Registration of Spherical Panoramic Images with Cadastral 3D Models

Aparna Taneja, Luca Ballan, Marc Pollefeys
Department of Computer Science, ETH Zurich, Switzerland
{*aparna.taneja,luca.ballan,marc.pollefeys*}*@inf.ethz.ch*

***Abstract*—The availability of geolocated panoramic images of urban environments has been increasing in the recent past thanks to services like Google StreetView, Microsoft StreetSide, and Navteq. Despite the fact that their primary application is in street navigation, these images can be used, along with cadastral information, for city planning, real-estate evaluation and tracking of changes in an urban environment. The geolocation information, provided with these images, is however not accurate enough for such applications: this inaccuracy can be observed in both the position and orientation of the camera, due to noise introduced during the acquisition.**
We propose a method to refine the calibration of these images leveraging cadastral 3D information, typically available in urban scenarios. We evaluated the algorithm on a city scale dataset, spanning commercial and residential areas, as well as the countryside.

I. Introduction

The availability of panoramic images depicting urban scenarios, has been increasing in the recent past, thanks to services like Google StreetView, Microsoft StreetSide, and Navteq. Even though, the primary intention behind capturing these images was street navigation, they certainly will become, in the near future, a huge source of information for other kinds of applications as well. It is sufficient in fact to consider that the panoramic images offered in StreetView, represent most of the streets of our globe with a spatial sampling which becomes very dense in urban environments. Applications such as real estate evaluation, tracking of changes, and city planning would certainly benefit from such a huge amount of data, particularly when used together with 3D information.

While in the recent past, a lot of attention has gone to develop techniques aimed at inferring this 3D information from a scene [1], [2], city administrations already maintain such information for cadastral applications. This information is typically encoded into 3D mesh models representing the main constructions of a city.

While these 3D models are geolocated very accurately, the same cannot be said about the panoramic images whose geolocation information suffers from substantial noise due to the way these images were originally captured: typically from a car driving around, recording its location and orientation with a GPS, a compass and inertial sensors.

Despite the post processing procedures typically performed on this data, in order to reduce the noise, the quality of the registration is still not sufficient for geo-applications. As an example, Figure 1(left) shows the result obtained by superimposing a cadastral 3D model on top of a StreetView image. The visible misalignment clearly indicates that this geo-location data cannot be used as it is.

We present an algorithm to automatically refine the pose of spherical panoramic images, such as the ones in Google StreetView, leveraging the cadastral 3D information typically available in urban environments. Precisely, we propose to first register these images with respect to the cadastral 3D model by aligning the building outlines. Since this model is accurately geolocated, the registration process results in an accurate geolocation of the images as well.

To overcome the challenges involved in the extraction of the building outlines in natural images, and to deal with occluders frequently occurring in urban scenarios, we propose an iterative optimization technique aimed at estimating jointly the camera pose and the building outlines.

Unlike previous approaches, we focus on StreetView images, and propose a simple and easy to implement technique to encourage the usage of these images in a wider range of applications than is possible now.

II. Related Work

The literature regarding registration of color images with respect to point clouds, range scans or 3D models is vast, and can be subdivided into three main classes.

The first class incorporates all the methods which perform registration by extracting features on the images, and by matching them with the corresponding points on the point cloud. Typical features used in these cases are lines [3], building bounding boxes [4], skylines [5], and SIFT descriptors [6], [7]. Once these features are matched with the corresponding points on the 3D model, a 2D-to-3D registration of the image is performed.

The second class represents the 3D-to-3D registration approaches which instead make use of multiple images to perform a coarse 3D reconstruction of the scene. This reconstruction is then registered with the original 3D model with the consequent estimation of the pose of the original images. This last step is typically performed using rigid [8] or non-rigid ICP [9], [10].

The last class proposes more complex techniques aimed at finding the registration parameters by adopting generative

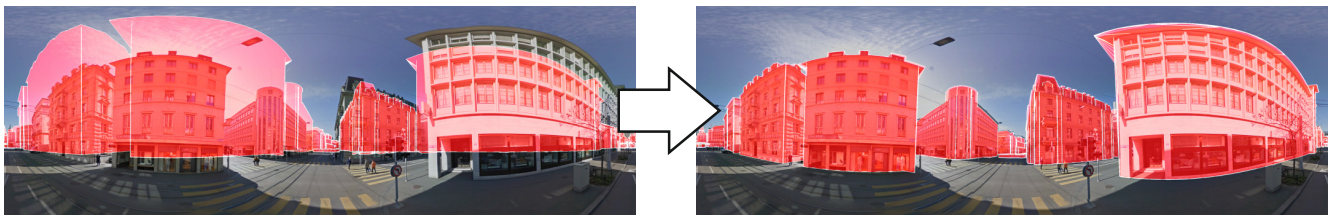


Figure 1. Overlay of a Google StreetView image with a cadastral 3D model before and after applying our algorithm.

approaches aiming at maximizing either the mutual information between the image and model, like in [11], or at maximizing the photo-consistency between images, when these are available, like in [12].

Our method falls in this last class. In particular, we propose a generative approach aiming at aligning an input spherical panoramic image with respect to a 3D model exploiting building outlines. Due to the difficulties in extracting these outlines from natural images, we propose to optimize for this segmentation jointly with the camera pose. Unlike 3D-to-2D registration approaches, our method does not rely on 3D-to-2D feature correspondences, but it uses a shape matching metric to quantify the accuracy of the alignment between the building outlines extracted from the images and from the model. Differently from [5], which assumes images captured from an upward facing camera in urban canyons, we chose to address the more challenging scenario of ground imagery, where occluders, such as trees, vehicles, or construction sites, can corrupt the visual information significantly.

III. The input

The administration of a city typically maintains cadastral information for the purpose of city planning and monitoring changes. This information is typically encoded into 3D mesh models representing the main constructions present in the city. These models are precisely geolocated, and their geometric accuracy is generally high. On the contrary, their level of detail is very basic, since, for most cadastral applications, small details are not necessary. In most of the cases, in fact, these models consists only of simple bounding boxes, approximating the buildings, augmented with features, like roofs and chimneys.

Google StreetView and Microsoft StreetSide offer a large and publically available dataset of panoramic images covering most of the streets of our planet. Each of these images consists of a full spherical panorama with resolution generally up to 3328×1664 pixels (even higher in some locations), covering a field of view of 360 degrees by 180 degrees. Google also provides different APIs to easily download these images from the web.

Each panoramic image comes with geolocation information encoding the latitude, the longitude, and the orientation at which the image was taken. Due to the way these images were acquired, this geolocation information is affected by a noise which can be as much as ± 5 meters in the location, and as much as ± 6 degrees in the orientation. Despite the fact that these values may seem low, superimposing an accurately geolocated 3D model on top of a StreetView image results in visible misalignments which might not be tolerable for some applications (see Figure 1(left)).

IV. Algorithm

In order to compute an accurate estimate for the position and the orientation at which a given panoramic image was captured, we exploit the information provided by the cadastral 3D model. In particular, we aim at registering this image with respect to the 3D model by matching building outlines. Since this model is accurately geolocated, the registration process would result in an accurate geolocation of the image as well.

Building outlines are very informative cues for registration because they represent multiple features in the scene such as the sky line, the road line, and the intra-building lines. While building outlines can be extracted easily from the cadastral 3D model, estimating the same from natural images, such as the ones downloaded from StreetView, is not as trivial.

In fact, the variety of elements typically present in an urban environment (e.g. traffic lights, shops, advertisements, construction sites, bus stops and rivers), as well as, different weather and lighting conditions which change the appearance of the scene, make the task of segmentation very challenging, and almost impossible to perform accurately without any good prior information. Moreover, occluders such as vegetation, vehicles, and pedestrians, often present in an urban environment, drastically contribute towards erroneous segmentations.

To cope for this, we propose an iterative pose estimation approach aiming at jointly optimizing for both the camera pose and the building outlines. Figure 2 shows a schematic overview of the proposed algorithm. An initial segmentation is first performed on the input images with the aim of labeling each pixel as belonging to sky, buildings, roads,

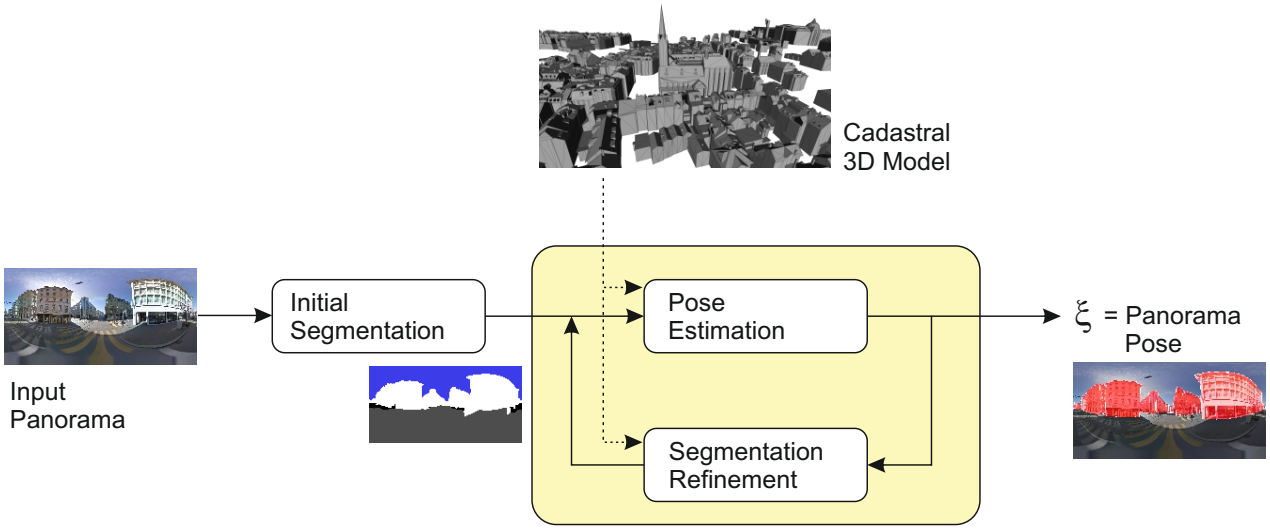


Figure 2. Overview of the proposed algorithm.

trees, vehicles or pedestrians. This information is then used to obtain a coarse pose estimate for the image. The initial segmentation is then refined using the computed pose. Pose estimation is performed again, and this process is repeated iteratively until convergence.

In the following sections, each step of the algorithm will be explained in detail.

A. Initial Segmentation

In the first step, an object class segmentation is performed on the input panoramas in order to label each of its pixels as belonging to sky, buildings, roads, trees, vehicles or pedestrians. At this point of the algorithm, we chose to ignore the pose provided by StreetView, since it is quite inaccurate to help in this process. Once this pose estimate becomes more accurate, it will be used in the subsequent refinement of the segmentation.

This segmentation is performed following the approach presented in [13], which aims at classifying each pixel of an image by maximizing the posterior probability of a conditional random field considering multiple image features extracted at different quantization levels. This classifier was trained on 70 manually labeled images representing examples of the different classes of objects to be recognized. The obtained classifier is then run on each input panorama.

The accuracy of this classifier is in general good, but it may result in over or under segmentation of some regions. Despite these labeling errors, the segmentation is accurate enough as a starting point for the pose estimation problem.

B. Pose estimation

In order to register the panoramic images with respect to the 3D model, we use a generative approach aiming at finding the pose of the camera that generates building outlines as similar as possible to the ones computed during the segmentation process.

Let $\xi = (\theta, t)$ denote the pose of the camera shooting the spherical image, where θ and t denote the camera orientation and the camera position respectively. Camera orientation $\theta \in \mathbb{R}^3$ is encoded using an angle-axis representation, relative to a reference system oriented in such a way that its Y-axis points towards the north pole, and its Z-axis is parallel to the normal of the ground. The camera position $t \in \mathbb{R}^3$ is expressed in the same reference system, precisely in metric latitude, longitude and altitude. Camera extrinsic parameters corresponding to the pose ξ can then be simply computed using the exponential map as

$$E = \begin{bmatrix} e^{\hat{\theta}} & t^T \\ 0 & 1 \end{bmatrix}^{-1} \tag{1}$$

Let S denote the building outlines extracted from the panoramic image during the segmentation process, and let us assume that a pixel value of 1 indicates that the corresponding point on the panorama belongs to a building silhouette, and 0 otherwise. Given the current estimate for the camera pose ξ, the corresponding building outlines of the cadastral 3D model are generated by means of rendering. Let $B(\xi)$ denote this image. Ideally, for a correct pose estimate, the building outlines $B(\xi)$ should align perfectly with the outlines S. We therefore need to find the pose ξ which maximizes the overlap between these two images, S and $B(\xi)$, or in other words, we need to minimize the following functional

$$\arg\min_{\xi} \|S - B(\xi)\|_0 \tag{2}$$

where $\|\cdot\|_0$ represents the L0-"norm", counting the number of mismatching pixels in the two images.

Due to the large non-linearities present in this functional, we chose to use an evolutionary sampling technique to optimize it. In particular, we chose to use Particle Swarm Optimization (PSO) [14]. PSO achieves optimization through the simulation of the social interactions happening in a population of particles evolving over time, i.e., the swarm. These particles move freely in the solution space influenced by their personal experience (the individual factor) as well as, the experience of the other particles (the social factor).

PSO is a simple and easy to implement algorithm which was found to be very effective for optimizing the functional in Equation 2. Since a lot of renderings are involved during this optimization procedure, we speed up this process by implementing both the rendering of the building outline image $B(\xi)$, and the computation of the L0-norm on the GPU.

The optimization is initialized using the pose provided by StreetView. Since no information is available regarding the altitude of the camera, this is initialized as the altitude of the closest point on the ground plane of the cadastral 3D model. The swarm is then generated by adding noise to this initial pose. The search space for the optimization is constrained by limiting the particles to not move further than 20 meters and to not rotate more than 15 degrees. In particular, camera roll was restricted to ± 1 degree.

In order to account for the presence of occluders such as cars, pedestrians and vegetation, which may drastically change the shape of the building outlines in S, we identify them from the segmentation and create a binary mask representing these pixels with a value of 1. These pixels are then not considered during the computation of Equation 2. In this way, the algorithm does not penalize either the presence or the absence of a building in those pixels.

C. Segmentation Refinement

Once a good estimate for the pose ξ is obtained from the previous pose estimation, the building outlines S are refined using, as prior information, S itself and the building outlines of the model $B(\xi)$, rendered using the pose ξ. This refinement is then performed following the matting technique proposed in [15].

A tri-map is first generated by marking each pixel of the panorama as 'building', 'not building', or 'undecided', on the basis of S and $B(\xi)$. Specifically, we label a pixel as 'building' if the corresponding pixels in both S and $B(\xi)$ are marked as 'building' (i.e., 1). On the contrary, we label a pixel as 'not building' when the corresponding pixels in both S and $B(\xi)$ are marked as 'not building' (i.e., 0). The remaining region is marked as 'undecided' and it is expanded with a dilation operator of radius 21 pixels to increase the uncertainty on the labeling. The matting algorithm then builds a local appearance model for both the 'building' and the 'not building' regions, and decides whether the 'undecided' pixels belong to a building or not.

V. Results

We ran our algorithm on a total of 14000 StreetView images spanning different urban scenarios from residential areas, to commercial areas, and outskirts (see Figure 3(left)).

The cadastral 3D model was obtained from the city administration in the form of a triangular mesh, where buildings were represented using templates encoding features like roofs, attic windows and chimneys. Balconies and streetside windows were often either missing from this model or inaccurately represented. Nevertheless, the maximum error on the model does not exceed 50 cm.

We ran our algorithm using the same settings for all the input images. The pose estimation and the segmentation refinement loop was repeated three times per image. In the first two stages, the optimization was run only for translation and yaw angle (i.e., the rotation about the vertical axis). Only at the final stage, the optimization was performed on all the six degrees of freedom. This choice was made to compensate for the fact that majority of the error in such data resides in the position, while the orientation is relatively accurate, particularly the pitch and the roll. Therefore, we preferred to optimize first for only the position and the yaw, to avoid over-fitting of the pose on to a coarse and generally not so accurate initial segmentation.

In each step, PSO was run on 80 particles for 90 iterations. The initial swarm noise was set to 7 meters in translation, and 6 degrees for rotation. This noise is reduced to half, for the second and the third step. The processing time on a single core machine was 8 minutes per image.

Pose estimation success rate was evaluated visually by observing the overlay of the 3D model onto the images using the estimated pose. Precisely, we considered an image 'incorrectly registered' if the projection of the model was more than 40 pixels away from the building contour. On the tested images, 74.8% were accurately registered by the algorithm, while only 8.7% were incorrectly registered. In the remaining 16.5% of the images, there was not enough visual information for us to decide if the pose was correctly estimated or not (as an example, when majority of the scene was occluded by trees).

The graph in Figure 3(right) shows the average residual computed over the tested 14000 images in all the 90 PSO iterations, at each individual step of the refinement process. The residue value indicates the percentage of pixels that were found to be misaligned at each evolution. The residue drops quickly during the first 30 iterations, and then reduces gradually over the next iterations. After the first run of PSO, the percentage dropped from approximately 11% to 8.1%. Since, the refined building outlines (after matting) are used as input for step 2 and 3 of the process, the residue drops down to 5.2% and finally to 3.9% at the end of step 3.

The graph in Figure 4 shows the camera pose corrections estimated with our method in both translation and rotation.

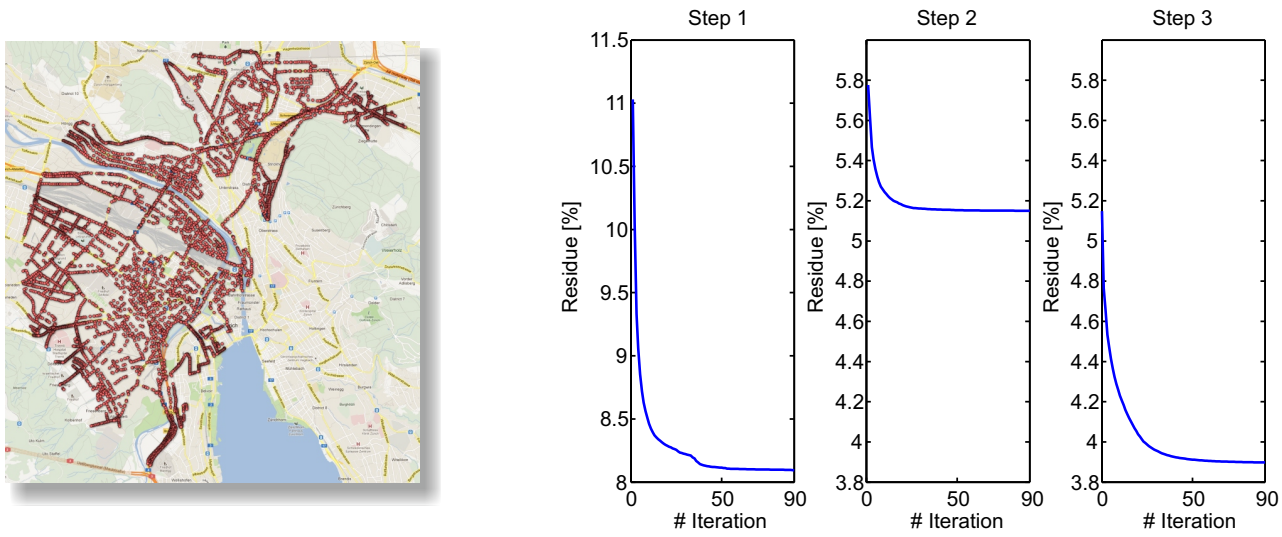


Figure 3. (Left) Coverage of the images used for the experiments displayed on the map. (Right) Average residual error (Eq. 2) obtained during the 90 PSO iterations at each refinement step. The residue is visualized as the percentage of misaligned pixels.

On an average, the computed corrections have a standard deviation of 3.7 meters for the translation, and 1.9 degrees for the rotation.

Figure 6 shows, for nine different cases, the images obtained by rendering the 3D model from the initial camera pose (left column) and the images obtained by rendering the 3D model from our refined camera pose (right column).

The first three images in the left column were captured in commercial areas, the next two were captured in the countryside and the remaining images were captured in residential areas. It can be seen that the algorithm performs very well for the first three images, specifically, in the images on the right column, the edges around the windows on the top of the building match perfectly with those on the model. For images number 4 and 5 captured in the countryside, it can be noted that despite the fact that majority of the scene is occupied by vegetation, the algorithm is able to register the images well. Images 6 and 7 demonstrate the typical scenario of residential areas, where vegetation and vehicles appear frequently, but only partially occluding the buildings.

Lastly, for the case of images numbered 8 and 9, the initial pose estimate from Google has a very big error. Moreover, there are major occlusions caused by trees, and in fact, the initial segmentation did not indicate the presence of buildings in those regions. Despite this, the algorithm performs reasonably well, but clearly the resulting images are still not perfectly aligned with the model. Please refer to the supplementary video for more results.

Comparison with groundtruth: To quantitatively evaluate the accuracy of the proposed algorithm with respect to noise in the input data, we conducted an additional experiment on some images of the original dataset. Precisely, we chose some well aligned panoramic images and we added structural noise to their initial segmentations. In particular, circles of radius varying between $30-60$ pixels were added and removed at random positions from these segmentations to simulate errors in the pixel classification (see Figure 5(b) for an example). A uniform noise between ± 10 degrees and ± 5 meters was then added to the correct pose of the image to simulate inaccuracies in the initial pose provided by StreetView.

The table in Figure 5 shows the statistics of the errors obtained for this experiment for varying quantities of structural noise. It is visible that significant errors in the initial segmentation do not influence much the final accuracy of the algorithm.

VI. Conclusions

We presented an algorithm to automatically refine the pose of spherical panoramic images, such as the ones in Google StreetView, leveraging the cadastral 3D information typically available in urban environments. These images were first registered with respect to the cadastral 3D model by aligning the building outlines. Since this model is accurately geolocated, the registration process results in an accurate geolocation of the images as well.

To overcome the difficulties encountered while extracting building outlines in natural images and to deal with occluders frequently occuring in urban scenarios such as vegetation, vehicles and construction sites, we propose an iterative optimization technique aimed at jointly estimating the camera pose and the building outlines.

We evaluated our algorithm on 14000 StreetView images spanning commercial and residential locations, as well as the outskirts of a city, where vegetation is predominant in the images. We showed that even with very few buildings

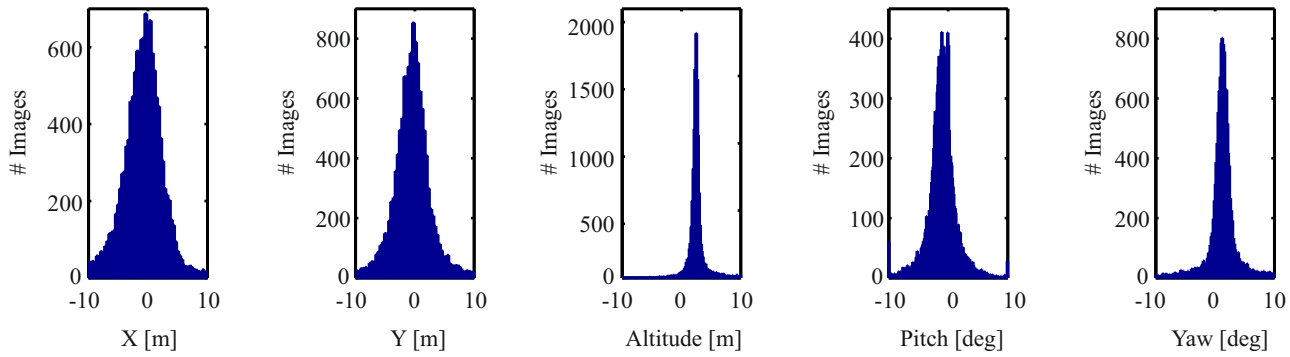
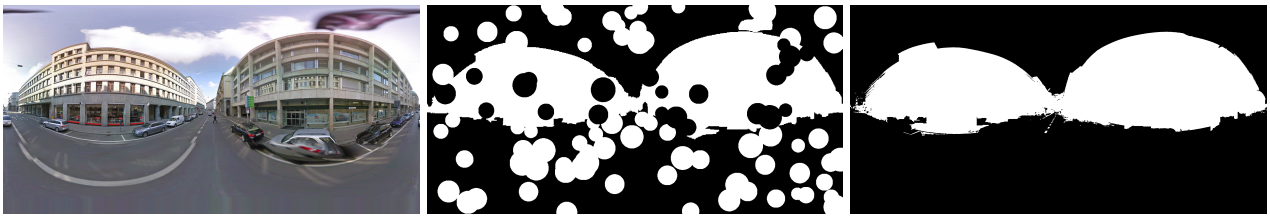


Figure 4. Histograms of orientation and position corrections estimated using our algorithm.



(a) (b) (c)

Structural noise (# Circles added/removed)	μ (Rotation) [deg]	σ (Rotation) [deg]	μ (Translation) [m]	σ (Translation) [m]
60	0.9328	0.4104	0.5530	0.1443
80	0.9722	0.4222	0.5825	0.1165
100	1.0038	0.476	0.583	0.1184

Figure 5. Quantitative evaluation of the performance of the algorithm with respect to noise in the input data. (a) One of the tested panoramas. (b) Initial Segmentation with structural noise. (c) Resulting Segmentation. The table reports the mean (μ) and standard deviation (σ) of errors in rotation and translation for varying structural noise.

visible in the images (see image number 4 and 5 in Figure 6) we were able to perform a very good registration.

On an average, the correction in the position has a standard deviation of 3.7 meters and 1.9 degrees in rotation. While these numbers may seem low in magnitude, they are actually reasonably high if these images are expected to be used for geo-applications that demand high accuracy on the model as well as the images.

As a conclusion, we proposed a simple and easy to implement technique to refine the registration of panoramic images, such as the ones available from Google StreetView. We believe that this will enable the usage of this humongous source of publically available data, opening up a wider range of applications than is possible now.

Acknowledgment

The research leading to these results has received funding from the ERC grant #210806 4DVideo under the EC's 7th Framework Programme (FP7/2007-2013), and from the Swiss National Science Foundation and Google.

References

[1] M. Pollefeys, D. Nister, and J. M. Frahm et al, "Detailed real-time urban 3d reconstruction from video," *IJCV*, vol. 78, 2008.

[2] C. Fruh and A. Zakhor, "Constructing 3-d city models by merging aerial and ground views," *IEEE Computer Graphics and Applications*, 2003.

[3] S. Christy and R. Horaud, "Iterative pose computation from line correspondences," *Journal of Computer Vision and Image Understanding*, 1999.

[4] L. Liu and I. Stamos, "Automatic 3d to 2d registration for the photorealistic rendering of urban scenes," in *CVPR*, 2005.

[5] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand, "Geolocalization using skylines from omni-images," in *ICCV Workshops*, 2009.

[6] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *International Conference on Computer Vision*, 2011.

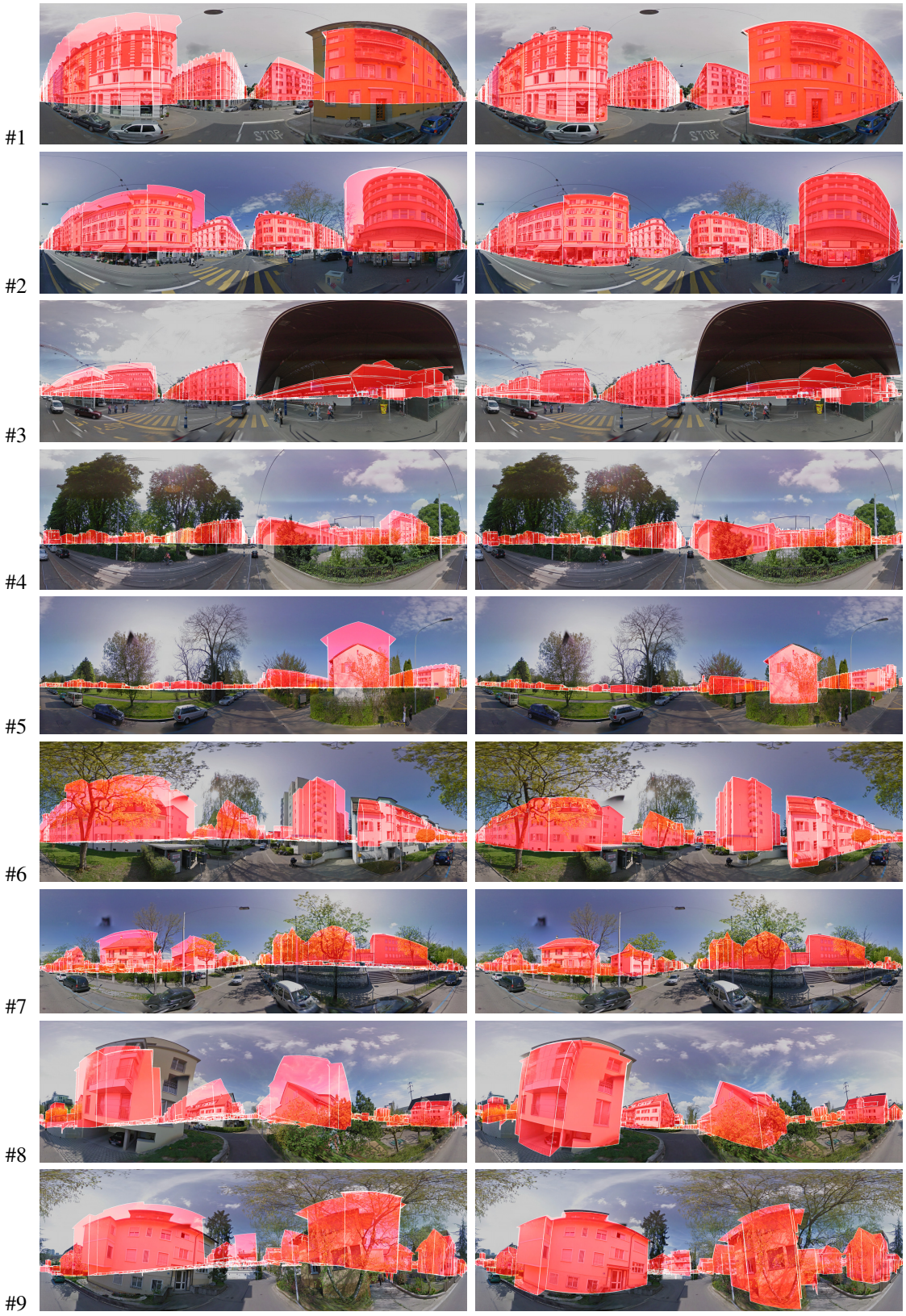


Figure 6. Overlay of some Google StreetView images with the cadastral 3D model before (left) and after applying our algorithm (right).

[7] J. Knopp, J. Sivic, and T. Pajdla, "Avoding confusing features in place recognition," in *Proceedings of the European Conference on Computer Vision*, 2010.

[8] W. Zhao, D. Nister, and S. Hsu, "Alignment of continuous video onto 3d point clouds," *PAMI*, 2005.

[9] P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome, "Towards geographical referencing of monocular slam reconstruction using 3d city models: Application to real-time accurate vision-based localization," *PAMI*, 2009.

[10] T. Pylvanainen, K. Roimela, R. Vedantham, J. Itaranta, and R. Grzeszczuk, "Automatic alignment and multi-view segmentation of street view data using 3d shape prior," in *3DPVT*, 2010.

[11] A. Mastin, J. Kepner, and J. Fisher, "Automatic registration of lidar and optical images of urban scenes," in *CVPR*, 2009.

[12] M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes, "Using photo-consistency to register 2d optical images of the human face to a 3d surface model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[13] L. Ladick, C. Russell, P. Kohli, and P. H. Torr, "Associative hierarchical crfs for object class image segmentation," in *International Conference on Computer Vision*, 2009.

[14] M. Clerc and J. Kennedy, "The particle swarm explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, 2002.

[15] A. Levin, D. Lischinski, and Y. Weiss, "A closed form solution to natural image matting," in *Computer Vision and Pattern Recognition*, 2006.