Image Based Detection of Geometric Changes in Urban Environments

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 an efficient technique to detect changes in the geometry of an urban environment using some images observing its current state. The proposed method can be used to significantly optimize the process of updating the 3D model of a city changing over time, by restricting this process to only those areas where changes are detected. With this application in mind, we designed our algorithm to specifically detect only structural changes in the environment, ignoring any changes in its appearance, and ignoring also all the changes which are not relevant for update purposes, such as cars, people etc. As a by-product, the algorithm also provides a coarse geometry of the detected changes. The performance of the proposed method was tested on four different kinds of urban environments and compared with two alternative techniques.

1. Introduction

Motivated by the success of online services such as GoogleEarth and StreetView, as well as by the expectation of future navigation applications, lot of attention has gone specifically to developing efficient techniques for reconstructing static 3D models of urban environments from imagery and/or range measurements captured from ground-based vehicles [18, 3], as well as aerial platforms [6, 23]. Recent developments in this area have proven that one can reach impressive levels of detail in these environments capturing even thin structures like trees and rails [13].

However, while the main structures in an urban scene remain unchanged for very long periods of time (decades or even centuries), on the scale of a city new structures are continuously being erected and old taken down [22]. As a consequence, any previously reconstructed 3D model becomes obsolete rapidly. Considering the vast number of applications that rely on such data, there is a need to explore efficient solutions to keep these models consistent with the current state of the environment.

The naïve solution of updating these models by repeating the process of data collection and reconstruction on the whole environment on a regular basis, is not only time consuming but also very expensive. In fact, while reconstruction algorithms are getting faster day by day exploiting parallelism on GPUs [11] or dedicated clusters of computers [1], the collection of the data, necessary for these algorithms, still needs dedicated setups (multiple cameras, sensors, scanners etc.) mounted on cars, driving around the city or on aerial vehicles flying over the area of interest,

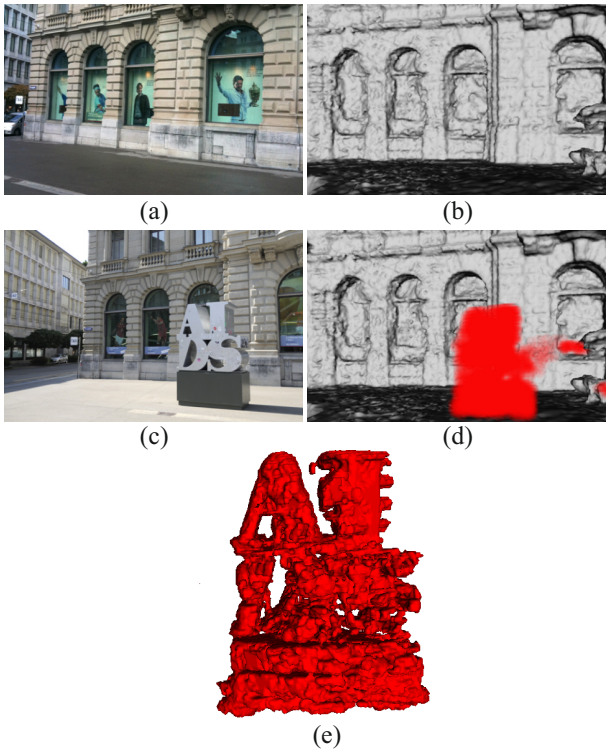


Figure 1. Example output of the proposed algorithm. (a) One of the images used to recover the initial geometry of the scene, shown in (b). (c) One of the images of the same location captured after some time: a new structure was placed. (d) Computed volumetric inconsistency map between the new images (c) and the initial geometry (b): red indicates inconsistencies. (e) Coarse geometry of the detected changes computed using our approach.

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00 ©2011 IEEE

with the sole intention of capturing data for reconstruction. The time and effort involved in this exhaustive data collection makes this approach impractical for a frequent update. A way to incrementally update these models which does not completely discard the existing information, needs to be explored.

This motivates our effort to leverage the existing 3D model and some images representing the current state of the scene, to efficiently determine which areas have undergone significant changes and which parts of the model are still accurate. In principle, these new images can be recorded from low resolution consumer cameras mounted on third party vehicles driving around the city for different purposes: for instance, postal vans or taxis due to their excellent coverage across the city. The captured data can then be processed offline to discover if any changes have occurred in the explored areas. An update process can then be planned by adding the locations of the observed changes, to a list of sites, to be visited during a future run with the scanning vehicle, to capture data with high quality sensors.

2. Related Work

For a broad applicability of the proposed idea, the hardware to be mounted on the vehicles, needs to be kept as minimal as possible. Therefore, we need to consider that, for each explored location, only sparse and low resolution imagery might be available. Detecting structural changes that may have occurred in an environment from only these images is not trivial.

Intuitively, a first approach would be to apply multi-view stereo (MVS) on these images to recover a local updated geometry of the scene. Geometric changes can then be detected by performing a 3D-to-3D comparison between this new model and the original one. The accuracy of such a comparison however, relies on the quality of the obtainable MVS reconstruction, which may be low in scenarios with sparse wide baseline imagery.

On the other hand, change detection literature offers a lot of solutions based on 2D-to-2D comparisons between images representing the old state of a scene and images representing its current state [19]. These approaches however are sensitive to changes in illumination and weather conditions across the old and the new images. To partially overcome these issues [17] proposed to learn, from the old images, a probabilistic appearance model of the 3D scene, to be used for comparison with the new images. [4] instead, proposed to detect changes based on the appearance and disappearance of 3D lines detected in the images.

These methods however, focus on generic appearance changes across the old and new images, which may or may not correspond to changes in the geometry of the scene. Since our aim is to keep the geometry of an urban environment up to date, we need to focus only on geometric changes that may have occurred, ignoring any changes in the appearance, such as different paints on a wall, new posters or new advertisements on boards etc.

In this paper, we propose a technique to detect changes in the geometry of an environment using a few low resolution images, observing its current state. The proposed algorithm exploits the existing geometry to detect inconsistencies across these images. In particular, it does not consider changes in the appearance or changes on objects that are not relevant for the purpose of keeping the model up to date, such as changes in vegetation, cars and pedestrians.

3. Algorithm

We assume that the last data acquisition and reconstruction of the urban environment took place at a certain time t_0, and that Γ indicates the 3D model resulting from such a procedure. This model will be used as a reference to detect all the future changes in the environment. At a subsequent time $t_1 > t_0$, a set of images is captured representing the current state of the urban scene.

These images are first registered with respect to the original geometry Γ (Section 3.1). A probabilistic framework is then used to verify their consistency with respect to Γ (Section 3.2). In order to ignore changes occurring on non-relevant parts of the scene, semantic knowledge of the environment is incorporated into the proposed framework (Section 3.3). As a final and optional step, a coarse update of the geometry can also be recovered (Section 3.4).

3.1. Image Registration

A lot of research has been devoted to this particular problem, especially for urban scenes. Both visual [30, 20, 21] and geometric information [2, 14] have already been exploited to approximately localize images in an environment. Once these images are roughly mapped to a location in a city, classical registration is used to refine the result.

In our scenario, irrespective of whether the original geometry Γ is built using imagery or range scan data, as long as there is some texture information available, feature correspondences, like SIFT [16], VIPS [26] or orthophoto-correspondences [2], can be used to relate the captured images with Γ. Since, each correspondence is related to a 3D point in Γ, the images can be registered using Direct Linear Transform (DLT) followed by a refinement step based on the reprojection error [9]. In cases where a significant change covered the majority of the field of view of an image, the number of found correspondences was insufficient to apply DLT. To recover from this, the images were first registered relative to each other, on a common coordinate system, using Structure from Motion [27]. If one or more of these images saw a sufficient part of the scene that had not changed, so they could also be registered with Γ using DLT, then the transformation between the two coordinates

system was computed and transferred to the remaining images as well. Clearly, if GPS or other additional information are available, the registration process becomes simpler.

3.2. Change Detection

To ensure the scalability of the proposed approach to large environments, the 3D model Γ is first subdivided into uniformly sized 3D regions and each of those is considered independently for detecting changes. Let I denote the set of captured images observing a specific 3D region. It is reasonable to assume that these images are taken around the same time so that changes in illumination of the environment can be neglected across them. In a practical situation, the timestamp can be used to discard the images that do not comply with the above assumption.

The considered 3D region is discretized into voxels. Let $\mathcal{V}$ represent these voxels and $\langle \mathcal{V}, \mathcal{E} \rangle$ the graph connecting them, such that each edge $e_{ij} \in \mathcal{E}$ connects only adjacent voxels (26-neighborhood). We aim to compute a binary labeling $\mathcal{L} = \{l_i\}_i$ for each element in $\mathcal{V}$ according to the occurred changes. Specifically, $l_i = 1$ indicates the presence of a change in voxel i, or in other words, it indicates that the current state of the environment in voxel i is inconsistent with the original geometry Γ. On the contrary, $l_i = 0$ indicates consistency. To label these voxels, we maximize the posterior probability of $\mathcal{L}$ given the observations I, i.e., we maximize $p(\mathcal{L}|I)$. Assuming dependence only across neighbouring voxels, this is equivalent to minimizing the Gibbs energy (please refer to [25] for details)

$$\sum_i \psi_i(l_i) + \sum_{e_{ij} \in E} \psi_{ij}(l_i, l_j), \tag{1}$$

where the unary term $\psi_i(l_i)$ represents the log-likelihood $-log(p(I|l_i))$ and the binary term $\psi_{ij}(l_i, l_j)$ accounts for the spatial dependencies across neighboring voxels, i.e., it is equal to $-log(p(l_i, l_j))$. We define the binary term $\psi_{ij}(l_i, l_j)$ such that it penalizes the assignment of different labels to adjacent voxels represented on the same image with similar colors. More precisely,

$$\psi_{ij}(l_i, l_j) = [l_i \neq l_j] \cdot \gamma / \left(\sum\nolimits_{I_t} ||c_t^i - c_t^j||^2 + 1 \right), \tag{2}$$

where $||c_t^i - c_t^j||$ is the L_2-norm of the difference between the RGB colors c_t^i and c_t^j of the two voxels i and j on the same image I_t. $\gamma > 0$ is a regularization factor.

Concerning the unary term $\psi_i(l_i)$, a first approach would be to store the appearance of each voxel from previous acquisitions, and to compare it with the images in I. Something similar was explored in [17]. However, this kind of approach is sensitive to changes between the old and the current appearance of the scene.

On the contrary, we use the geometry Γ to transfer the current appearance across the images in I. Previous works like [10, 24, 29] have shown that such an approach can be used to recover dynamic elements in a scene such as people walking in an environment under surveillance. Following a similar intuition, for each pair of images in I, say (I_t, I_s), we render a new image by projecting the colors of the source image I_s into the target image I_t using the geometry Γ and the registration parameters for both I_t and I_s. More precisely, each ray corresponding to a pixel in I_t, is cast to Γ and reflected back into the image plane of I_s to retrieve a pixel color. Subsequently, this new image is compared with the original image I_t to obtain a sort of 2D inconsistency map between I_t and I_s, that we will denote with the symbol $M_{t \leftarrow s}$. Figure 2 depicts this procedure. To account for possible errors in the registration or in Γ, this comparison was performed on a 7x7 window as in [24].

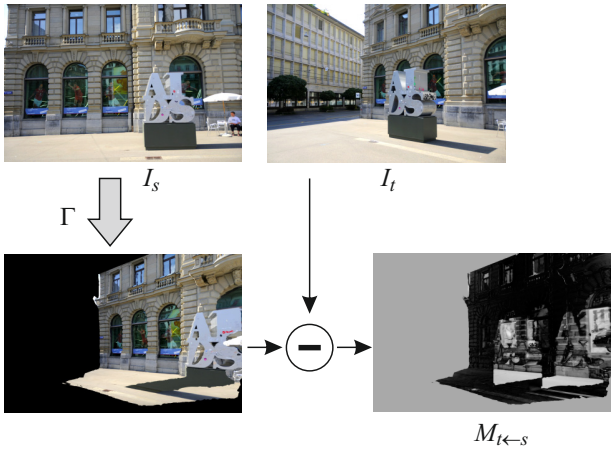


Figure 2. Image formation process of a 2D inconsistency map $M_{t \leftarrow s}$ computed for the scene shown in Figure 1. Since the new structure in front of the building, was not modeled by the original geometry Γ, the resulting image $M_{t \leftarrow s}$ reveals some inconsistencies in the corresponding pixels.

Ideally, if the geometry Γ still represents the current state of the scene observed by the images I, these images should reproject onto each other correctly, i.e., the 2D inconsistency maps $M_{t \leftarrow s}$ should be all zero. On the contrary, if some $M_{t \leftarrow s}$ differ from zero then there is an evidence of a possible change.

Let $M = \{M_{t \leftarrow s} | \forall t, s\}$ be the set of 2D inconsistency maps obtained from all the possible image pairs in I. From a probabilistic point of view, M is a random vector linked deterministically to the images in I, i.e., its conditional probability distributions $p(M|I)$ and $p(I|M)$ differ from zero only when all the inconsistency maps $M_{t \leftarrow s}$ in M are obtained with the previously described procedure.

By marginalizing over M, the probability $p(I|l_i)$, related to the unary term $\psi_i(l_i)$, becomes

$$p(I|l_i) = \sum\nolimits_M p(I|M, l_i)\, p(M|l_i) \propto p(M|l_i), \tag{3}$$

where, the proportionality holds since all the terms inside the sum are zero except for only a specific M. Minimizing Equation 1 using $p(M|l_i)$ in place of $p(I|l_i)$ is therefore equivalent.

In general, when a change in the geometry occurs, two evidences of this change are visible in each $M_{t\leftarrow s}$ map: one corresponding to the pixels of the change observed by I_t, and the other being the pixels of the change observed by I_s projected into I_t (see Figure 2). Let $\pi^i_{t\leftarrow s}$ denote the set of pixels providing these evidences for a specific voxel i. For tractability, we assume independence in the image formation process for each pixel q in each inconsistency map $M_{t\leftarrow s}$, therefore,

$$p(M|l_i) = \prod_{t,s} \prod_{q\in\pi^i_{t\leftarrow s}} p\left(M_{t\leftarrow s}(q)\,|l_i\right). \qquad (4)$$

We then define $p\left(M_{t\leftarrow s}(q)\,|l_i\right)$ to be

$$p\left(M_{t\leftarrow s}(q)\,|l_i\right) = \begin{cases} e^{-\frac{M_{t\leftarrow s}(q)^2}{2\sigma^2}} & l_i = 0 \\ U & l_i = 1 \end{cases}. \qquad (5)$$

Equation 5 states that, if a voxel i has not changed since the last acquisition, all the corresponding pixels in $M_{t\leftarrow s}$ should follow a normal distribution centered around zero. In other words, in those pixels, the two images I_t and the projection of I_s into I_t, should agree. On the contrary, if voxel i has changed, nothing can be said about the values of those pixels, and so we approximate their probability with the least informative one, i.e., the uniform distribution U.

In order to speed up the computation of $\psi_i(l_i)$, we reduce the number of considered image pairs in M by selecting only those with sufficient overlap in their field of view and discarding also the symmetric ones.

Since the defined unary and binary terms satisfy the metric requirements, graph cuts [12] was used to minimize Eq. 1. The obtained labeling $\mathcal{L}$ corresponds to a volumetric inconsistency map between the original model Γ and the current state of the environment. An example of this map can be seen in Figure 1(d), where only the voxels labeled as 1 are displayed in red. Voxels are rendered using transparency to emphasize the volumetric nature of the result.

3.3. Change Understanding

By minimizing the energy in Equation 1 we aim to detect all the geometric changes that may have occurred in the environment since the last acquisition. However, for the problem being addressed in this paper, some of these changes might not be relevant and should be discarded by the algorithm: for instance, people walking on a street, cars parked in front of buildings, natural vegetation etc. We avoid detecting such changes by incorporating some semantic knowledge about these objects into our framework.

Let us consider r mutually exclusive classes of objects $\{0, 1, \ldots, r-1\}$. Let class 0 denote relevant objects while all the other classes denote only irrelevant objects. Let $p(\omega^q_t = c)$ represent the probability of a pixel q in image I_t to belong to an object of a specific class c. We account for these probabilities in Equation 5 by increasing the uncertainties of $p\left(M_{t\leftarrow s}(q)\,|l_i\right)$ when either the information coming from the source or the target image belongs to a non relevant object. Specifically, we use the same technique described in the previous section to transfer information from a source image I_s to the image plane of a target image I_t. This time, instead of transferring colors, we transfer the probability $p(\omega^q_s = 0)$ related to the source image. Let $\omega^q_{t\leftarrow s}$ denote the random variable related to such a projection, i.e., computed by mapping the random variable ω^q_s into I_t.

What we stated before can be formalized by defining the conditional probability $p\left(M_{t\leftarrow s}(q)\,|l_i, \omega^q_t, \omega^q_{t\leftarrow s}\right)$ equal to

$$\begin{cases} p\left(M_{t\leftarrow s}(q)\,|l_i\right) & \omega^q_t = 0 \wedge \omega^q_{t\leftarrow s} = 0 \\ U & \text{otherwise} \end{cases}, \qquad (6)$$

where $p\left(M_{t\leftarrow s}(q)\,|l_i\right)$ is defined as in Equation 5 and U denotes the uniform distribution. By marginalizing over ω^q_t and $\omega^q_{t\leftarrow s}$, the new probability distribution of $M_{t\leftarrow s}(q)$ given l_i, call it $\widetilde{p}\left(M_{t\leftarrow s}(q)\,|l_i\right)$, becomes

$$\sum p\left(M_{t\leftarrow s}(q)\,|l_i, \omega^q_t, \omega^q_{t\leftarrow s}\right) p\left(\omega^q_t, \omega^q_{t\leftarrow s}\right). \qquad (7)$$

Before substituting Equation 6 into Equation 7, we simplify the notation introducing the symbol Ω^q_t to indicate the probability $p(\omega^q_t = 0)$, and the symbol $\Omega^q_{t\leftarrow s}$ to indicate the probability $p(\omega^q_{t\leftarrow s} = 0)$. Now, assuming independence between ω^q_t and $\omega^q_{t\leftarrow s}$, Equation 7 is rewritten as

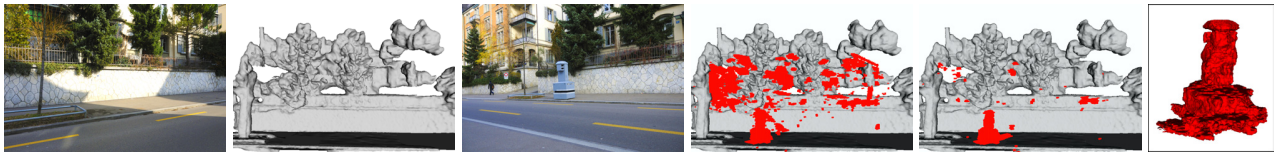
$$(1 - \Omega^q_t\Omega^q_{t\leftarrow s}) \cdot U + \Omega^q_t\Omega^q_{t\leftarrow s} \cdot p\left(M_{t\leftarrow s}(q)\,|l_i\right). \qquad (8)$$

In this way, if either Ω^q_t or $\Omega^q_{t\leftarrow s}$ have low values, the probability distribution of $M_{t\leftarrow s}(q)$ given any possible voxel labeling tends to be uniform, consequently pixel q does not carry any discriminative information for the voxels.

In our current implementation, we focused on the most commonly encountered cases of irrelevant changes in urban scenes namely changes in vegetation, cars and pedestrians. In order to compute the probabilities $p(\omega^q_t = c)$ for vegetation we used the same patch based k-nearest-neighbors approach on both color and edge features as described in [8]. For cars and pedestrians instead, we used the same approach as presented in [5].

3.4. Model Update

Ideally, once a significant change is detected in the environment, a new data acquisition with high quality sensors can be planned focusing only on the changed areas.



(a) (b) (c) (d) (e) (f)

Figure 3. (a) One of the images used to recover the initial geometry, (b) initial geometry, (c) one of the new images, (d) and (e) volumetric inconsistency maps obtained without and with accounting for the semantic information, (f) update obtained as described in Section 3.4.

(a) (b) (c) (d) (e)

Figure 4. (a) One of the images used to recover the initial geometry, (b) initial geometry, (c) one of the new images, (d) and (e) volumetric inconsistency maps obtained without and with accounting for the semantic information projected onto the image plane of image (c).

Advanced reconstruction algorithms can then be applied on this new data to recover an accurate updated 3D model.

In the meanwhile, a coarse geometry of the changes can be computed as a temporary update to the model, using the available images. To perform this, the detected 3D inconsistencies are first grouped into clusters using connected components, and only clusters with significant sizes are considered for an update. The volumetric inconsistency maps are then recomputed for each of these clusters independently, at a higher resolution.

In order to incorporate the detected changes into the model, the existing geometry Γ is first converted into its volumetric representation. Specifically, a voxel is labeled 1 if it is inside Γ, and 0 otherwise. We then apply the XOR operator between this voxelization and the computed volumetric inconsistency map, considering four cases: $(0,1)$ means that an element has been added in the scene, $(1,1)$ means that an element has been removed, $(0,0)$ and $(1,0)$ mean that the state of the geometry inside this voxel has not changed. In the end, the update is obtained by applying the marching cubes algorithm [15] to the resulting labeling. An example of such an update is shown in Figure 1(e).

4. Experiments and Discussion

The proposed algorithm was evaluated on four different urban environments. In all the experiments, the initial geometry Γ was recovered using imagery, specifically using [28]. After some time had elapsed, some new images of the same locations were captured using a 0.8Mpixel consumer camera from the street side. These images were then registered with respect to Γ as described in Section 3.1. The achieved reprojection error for the registration was on an average between 1.5 and 2 pixels. The scale of the considered locations varied between $150m^2$ and $4500m^2$. The chosen voxel size was $25cm$ in each dimension and the size of each considered 3D region was limited to $1000m^2$. On an average, 8 newly captured images were used to compute the volumetric inconsistency maps for each of these regions.

4.1. Qualitative Evaluation

In the first dataset (Figure 1), we analyzed the case where a new structure was placed in front of a building inside a commercial area. As can be seen from the images in Figure 1(a) and (c), the posters displayed on the windows had changed between the first and the second round of acquisition. This is frequent in urban environments, especially in commercial areas, and would be a serious issue for those methods which use the appearance from the last acquisition to detect changes. Since our algorithm uses only the new set of images for comparison, it correctly detects the new structure, ignoring the changes on the posters, which we are not interested in detecting. The resulting volumetric inconsistency map is shown in Figure 1(d). Some voxels besides the new structure were also labeled as changed, however, these get discarded during the update process, as described in Section 3.4. Figure 1(e) shows the obtained updated model.

In the second dataset (Figure 3), a speed monitoring device was placed on a street whose geometry was acquired two weeks before. Since the images were taken on two separate days and at different times of the day, the lighting conditions were completely different. Despite this and due to the robustness offered by the SIFT descriptor, a sufficient number of correspondences could be established between the old and the new images, allowing the registration of all the new images with respect to Γ.

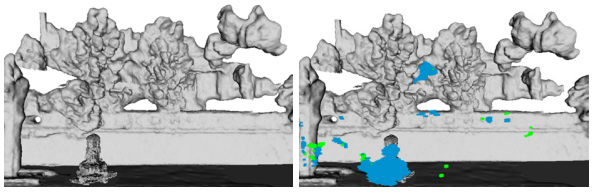
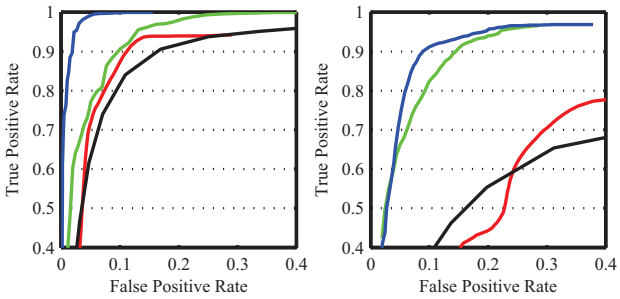


Figure 5. Result of the reverse experiment performed on the same dataset of Figure 3. (left) Initial geometry, (right) Volumetric inconsistency map after the XOR operator: (blue) voxels to be removed, (green) voxels to be added.



(a) (b)

Figure 6. Change detection ROC for the second (a) and the third dataset (b): Blue and Green curve: results obtained using our method with and without accounting for semantic information respectively. Red curve: result obtained using a 3D-to-3D comparison technique. Black curve: result obtained using a 2D-to-2D comparison technique. (Best viewed in color)

Figure 3(d) and 3(e) show a comparison between the volumetric inconsistency maps obtained without and with taking semantic information into account, as described in Section 3.2 and 3.3 respectively. In the former case, since the geometry of the trees and the bushes changes with time (due to movement of leaves or seasonal changes), most of the corresponding voxels were labeled as inconsistent. Using semantic information instead, these changes were discarded revealing clearly the structure of the device. Figure 3(f) shows the result obtained after the model update.

A reverse experiment was performed on the same dataset to evaluate the algorithm behavior on an object removal case. The previously obtained updated geometry was used in place of the original model, and the old images without the device, were used instead of the newly captured images I. The obtained result is shown in Figure 5, where each inconsistent voxel is colored in either blue or green to indicate that something has been removed or added, respectively. Inconsistencies were found on the majority of the removed device except for its upper part, which was excluded because some of the corresponding pixels in the $M_{t \leftarrow s}$'s images overlapped with the bushes, which had a high probability of being irrelevant objects.

In the third dataset, we considered a street in a residential area (Figure 4). At the time of the second acquisition, multiple structures had been added in front of the building, covering a considerable part of the field of view of the captured images. The semantic information helped to discard irrelevant changes like the car parked behind the new structures and the bushes. Although the algorithm correctly detected the new structures, very little could be inferred about the parts of the building occluded by them. In fact, as can be seen in Figure 4(e), some false positives were detected around the two windows behind the new structures. This was due to the fact that these two regions were represented in only one of the new images and therefore, no 2D inconsistency map had information about their true state.

In the fourth dataset (Figure 7), 25 new images were captured around a big intersection whose geometry was acquired two months before. The environment extends for about $4500m^2$ and it was split into four 3D regions. For each of these regions, 6 new images were selected and used for detecting changes. Inconsistencies were detected around a booth and a stall that were added on the footpath. Some small and sparse inconsistencies were also detected in the center of the intersection due to the tram wires and some poles that were not captured in the initial acquisition, due to their small size. Note that, in presence of a change some neighboring voxels are incorrectly labeled as inconsistent if none of the images give information about them. This is visible in the top view for the booth whose detected inconsistencies exceed the area of the actual change.

4.2. Quantitative Evaluation

For a quantitative evaluation of the proposed algorithm, we generated a ground-truth by manually segmenting the occurred changes on each image in I. These were then compared against the masks obtained by projecting the computed volumetric inconsistency maps onto the images I. This procedure was repeated 50 times for different values of σ in Equation 5. The fraction of correctly labeled pixels against the fraction of falsely labeled ones were computed for all the images in I, and displayed in a ROC curve.

Figure 6 shows the ROC curves obtained for the second and the third dataset. It is evident that, taking the semantic information into account (blue curve) decreases the falsely detected changes significantly. In the third dataset, the performance of the algorithm decreased due to lack of information already observed in the previous section.

4.3. Comparison with Alternative Techniques

Since there is no previous work focusing specifically on geometric changes, we propose two alternative techniques, and evaluate their performance against our approach.

The most appropriate baseline for comparison is the 3D-

to-3D approach mentioned in Section 2. We applied multi-view stereo, precisely PMVS [7]+Poisson reconstruction, to the newly captured images. Changes were then detected by thresholding the differences between the depthmaps obtained by rendering this new reconstruction and the original model Γ, from the point of view of the new images.

When multi-view stereo was able to recover an accurate 3D geometry, the results obtained by the 3D-to-3D method were comparable with those obtained by running our approach without accounting for semantic information, see the red curve in Figure 6(a). On the contrary, in the case when the images were captured sparsely, with wide baselines or in the presence of textureless regions, the resulting poor reconstruction of the scene reduced drastically the discriminative property of this approach, see Figure 6(b).

We also implemented a 2D-to-2D change detection approach performing a comparison between the new and the old images. The same reprojection technique presented in Section 3.2, was used to compensate for the difference in viewpoints across the two sets of images. A global color calibration and a local luminance normalization were performed to compensate for the different lighting conditions. Despite this last expedient, the differences in appearance across the two image sets, not corresponding to geometric changes, biased the results, increasing the number of false positives (see the black curve in Figure 6).

On a single core working at 2.8GHz, the running time per region was 35 minutes for the 3D-to-3D method (MVS+comparison), 5 seconds for the 2D-to-2D method, and 1 minute for our approach.

5. Conclusions and Future Work

In this paper, we proposed an efficient technique to detect changes in the geometry of an urban environment that may have occurred since its last 3D acquisition, using some images representing its current state. The proposed algorithm can be used to significantly optimize a model update process by restricting the data acquisition and update to only those areas where changes are detected. Unlike the high resolution and dense imagery needed to remodel the entire environment from scratch, we need as few as 8 low resolution images to detect the possible changes for each region of size $1000m^2$.

In the experiments section, we showed that the proposed method was able to correctly classify changes on four different urban environments. Since only the current images of the scene were used, the results were not influenced by changes in illumination across the old and the new images, such as in Figure 3, or changes in the model texture, such as the changing posters in Figure 1. Moreover, the use of semantic information allowed us to ignore the changes corresponding to irrelevant objects, as shown in Figure 4. The algorithm proved to be robust to deal with relatively large and cluttered environments like the one in Figure 7.

We also proposed two alternative techniques and analyzed their performance. Even without accounting for semantic information, our approach outperformed these two techniques. Moreover, in the case when the obtained results were comparable (as in Figure 6(a) for the 3D-to-3D approach), this only came at the cost of more computational time (35 min. vs. 1 min.). This is because, while the proposed method has to consider consistency only with a single depth hypothesis, i.e., the original geometry, multi-view stereo, required by the 3D-to-3D approach, needs to consider all possible depths.

Limitations: The volume detected using our approach always bounds the actual change. In fact, false positives may be detected in areas surrounding a change in case these areas are not seen by at least two images. Moreover, like other change detection techniques, our approach still suffers in case of strong reflective surfaces, which may generate false positives in the $M_{t \leftarrow s}$ maps. The computational time of 1 minute per region is reasonable. However, as a future extension, the intensive step of solving for Equation 1, can be triggered only when a significant evidence of change is observed in the $M_{t \leftarrow s}$ images (rare event in the intended application scenario), reducing the total computational time drastically.

Acknowledgements: 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), SNF and Google.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009.

[2] G. Baatz, K. Koser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Handling urban location recognition as a 2d homothetic problem. In *ECCV*, 2010.

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

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

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

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

[7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, pages 1362–1376, 2010.

[8] D. Gallup, J. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *CVPR*, 2010.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

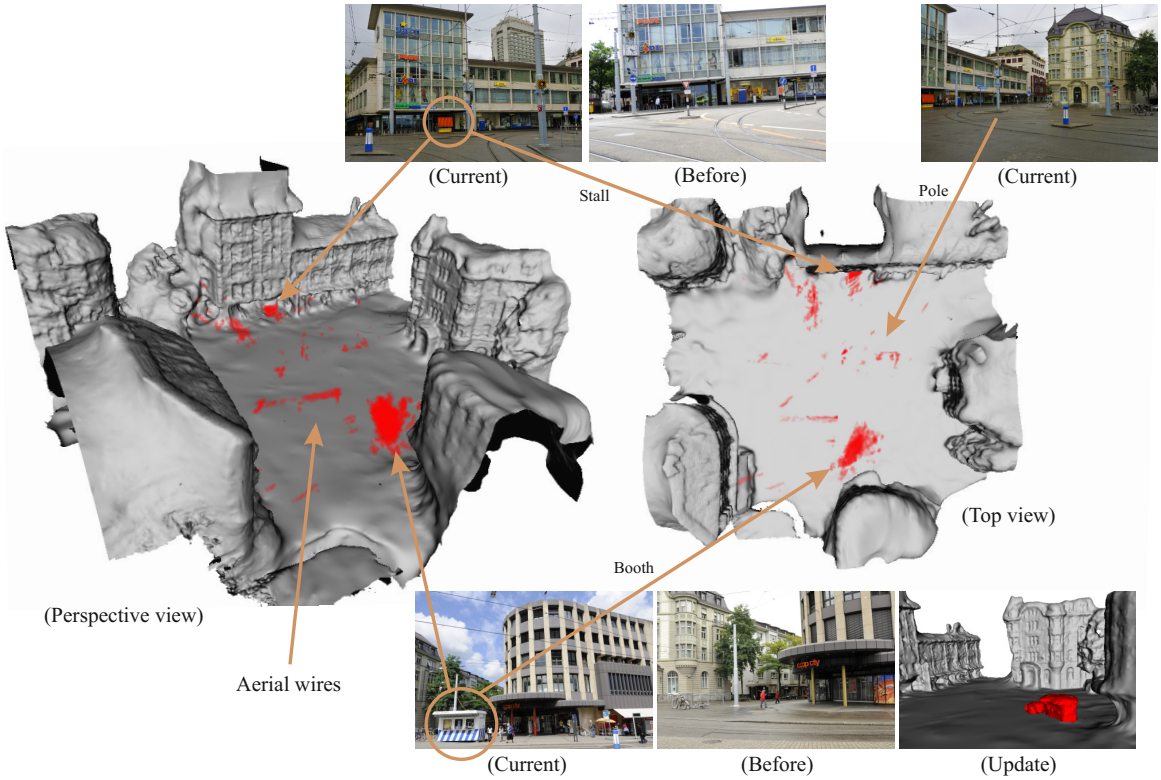


Figure 7. Volumetric inconsistency map computed on a big intersection using 25 new images. Two big clusters were detected corresponding to a stall (above) and a booth (below). The model update of the booth was, in this case, computed using standard multi-view stereo applied only to the area where the change was detected.

[10] Y. Ivanov, A. Bobick, and J. Liu. Fast lighting independent background subtraction. *IJCV*, 2000.

[11] J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram et al. Building rome on a cloudless day. In *ECCV*, 2010.

[12] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[13] P. Labatut, J.-P. Pons, and R. Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*, 28, 2009.

[14] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010.

[15] W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*, 1987.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[17] T. Pollard and J. L. Mundy. Change detection in a 3-d world. In *CVPR*, 2007.

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

[19] 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.

[20] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, pages 819–828, 2004.

[21] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, 2007.

[22] G. Schindler and F. Dellaert. Probabilistic temporal inference on reconstructed 3d scenes. In *CVPR*, 2010.

[23] Y. Takase, N. Sho, A. Sone, and K. Shimiya. Automatic generation of 3d city models and related applications. In *ISPRS*, pages 113–120, 2003.

[24] A. Taneja, L. Ballan, and M. Pollefeys. Modeling dynamic scenes recorded with freely moving cameras. In *ACCV*, pages 613–626, 2010.

[25] M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. 2008.

[26] C. Wu, B. Clipp, L. Xiaowei, J.-M. Frahm, and M. Pollefeys. 3d model matching with viewpoint-invariant patches (vip). In *CVPR*, 2008.

[27] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, 2010.

[28] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust TV-L^1 range image integration. In *ICCV*, 2007.

[29] G. Zhang, J. Jia, W. Xiong, T.-T. Wong, P.-A. Heng, and H. Bao. Moving object extraction with a hand-held camera. *ICCV*, 0:1–8, 2007.

[30] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, pages 33–40, 2006.