City-Scale Change Detection in Cadastral 3D Models using Images

Aparna Taneja
ETH Zurich
aparna.taneja@inf.ethz.ch

Luca Ballan
ETH Zurich
luca.ballan@inf.ethz.ch

Marc Pollefeys
ETH Zurich
marc.pollefeys@inf.ethz.ch

Abstract

In this paper, we propose a method to detect changes in the geometry of a city using panoramic images captured by a car driving around the city. We designed our approach to account for all the challenges involved in a large scale application of change detection, such as, inaccuracies in the input geometry, errors in the geo-location data of the images, as well as, the limited amount of information due to sparse imagery.

We evaluated our approach on an area of 6 *square kilometers inside a city, using* 3420 *images downloaded from Google StreetView. These images besides being publicly available, are also a good example of panoramic images captured with a driving vehicle, and hence demonstrating all the possible challenges resulting from such an acquisition. We also quantitatively compared the performance of our approach with respect to a ground truth, as well as to prior work. This evaluation shows that our approach outperforms the current state of the art.*

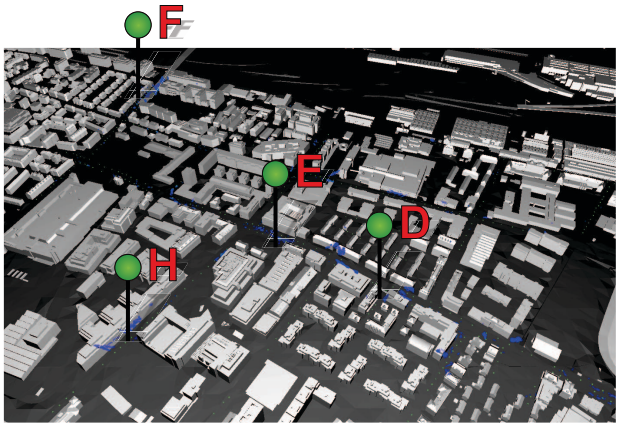


Figure 1. Changes detected on the cadastral 3D model of a city using panoramic images. The detected changes are marked in blue, while the locations of the input images are represented as green points. Green markers indicate some of the changed locations recognized using our approach. For the corresponding images please refer to Figure 5.

1. Introduction

Motivated by the vast number of services benefiting from 3D visualizations of urban scenarios, a lot of work has taken place in the recent past, to obtain accurate 3D reconstructions of cities. Many efficient techniques have been proposed to obtain such models from imagery and/or range measurements captured from groundbased vehicles [2], as well as aerial platforms [5]. In fact, most city administrations already maintain such information for cadastral applications such as city planning, real estate evaluation and so on.

However, cities are dynamic in nature, evolving over time, with new buildings being constructed and old ones taken down [6]. As these changes occur, any previous reconstructions do not comply with the current state of the city and need to be updated accordingly. A naive way to update these reconstructions is to again collect data all over the city, and rebuild the 3D model from scratch. Clearly, capturing such high quality data with laser scanners or with high resolution cameras on the scale of a city is not feasible on a frequent basis.

Recent works like [15] proposed to efficiently perform this update task by first localizing in the environment the areas where geometric changes have occurred, and then by running the high quality data collection selectively only on those locations where significant changes have been detected. Their work showed convincing results on multiple urban scenarios detecting changes from images.

However, the evaluated locations were all spatially constrained, and while some suggestions were presented to make the approach scalable to large environments, it needs to be adapted significantly to address the different challenges involved in a city scale application of change detection. Namely,

- **Inaccuracies in the cadastral 3D model**: Cadastral information, maintained by city administrations, is typically encoded as 3D mesh models representing the main constructions in the city. Since their main ap-

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.22

plication is planning and monitoring, their accuracy is reasonably high. However, their level of detail is quite basic with simple bounding boxes, approximating the buildings shapes, augmented sometimes with features, like roofs and chimneys.
A large scale change detection algorithm therefore, needs to differentiate between real changes in the geometry and changes induced by inaccuracies in these cadastral 3D models.

In the envisioned scenario of a city scale change detection application and model update, images depicting the current state of the city are captured as panoramic images, from cars driving around the city. Several commercial systems have been deployed to capture such kind of imagery, such as Google StreetView and Microsoft StreetSide. The data acquired with such systems however, presents two big challenges when used in the context of change detection, namely:

- **Inaccuracy in the geo-location information**: The geo-location information tagging these images is typically provided by GPS and IMU units mounted on the car. However, the data recorded with such devices is typically noisy, and while the position and orientation inaccuracies may be tolerable for applications such as street navigation, they are definitely not for the purpose of change detection.

- **Sparsely captured imagery**: Since the acquired images are not just representing a few streets in an urban environment, but actually entire cities, the spatial capturing rate of these images might not be very dense. Therefore a building well visible in one image, will be only partially visible in a nearby image.

  A large scale change detection algorithm needs to be able to cope with such sparse imagery.

In this paper, we propose a method to detect changes in the geometry of a city. While our formulation builds on the work of [15], we explicitly address the challenges involved in a large scale application of change detection. In particular, we use cadastral 3D models provided by the city administration and panoramic images captured all over the city. For our experiments we used the Google StreetView images which, besides being publicly available, are also a good example of panoramic images captured with a driving vehicle on the scale of a city.

2. Related Work

There has been a lot of work in the field of change detection mostly focusing on comparing images of a scene captured at an earlier time instant with images captured later [12]. Such a comparison is usually sensitive to changes in illumination and weather conditions across the old and the new images. To partially overcome these issues [10] proposed to learn, from the old images, a probabilistic appearance model of the 3D scene. This is then used for comparison with the new images. As an alternative, [3] proposed to detect changes based on the appearance and disappearance of 3D lines in the scene.

All of these methods however, focus on detecting general changes in the appearance of a scene. These changes may or may not correspond to changes in the geometry. On the other hand, [15] proposed a method to detect only the geometric changes occurring in an environment. Their method is based on the assumption that if a pair of images represents a 3D model exactly, then these images must project consistently one into the other. Viceversa, if this projection reveals inconsistencies then the geometry represented in the images is different from the original one.

In this paper, we extend their approach to account for the challenges involved in a city scale application of a change detection algorithm, as mentioned in the introduction.

3. Change Detection

Given a cadastral 3D model of a city and a set of panoramic images depicting its current state, the goal of the proposed algorithm is to detect geometric changes that may have occurred between the time the 3D model was built and the time the new images were captured.

We perform this task following an approach similar to the one proposed in [15]. For the reader's convenience, we briefly recall, in this section, the major concepts presented in [15], namely, the inconsistency map and the used probabilistic framework.

For each pair of images I_s and I_t observing a location in the environment, the geometry of the environment is used to project the source image I_s into the point of view of the target image I_t. The resulting image projection, denoted as $I_{t \leftarrow s}$, is then compared with the original target image I_t to obtain a pixel-wise map of inconsistencies between the geometry and the images I_s and I_t. This map is referred to as the inconsistency map, and is denoted with the symbol $M_{t \leftarrow s}$. Formally, $M_{t \leftarrow s} = |I_{t \leftarrow s} - I_t|$, where $|\cdot|$ represents the pixel-wise absolute differences between $I_{t \leftarrow s}$ and I_t.

In principle, if the two images I_s and I_t are consistent with the geometry, then the resulting inconsistency map $M_{t \leftarrow s}$ is zero everywhere. Viceversa, in case of a change, some of its values might differ from zero. In order to localize the occurred changes in the city, the entire city is discretized into a grid of uniformly sized voxels, precisely of size $1\ m^3$ each.

The goal of the change detection algorithm is to estimate a binary labeling $\mathcal{L} = \{l_i\}_i$ for each voxel i in this grid, indicating the presence, or the absence, of a change inside that voxel (with $l_i = 1$ and $l_i = 0$, respectively). An estimate

for this labeling can be obtained by maximizing the posterior probability of $\mathcal{L}$ given the input images $\mathcal{I} = \{I_k\}_k$ as observation. By using the Bayes' rule, this corresponds to

$$P\left(l_i|\mathcal{I}\right) = \frac{P\left(\mathcal{I}|l_i\right)P\left(l_i\right)}{P\left(\mathcal{I}\right)} \tag{1}$$

where the generative model $P\left(\mathcal{I}|l_i\right)$ is computed on the basis of the inconsistency maps. Precisely as

$$P\left(\mathcal{I}|l_i\right) = \prod_{t,s} P\left(M_{t\leftarrow s}|l_i\right) \tag{2}$$

where the probability $P(M_{t\leftarrow s}|l_i)$ is modeled by a uniform distribution U in case of a change, and by a truncated Gaussian distribution centered in 0 in case of no change, i.e.

$$P\left(M_{t\leftarrow s}\left(q\right) = x|l_i\right) = \begin{cases} H\left(x\right)\frac{2}{\sigma_c\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma_c^2}} & l_i = 0 \\ U & l_i = 1 \end{cases} \tag{3}$$

where $H\left(x\right)$ is the Heaviside step function, and q is a generic pixel of $M_{t\leftarrow s}$. Since changes corresponding to vehicles, pedestrians and vegetation are not relevant for the purpose of updating a 3D model, a classifier is used to recognize those classes of objects in the images [4]. Pixels belonging to those classes are then not considered during the change inference process.

This approach on its own is however not sufficient to deal with the challenges involved in a large scale application of change detection, such as geometric inaccuracies, geolocation information inaccuracies, and wide baseline imagery. The following sections address all these challenges and propose a solution to cope with each of them.

3.1. Inaccuracies in the geo-location information

In a scenario where a car is driving around capturing panoramic images in a city, the geo-location information, providing the position and orientation where each of these images were taken, is typically captured using sensors like GPSs and IMUs. The data recorded by these sensors is in general noisy, with errors being on the order of ± 5 meters in the location and ± 5 degrees in the orientation.

One way to refine these estimates is to exploit the available 3D model and register each image with respect to it. Different approaches have been proposed in literature to perform this task. However, registration techniques based on image feature descriptors, e.g. [14], cannot be applied in this case due to the typical absence of texture information in the cadastral 3D models. Other registration techniques, like the ones based on rigid [17] and non-rigid ICP [9, 11], are also not applicable since they would require a 3D reconstruction from the acquired images. This in general cannot be achieved due to the sparse sampling nature of the captured images.

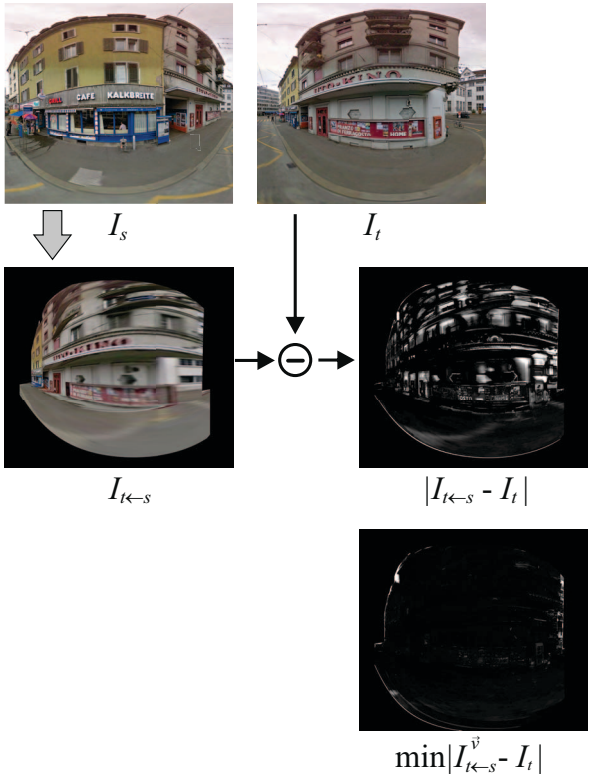


Figure 2. Inconsistency maps corresponding to the pair of images I_s and I_t, obtained using the approach of [15] ($|I_{t\leftarrow s} - I_t|$) and the one obtained using Equation 6 (below), accounting for geometric inaccuracies. False changes due to missing details on the building facade disappear in the latter.

An alternative is provided by [8, 13, 16]. For each panoramic image, an object class segmentation is performed in order to estimate the building outlines in these images [7]. Let S_t denote the building outlines estimated on the image I_t. Each pixel of S_t is labeled as 1 in case the pixel belongs to a building, and 0 otherwise. Let ξ_t represent the current estimate for the pose of image I_t (the geolocation information), and let $B(\xi_t)$ denote the corresponding building outlines obtained by rendering the cadastral 3D model at pose ξ_t. Ideally, at the correct pose estimate, the building outlines $B(\xi_t)$ align perfectly with the actual outlines S_t. Formally, the correct pose ξ_t corresponds to the minimization of the following functional:

$$\underset{\xi_t}{\operatorname{argmin}} \ \|S_t - B(\xi_t)\|_0 \tag{4}$$

where $\|\cdot\|_0$ represents the L0-"norm", counting the number of mismatching pixels in the two images. In order to be robust with respect to the presence of occluders such as cars, pedestrians and vegetation, this score is not evaluated for pixels belonging to one of these classes.

3.2. Our registration approach

In general, minimizing for Equation 4 results in an accurate registration of the input images with respect to the cadastral 3D model. However, while the individual errors in the registration might be small, these errors quickly accumulate during the reprojection process. Since the proposed change detection algorithm bases its inference on the reprojected images $I_{t\leftarrow s}$, even small errors in the registration are not tolerable, since they will generate false evidence of a change in the inconsistency maps $M_{t\leftarrow s}$.

Minimizing for Equation 4 is therefore insufficient for our purpose, and a registration technique accounting also for the relative alignment between neighboring images, needs to be designed.

We do this, by adding an extra term in Equation 4 accounting for the reprojection error $M_{t\leftarrow s} = |I_{t\leftarrow s} - I_t|$ between neighboring images. We then perform the pose estimation over a window of $n = 5$ consecutive panoramic images. Precisely, let $I_1, \ldots, I_n$ be n consecutive images, and let $\xi_1, \ldots, \xi_n$ represent their related pose parameters, the joint registration of these images is obtained by minimizing the following functional

$$\underset{\xi_1,\ldots,\xi_n}{\operatorname{argmin}} \sum_{t\in[1,\ldots,n]} \left[\|S_t - B_t(\xi_t)\|_0 + \sum_{s\in[1,\ldots,n]} \|M_{t\leftarrow s}\|_1 \right] \tag{5}$$

where $\|M_{t\leftarrow s}\|_1$ represents the sum of all the pixel-wise absolute differences between the images $I_{t\leftarrow s}$ and I_t.

This joint minimization considers both the individual alignment error, of an image with the 3D model, and the relative alignment error of an image with respect its neighbors. This makes the pose estimation more robust to outliers, such as changes in the geometry and/or segmentation errors in the images.

Due to the extreme non linearities in Equation 5, this minimization is performed using a sample based technique. In particular we used Particle Swarm Optimization [1], which is an evolutionary algorithm computing the evolution of a swarm of particles influenced at each iteration by the particle's own experience (the cognitive factor) and also by the swarm's experience (the social factor).

3.3. Dealing with geometric inaccuracies

Cadastral 3D models typically show low level of detail. In fact, while these models correctly represent the volume of the buildings in a city as bounding boxes augmented with simple features like roofs and chimneys, details like balconies, streetside windows, extended roofs, and in general any protruding structures on the building facades, are typically missing or inaccurately represented. Consequently, the projections of each of these structures from one image into another can result in high inconsistency values in the $M_{t\leftarrow s}$ maps. This consequently degrades the detection performance by increasing the number of false detections.

To account for these geometric inaccuracies, we draw multiple hypotheses on the real extent of the missing or the inaccurately represented structures, by shifting the building walls on the ground plane. For each of these hypotheses, the corresponding inconsistency map is computed. In principle, the inconsistency map $M_{t\leftarrow s}$ resulting from a geometry which perfectly represents the actual building corresponds to the pixel-wise minimum of the individual inconsistency maps produced by each hypothesis. Formally, let $I^{\vec{v}}_{t\leftarrow s}$ be the image projection obtained by translating the building walls by a vector $\vec{v}$, where $\vec{v}$ is a vector on the ground plane. Then the inconsistency map resulting from a perfectly represented geometry is

$$M_{t\leftarrow s} = \min_{\vec{v}\in\mathcal{S}} |I^{\vec{v}}_{t\leftarrow s} - I_t| \tag{6}$$

where $|\cdot|$ indicates the pixel-wise absolute values, and $\mathcal{S}$ represents the set of translation vectors $\vec{v}$ used to compensate for the protruding structures. In particular, we set $\mathcal{S}$ equal to all the possible ground translations of an amount smaller than 0.5 meters. The computation of the $I^{\vec{v}}_{t\leftarrow s}$ and the $M_{t\leftarrow s}$ images is done on the GPU, making this process very fast.

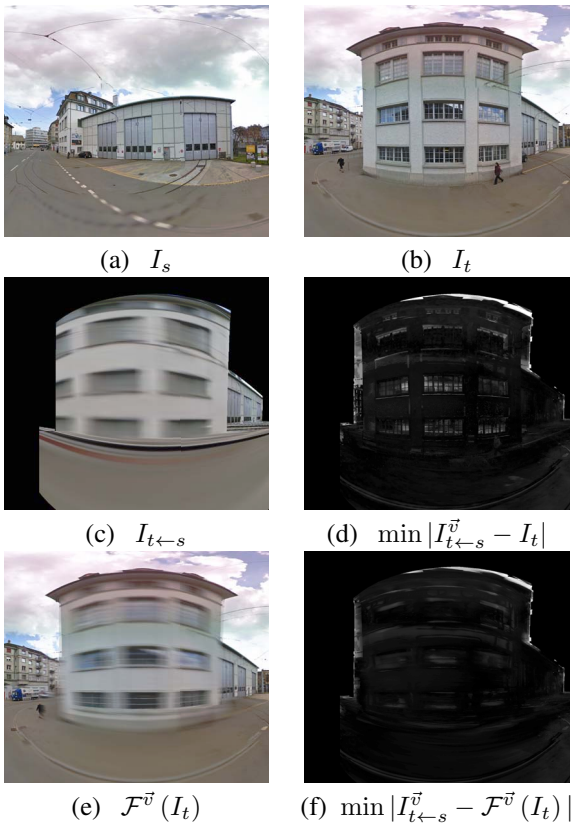
Figure 2 shows the effects of the usage of this approach on the generated $M_{t\leftarrow s}$ maps in a scenario where the balconies and the extended roof of a building facade were missing from the 3D model. It is visible, in the bottom image, that false inconsistencies disappear when multiple hypotheses are evaluated for the location of these elements.

3.4. Dealing with sparse imagery

While the multiple hypotheses approach introduced in the previous section allows us to account for small inaccuracies in the cadastral 3D model, another issue needs to be considered when projecting images captured very far apart.

In these cases in fact, high perspective distortions and image sub-samplings corrupt the reprojected image $I_{t\leftarrow s}$ by generating blurring artifacts (Figure 3(c)), and consequently decreasing the accuracy of the detector by generating more false positives (Figure 3(d)). In fact, in these situations, a pixel in the source image I_s does not project into a unique pixel in the target image plane I_t, but instead, into multiple ones, causing the blurring.

A work around to this problem is to avoid comparing images that are farther than a certain distance. This however would also reduce the amount of information at our disposal for the change detection inference. Since we already have a limited amount of data observing the same location, due to the sparse imagery, we need to use all possible images inside a certain radius even if that means considering images captured more than 30 meters apart.



(a) I_s (b) I_t (c) $I_{t\leftarrow s}$ (d) $\min|I^{\vec{v}}_{t\leftarrow s} - I_t|$ (e) $\mathcal{F}^{\vec{v}}(I_t)$ (f) $\min|I^{\vec{v}}_{t\leftarrow s} - \mathcal{F}^{\vec{v}}(I_t)|$

Figure 3. Example scenario where the source image I_s was captured more than 30 m away from the target image I_t. (c) Reprojected image. (d) Inconsistency map obtained as a result of Equation 6. (e) Image obtained after filtering I_t with the spatially varying kernel defined in Section 3.4. (f) Inconsistency map obtained as result of Equation 7.

Therefore, we chose to explicitly account for the artifacts generated in case of large baselines, by simulating them also in the target image I_t. Precisely, we estimate the shape that each pixel of the source image I_s would have in the target image I_t. This can be easily performed on the GPU by approximating the original pixel shape with a circle of radius 0.5 pixel units. Its projection on the target image would result in an ellipse centered on a point p. This ellipse describes the amount of blur that the image $I_{t\leftarrow s}$ is affected by in p.

Therefore, to better compare the reprojected image $I_{t\leftarrow s}$ with the target image I_t, we simulate in I_t the same blurring artifacts as in $I_{t\leftarrow s}$, by applying to each pixel of I_t a spatial filter shaped accordingly to the ellipse projecting into p. Basically, this corresponds to filtering I_t with a spatially varying kernel. Let $\mathcal{F}^{\vec{v}}(I_t)$ be the image resulting after this process assuming a translation vector of $\vec{v}$ for the geometry. Equation 6 becomes

$$M_{t\leftarrow s} = \min_{\vec{v}\in\mathcal{S}} |I^{\vec{v}}_{t\leftarrow s} - \mathcal{F}^{\vec{v}}(I_t)| \tag{7}$$

Figure 3(f) and (d) show the $M_{t\leftarrow s}$ images obtained with and without the filtering operation. It is visible that accounting for these distortions/blurring artifacts significantly improves the $M_{t\leftarrow s}$ image by eliminating the false inconsistencies caused by the large baseline between the images.

3.5. Additional cue: building outlines consistency

To improve the performance of our detection algorithm, we introduce an additional cue to the original generative model $P(\mathcal{I}|l_i)$ of Equation 2, accounting for building outlines consistency. In principle, in case of no change, not only should the inconsistency $M_{t\leftarrow s}$ maps be zero, but the corresponding building outlines seen in the images should be consistent with those in the geometry as well.

Formally, let C_t be the image representing the pixel-wise inconsistencies between the building outlines estimated from the image I_t and the outlines of the 3D model visible from the point of view of I_t, i.e.,

$$C_t = |S_t - B(\xi_t)| \tag{8}$$

where $|\cdot|$ indicates the pixel-wise absolute value. Ideally, in case of no change, C_t is zero everywhere. Viceversa, in case of a change, some of its values might differ from zero. We model this behavior by updating the generative model $P(\mathcal{I}|l_i)$ of Equation 2 as follows

$$P(\mathcal{I}|l_i) = \prod_{t,s} P(M_{t\leftarrow s}|l_i) \prod_{t} P(C_t|l_i) \tag{9}$$

While the first series of products indicate the independence between the image formation process of the different inconsistency maps $M_{t\leftarrow s}$, the second series of products underlines the independence between the image formation process of the building outlines seen from the different images I_t. Further, assuming that the conditional probability of C_t given a voxel label l_i is only influenced by the pixels in the footprint of voxel i on C_t, we introduce an additional random variable η^i_t representing the fraction of incorrectly labeled pixels in this footprint. Formally, given $\eta^i_t = \frac{1}{N}\sum C_t(q)$, $P(C_t|l_i)$ is equal to $P(\eta^i_t|l_i)$ and

$$P(\eta^i_t = x|l_i) = \begin{cases} H(x)\,\frac{2}{\sigma_s\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_s^2}} & l_i = 0 \\ U & l_i = 1 \end{cases} \tag{10}$$

where H is same as in Equation 3.

As observed earlier, inaccuracies in the geometry might lead to false inconsistencies in C_t. To cope for this, we adopt a similar approach as was proposed for the $M_{t\leftarrow s}$ maps, that is, we define C_t similarly as in Equation 6, i.e.

$$C_t = \min_{\vec{v}\in\mathcal{S}} |S_t - B^{\vec{v}}(\xi_t)|. \tag{11}$$

4. Results

The proposed approach was evaluated on an area of 6 square kilometers (about 2.31 square miles) inside a city. In total, 3420 panoramic images were used to detect changes in this environment.

In particular, we used images downloaded from Google StreetView. Each of these images consists of a full spherical panorama with resolution generally up to 3328×1664 pixels, covering a field of view of 360 degrees by 180 degrees. In the tested location, these images were captured on an average once every 10 meters, although this distance increased in some regions. Since the primary application of these images is street navigation their quality is, in general, not very high. In fact, besides being low resolution, they display numerous artifacts mainly due to blending errors, and moving objects.

The geo-location data for each panoramic image was obtained also from the Google StreetView service. Since this data is in general too inaccurate for the purpose of change detection, showing errors with a standard deviation of 3.7 meters in translation, and 2 degrees in orientation [16], this was refined using the method proposed in Section 3.2. Precisely, Equation 5 was optimized using an initial swarm noise of 7 meters in translation and 6 degrees in rotation.

The cadastral 3D model was instead obtained from the city administration, and its claimed accuracy was 0.5 meters. For this reason we chose the translation vectors in $\mathcal{S}$ to have a magnitude of up to 0.5 meters.

We learnt the parameters of our detector on a small subset of the available data. In particular, the object class classifier was trained on 70 manually labeled images chosen randomly across the city. Similarly, the parameters σ_c and σ_s, modeling the color and building outline consistency respectively (see Equation 3 and Equation 10), were estimated on another set of 75 images where each pixel was manually labeled as change or no change. The probability distribution of the corresponding $M_{t \leftarrow s}$ and C_t maps was then estimated from these images. In the end a Gaussian distribution was fit onto those distributions.

Figure 5 and Figure 1 show the changes detected by our approach on two small regions of the processed cadastral 3D model. The green dots denote the locations of the input panoramas, while the blue dots represent voxels labeled as change. The green markers act as a reference for the images below. Each of those images shows the cadastral 3D model (red) overlaid on one of the input panoramic images captured at that location.

It is visible that a high density of the blue voxels in the map corresponds to a change revealed by the input images. For instance, location (A) depicts a scenario where more floors were added to a building. In the map in fact, blue voxels can be seen on the top of the corresponding building.

Locations (B), (E) and (F) show three scenarios where an entire building had been constructed since the model acquisition. In particular (F) shows a building under construction.

Locations (C) and (D) reveal relatively small changes corresponding to a new roof, and a connecting hallway between buildings, respectively. Updating the model with such details might be useful, for instance to generate a warning if a large truck has to pass through this street.

Locations (G) and (H) instead show two examples of false changes that were detected due to trees (mislabeled as building by the classifier), and due to strong reflections, respectively.

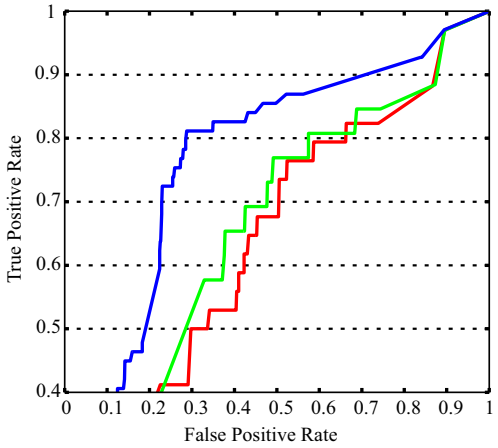


Figure 4. Evaluation of the algorithm performance. (Blue) ROC curve obtained using our method. (Red) ROC obtained using the method proposed in [15]. (Green) ROC obtained using the method of [15] also incorporating the refinement of section 3.4.

4.1. Quantitative evaluation and comparison with prior work

We generated ground truth data by manually labeling each panoramic image as corresponding to a change or not. In particular, for this experiment, we focused only on a restricted subset of the original dataset consisting of 1000 images. The labeling was performed on the basis that, an image represents a change if an actual change in the geometry was visible from approximately 25 meters distance. On this particular subset of the dataset, 76 images were labeled as change.

We compared this ground truth with the results obtained using our change detection algorithm. Precisely, using the same labeling methodology as for the ground truth, an image was labeled as corresponding to a change if a sufficient number of voxels were detected as change in a radius of 25 meters from the image location. This threshold was set to 30 voxels in our experiments.

The ROC curve in Figure 4 shows the performance of our algorithm (blue curve). In the same figure, the red curve

shows the performance of the method proposed in [15]. Precisely, we ran this method on exactly the same data (images + registration). For a fair comparison, we also incorporated the building outline consistency term of Section 3.5 into their approach. The green curve instead shows the performance of this method also incorporating the robustness against distortion effects (section 3.4).

It is visible that for the same number of true positives, our approach results in much less false detections. Our approach, in fact benefits from the considerations made in Section 3.3 and in Section 3.4, making it more robust to inaccuracies in the geometry and to wide baseline imagery.

5. Conclusions

In this paper, we proposed a method to detect changes in the geometry of a city using panoramic images captured by a car driving around the city. We extended the work of [15] to account for all the challenges involved in a large scale application of change detection.

In particular, we showed how to deal with the geometric inaccuracies typically present in a cadastral 3D model, by evaluating different hypotheses on the correct geometry of the buildings contained in it.

We showed how to deal with errors in the geo-location data of the input images, by proposing a registration technique aimed at minimizing the absolute alignment error of each image with respect to the 3D model, as well as the relative alignment error with respect to its neighboring images.

To cope for the limited amount of images observing a location, we proposed a robust comparison method explicitly compensating for the image sub-sampling artifacts and the high perspective distortions resulting in case of large baseline imagery. To further improve the detection accuracy, we proposed to use building outlines as an additional cue for our change detection inference.

The performance of our algorithm was evaluated on the scale of a city (6 square kilometers area) using 3420 images downloaded from Google StreetView. These images, besides being publicly available, are also a good example of panoramic images captured with a driving vehicle on the scale of a city. This dataset is known to be very challenging due to the sparse capturing rate (on an average every 10 meters), their low resolution, the blending artifacts, and their inaccurate geo-location data.

On a quantitative evaluation our algorithm outperformed the current state of the art (see Figure 4). However, as is clearly visible, there is still space for improvement due to the relative large number of detected false positives. This is mainly due to strong reflections and errors in the segmentation, particularly on trees (and especially those without foliage, as in Figure 5G). A bigger training set accounting for different appearance of trees across seasons would definitely improve the performance of the algorithm. Another improvement can be obtained by detecting windows as well which are typical sources of reflections.

Acknowledgment: The research leading to these results has received funding from the ERC grant #210806 4DVideo under FP7/2007-2013, SNSF, and Google.

References

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

[2] N. Cornelis, B. Leibe, K. Cornelis, and L. Gool. 3d urban scene modeling integrating recognition and reconstruction. *IJCV*, pages 121–141, 2008.

[3] I. Eden and D. B. Cooper. Using 3d line segments for robust and efficient change detection from multiple noisy images. In *ECCV*, 2008.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010.

[5] C. Fruh and A. Zakhor. Constructing 3-d city models by merging aerial and ground views. *IEEE CGA*, 2003.

[6] M. Golparvar-Fard, F. Pena-Mora, and S. Savarese. Monitoring changes of 3d building elements from unordered photo collections. In *ICCV Workshops*, 2011.

[7] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical crfs for object class image segmentation. In *International Conference on Computer Vision*, 2009.

[8] L. Liu and I. Stamos. Automatic 3d to 2d registration for the photorealistic rendering of urban scenes. In *CVPR*, 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. Pollard and J. L. Mundy. Change detection in a 3-d world. In *CVPR*, 2007.

[11] 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.

[12] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14:294–307, 2005.

[13] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand. Skyline2gps: Localization in urban canyons using omni-skylines. In *IROS*, 2010.

[14] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, 2011.

[15] A. Taneja, L. Ballan, and M. Pollefeys. Image based detection of geometric changes in urban environments. In *ICCV*, 2011.

[16] A. Taneja, L. Ballan, and M. Pollefeys. Registration of spherical panoramic images with cadastral 3d models. In *3DIMPVT*, 2012.

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

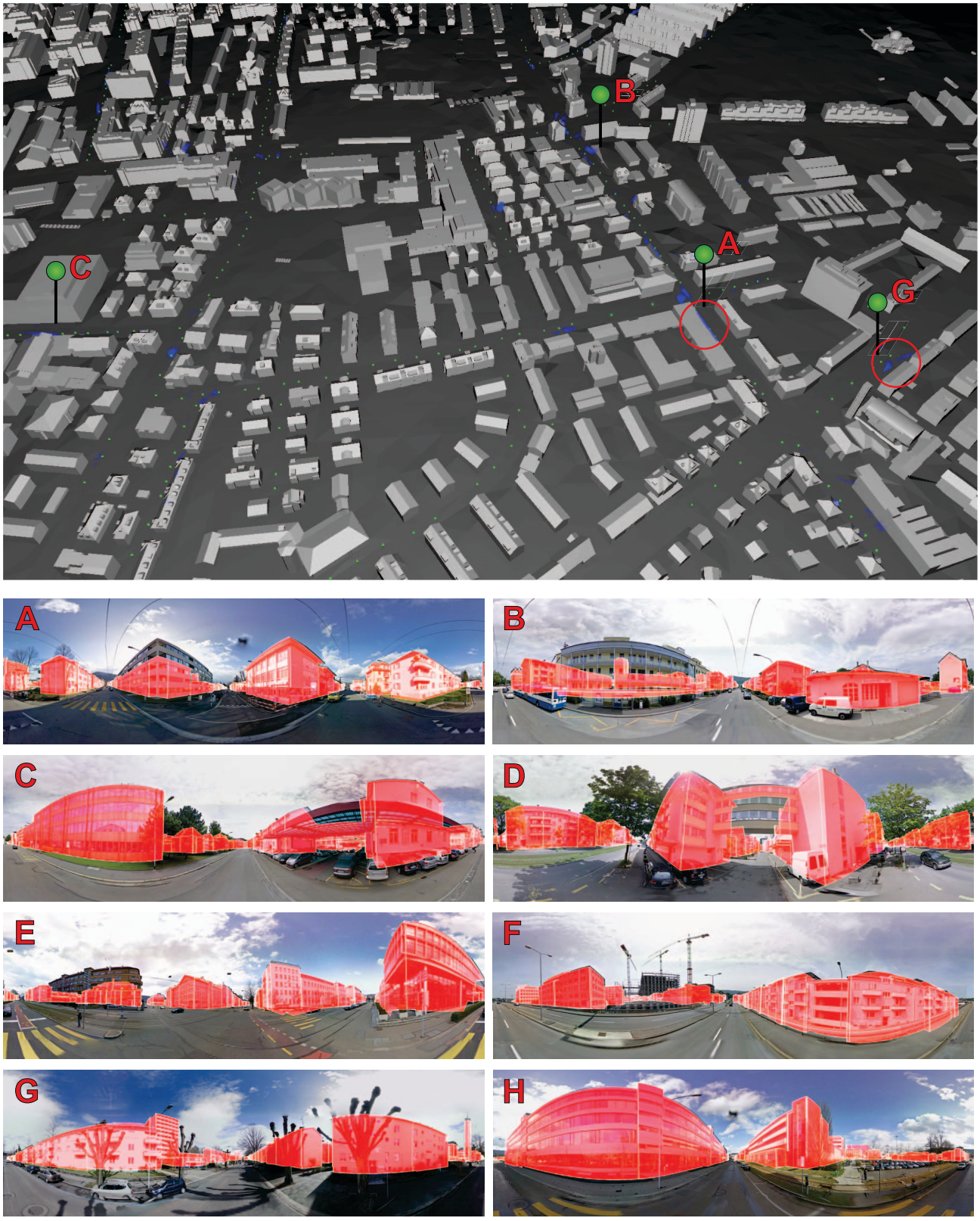


Figure 5. (Top) Cadastral 3D model overlaid with the voxel grid. Voxels detected as a change are marked in blue. The input images are shown as green dots, while the green markers indicate some of the changed locations recognized using our approach. (Bottom) Images corresponding to the green markers in the map overlaid with the cadastral model. For locations D, E, F and H please refer to Figure 1.