inaccuracies in the detected vanishing points: For a fixed focal distance, the orientation of the ground plane with respect to the camera is determined

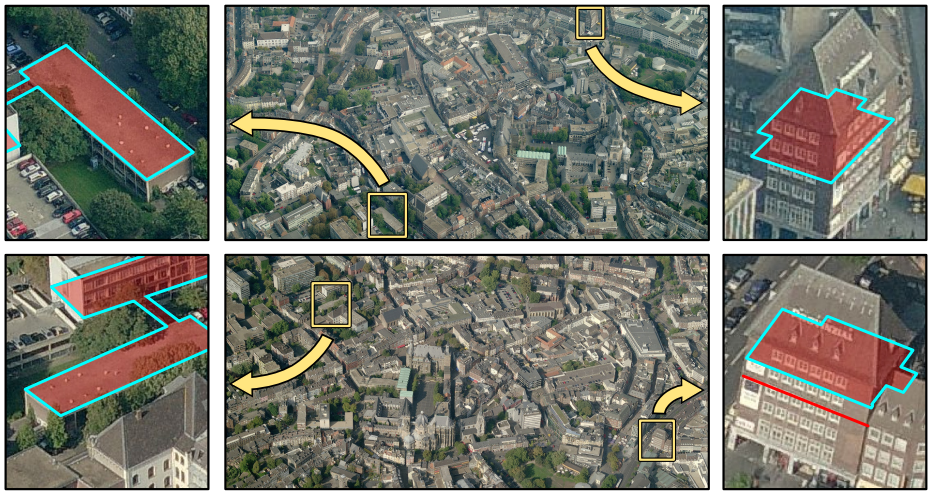

Fig. 5. Visualization of height differences between pairs of images. The map is projected to compatible positions for a certain region of the map (left). Due to slightly inaccurate vanishing points, the orientations of the cameras are slightly tilted. This yields incompatible map projections in other map regions. The expected map position is marked with a red line on the facade (right). We solve this problem by optimizing the parameters of all cameras including the vanishing points in the final step of the registration pipeline.

by the vanishing point only. While the vanishing points detected in Section 2.1 yield plausible alignments for each individual image, comparing the ground plane orientations for overlapping pairs of images as done in Fig. 5 reveals slightly incompatible orientations. Due to limited image quality and resolution, we cannot expect to improve the precision of the vanishing point detection to a sufficient level. We therefore decided to integrate the vanishing points as varying parameters into the final global optimization and thereby recover compatible orientations of all images with respect to the ground plane.

To be able to do so, we need to define constraints that act as coupling forces between different images and that are able to capture the orientation differences we want to remove. A viable approach is to detect horizontal (in scene space) edges on building facades and match them across two or more images. While the systematic detection of horizontal facade edges is difficult without scene knowledge, it becomes feasible due to the individual registrations of each image with the map: For each image, we can now determine visible map edges, restrict the search for facade line segments to narrow vertical bands (cf. Fig. 6a), and discard facade lines with false orientations. To match facade line segments between images, we need to take the unknown ground plane orientation differences into account. From the above analysis follows that the orientation difference between two images can be compensated for by a bivariate linear height function, i.e., by a planar offset. We thus determine an appropriate height function for each

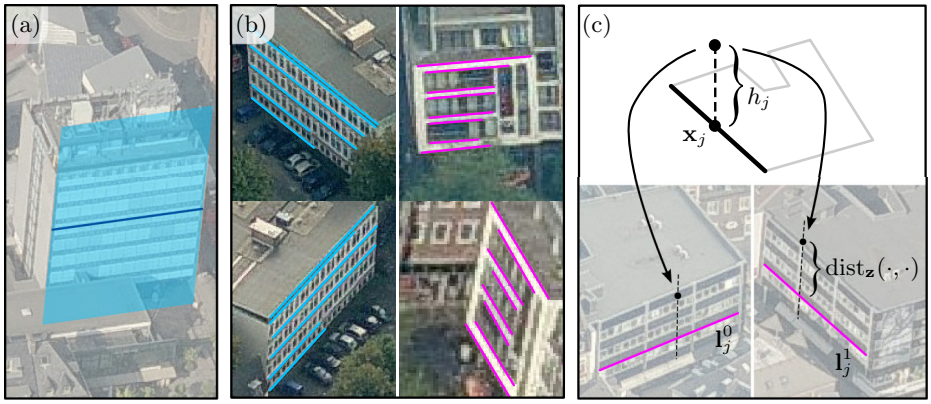

Fig. 6. (a) Search area for horizontal facade edges defined by the projection of a map edge. The height of the search area is defined by the expected height of buildings. We use 20m above and below each edge in all our experiments. (b) Examples of matching facade edges in two different views. (c) Construction of facade edge constraints. The unknown height values h_j are part of the optimization as varying parameters.

(but one) aerial image using a RANSAC procedure. The size of the sampling set is 3, the set of candidates consists of all possible pairs of line segments on the same facade in both images which additionally have the same gradient orientation. All pairs of facade edges that agree with the winning hypothesis are used as constraints in the subsequent global optimization. Notice that for a single facade several pairs of edges can agree with the winning hypothesis as depicted in Fig. 6b.

The global optimization is solely based on constraints measuring the distance between projections of 3D vertices to 2D lines. We reuse the correspondences between map corner vertices and vertical image lines and add horizontal line constraints for facade edges visible in two or more images. Hence, in addition to the correspondences $(\mathbf{l}_i^k, \mathbf{v}_i)$ from Section 2.2 (with an additional index k counting images), we construct correspondences of the form $(\mathbf{L}_j, \mathbf{x}_j)$ with $\mathbf{L}_j$ being a set of horizontal lines in two or more images corresponding to the same map edge, and $\mathbf{x}_j$ being the 3D center point of this edge. See Fig. 6c for an illustration for the case of two images. The objective function of the global optimization over all cameras is

$$E(\{P_k\},\{h_j\}) := \sum_{(\mathbf{l}_i^k,\mathbf{v}_i)} \mathrm{dist}_2\big(\mathbf{l}_i^k, P_k\mathbf{v}_i\big)^2 + \sum_{(\mathbf{L}_j,\mathbf{x}_j)} \sum_{\mathbf{l}_j^k\in\mathbf{L}_j} \mathrm{dist}_{\mathbf{z}}\big(\mathbf{l}_j^k, P_k(\mathbf{x}_j + h_j\mathbf{z})\big)^2. \tag{3}$$

Since the per-constraint height values h_j above the map's supporting plane are unknown, they are part of the optimization as varying parameters. $\mathbf{z}$ denotes the scene's vertical direction. Notice that for facade edge terms we do not compute the minimal Euclidean distance but rather the correct distance along the projection of $\mathbf{z}$, denoted by $\mathrm{dist}_{\mathbf{z}}$ (cf. Fig. 6c). In this procedure there is no need for

Fig. 7. Left: Registration result for one out of 36 images (3×3 for each cardinal direction) of an urban area. Right: Projection of a 3D building model into 4 images (out of 11 in which it is visible) to verify the precision of the automatically obtained registrations. The projections of the model are aligned with the images with only minor deviations of at most 1-2 pixels, which translates into a maximal positional imprecision of 15-30cm in scene space.

artificial height constraints anymore. To prevent the solution from collapsing, we simply fix the first height value to $h_0 := 0$. The parameters are again optimized using the Levenberg-Marquardt algorithm. We now perform a full optimization of all 6 extrinsic and, if required, also of the intrinsic parameters of all cameras simultaneously. Please notice that the employed optimization strategy is prone to converge to a local minimum if not initialized properly. Due to the good initial per-image registrations obtained in Section 2.2, we have, however, never encountered a case where the optimization converged to a local minimum.

3 Results

In the first experiment, we have applied our algorithm to a set of 36 oblique images (i.e., 3×3 for each of the four cardinal directions) of an urban region. The images, which have been downloaded from [10], have a resolution of 4008×2672. Neighboring images of the same cardinal direction have an overlap of about 30-40%. The per-image processing steps (detection of vanishing point and vertical lines, computation of initial registration, detection of horizontal facade lines and height offset estimation) take about 20 seconds for each image on an Intel Core i7 920 CPU. The subsequent full Levenberg-Marquardt optimization of all parameters for 36 images took 80 seconds with $7 \times 36 = 252$ varying camera parameters and 16,340 varying height values, as well as 9,617 vertical and 46,915 horizontal line constraints. The resulting RMSE of (3) is 0.863 pixels per 3D vertex to 2D image line projection. Vertical vanishing points move by 150 pixels on average during the optimization. This translates into an orientation change of the ground plane by 0.8 degrees.

To validate the accuracy of the recovered registration, we have constructed several 3D building models and projected them into various different views. The

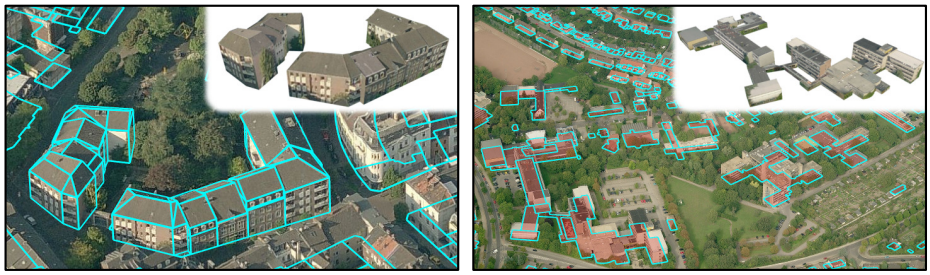

Fig. 8. Left: Result of 5 minutes of modeling with a prototype system which is based on the automatically computed registrations and the cadastral map. Right: Application of our approach to a sub-urban region. Even though less vertical and horizontal edges are available in such images, our system is able to recover precise registrations.

footprint of the highest building in Fig. 7 has dimensions 30m×12m. Visual inspection (due to the lack of ground truth registrations) shows a precise alignment of the 3D scene with the images within 1-2 pixels. This translates into an accuracy in scene space of below 15-30cm.

With the registration in place, the generation of a correct terrain height map and the adjustment of building heights both become simple one-dimensional problems. In particular, a valid height map can be generated by means of linearly interpolating very few constraints. To further validate the quality of our registrations, we have implemented a simple interactive modeling system similar to those of [3, 4] to rapidly create 3D buildings. The precise registration enables a modeling approach that overlays the current state of the model on top of the aerial images, thereby allowing for the easy reconstruction of correct building shapes and dimensions. Fig. 8(left) shows the result of just about 5 minutes of manual modeling using the automatically generated registration and the cadastral map as a basis.

In a second experiment we have applied the automatic registration approach to a sub-urban region, cf. Fig. 8(right). Even though much less vertical and horizontal lines have been detected, our system still works as expected and generates a precise registration. For more result please see the supplemental video.

4 Discussion

The main sources of information exploited in our work are horizontal and vertical lines in the input images. Thus, our method only works correctly if a sufficient number of lines is available. During this project we have found, however, that a large number of both kinds of lines can safely be assumed to be present in images of urban regions: Vertical edges frequently appear at the corners of buildings or due to the different appearances of neighboring facades; horizontal edges are induced by the rims of roofs, by balconies, or by windows. We have never encountered a case where the system failed due to too few available lines. For

the detection of vertical vanishing points (cf. Section 2.1), more sophisticated methods like, e.g., [25] are available. However, we use a simpler approach that exploits a-priori knowledge about the position of the vanishing points since it has turned out to be extremely robust, and since perfect precision that renders the adjustment of the vanishing points unnecessary during the global optimization (cf. Section 2.3) cannot be expected for any alternative method.

Our system has a few intuitive parameters that need to be specified by the user. Foremost, a threshold is required to distinguish inliers from outliers during the search for 3D vertex to 2D line correspondences (cf. Section 2.2) and for matching horizontal facade lines (cf. Section 2.3). For both cases a distance threshold of 2.0 pixels has worked well in all our experiments. In the search for vertex-to-line correspondences to determine per-image registrations, we have found that we usually have to deal with an inlier ratio of only 6-7%. For a sampling set size of 3 correspondence we therefore require about 20k RANSAC iterations for a confidence of 99% to find an inlier-only subset at least once. The RANSAC process in Section 2.3 is less problematic since the inlier ratio usually is larger than 13%. Thus, for 3 random correspondences in each iteration, 2.1k iterations are sufficient.

If no information about the position and orientation of the input images is known (as it may be the case for images from the internet), our approach enables a simple interface to specify rough initial registrations: Due to the recovered vanishing points, the user needs to only specify a one-dimensional orientation α (cf. Fig. 2) and the rough translation $\mathbf{c}$ of the camera. Both operations can be mapped to simple interactions in an interface that overlays the input images with the cadastral map. After a precise estimate of the first image's registration parameters has been computed (cf. Section 2.2), these parameters are used as starting values for neighboring views, thereby turning the process of providing rough initial registrations into a matter of seconds per image.

From the constraints used in the global optimization, a rough estimate of the terrain's height map can be derived. Vertical line segments provide height information by their lower endpoint, for horizontal line segments height values h_j have been explicitly computed (cf. Section 2.3). Thus, a height map can be constructed by collecting the minimal height value for each building footprint and by propagating height information to buildings without constraints by linear interpolation. While this construction yields only a very rough approximation, it is able to compensate for large-scale variations of the terrain elevation.

Acknowledgment: This project was funded by the DFG Cluster of Excellence *UMIC* (DFG EXC 89), and the Aachen Institute for Advanced Study in Computational Engineering Science (AICES).

References

1. Lemmens, M., Lemmen, C., Wubbe, M.: Pictometry: Potentials for land administration. In: Proc. of the 6th FIG reg. conf., Int'l Fed. of Surveyers (2007)
2. Vanegas, C.A., Aliaga, D.G., Benes, B.: Building reconstruction using manhattan-world grammars. In: Proc. of CVPR. (2010)

3. Google Building Maker: A 3d city modeling approach based on oblique aerial images. http://sketchup.google.com/3dwh/buildingmaker.html (2010)
4. Gülch, E.: Extraction of 3d objects from aerial photographs. Proc. COST UCE ACTION C4 Workshop (1996)
5. Frueh, C., Sammon, R., Zakhor, A.: Automated texture mapping of 3d city models with oblique aerial imagery. In: Proc. of 3DPVT. (2004) 396–403
6. Ding, M., Lyngbaek, K., Zakhor, A.: Automatic registration of aerial imagery with untextured 3d lidar models. In: Proc. of CVPR. (2008)
7. Wang, L., Neumann, U.: A robust approach for automatic registration of aerial images with untextured aerial lidar data. In: Proc. of CVPR. (2009)
8. Gerke, M.: Dense matching in high resolution oblique airborne images. CMRT09 (2009) 77–82
9. Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D.: Deep photo: Model-based photograph enhancement and viewing. In: Proc. of SIGGRAPH Asia. (2008)
10. Microsoft Corp.: Bing maps. http://www.bing.com/maps (2010)
11. Sheikh, Y., Khan, S., Shah, M., Cannata, R.: Geodetic alignment of aerial video frames. Video Registration, Video Computing Series (2003)
12. Wu, X., Carceroni, R., Fang, H., Zelinka, S., Kirmse, A.: Automatic alignment of large-scale aerial rasters to road-maps. In: Proc. of ACM GIS. (2007)
13. Mishra, P., Ofek, E., Kimchi, G.: Validation of vector data using oblique images. In: Proc. of ACM GIS. (2008)
14. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
15. Fogel, D.N., Tinney, L.R.: Image registration using multiquadric functions, the finite element method, bivariate mapping polynomials and thin plate spline. Technical Report 96-1, National Center for Geographic Information and Analysis (1996)
16. Mena, J.B.: State of the art on automatic road extraction for gis update: a novel classification. Pattern Recogn. Lett. **24** (2003) 3037–3058
17. Gerke, M., Nyaruhuma, A.: Incorporating scene constraints into the triangulation of airborne oblique images. In: ISPRS XXXVIII 1-4-7/WS. (2009)
18. Läbe, T., Förstner, W.: Automatic relative orientation of images. In: Proc. of the 5th Turkish-German Joint Geodetic Days. (2006)
19. Cramer, M., Stallmann, D.: System calibration for direct georeferencing. In: IAPRS, Volume XXXIV, Com. III, Part A. (2002) 79–84
20. Grenzdörffer, G.J., Guretzki, M., Friedlander, I.: Photogrammetric image acquisition and image analysis of oblique imagery. The Photogrammetric Record **23** (2008) 372–386
21. Stilla, U., Kolecki, J., Hoegner, L.: Texture mapping of 3d building models with oblique direct geo-referenced airborne IR image sequences. In: ISPRS Workshop: High-resolution earth Imaging for geospatial information. (2009)
22. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence **8** (1986) 679–714
23. Liebowitz, D., Zisserman, A.: Metric rectification for perspective images of planes. In: Proc. of CVPR. (1998) 482–488
24. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press (2003)
25. Almansa, A., Desolneux, A., Vamech, S.: Vanishing point detection without any a priori information. IEEE PAMI **25** (2003) 502–507