# Structural change detection by direct 3D model comparison

# Structural Change Detection by Direct 3D Model Comparison

Marco Fanfani and Carlo Colombo
*Department of Information Engineering (DINFO), University of Florence, Via S. Marta 3, 50139, Florence, Italy*
{*marco.fanfani, carlo.colombo*}*@unifi.it*

Keywords: Change Detection, 3D Reconstruction, Structure from Motion.

Abstract: Tracking the structural evolution of a site has important fields of application, ranging from documenting the excavation progress during an archaeological campaign, to hydro-geological monitoring. In this paper, we propose a simple yet effective method that exploits vision-based reconstructed 3D models of a time-changing environment to automatically detect any geometric changes in it. Changes are localized by direct comparison of time-separated 3D point clouds according to a majority voting scheme based on three criteria that compare density, shape and distribution of 3D points. As a by-product, a 4D (space + time) map of the scene can also be generated and visualized. Experimental results obtained with two distinct scenarios (object removal and object displacement) provide both a qualitative and quantitative insight into method accuracy.

## 1 INTRODUCTION

Monitoring the evolution over time of a three-dimensional environment can play a key role in several application fields. For example, during an archaeological campaign it is often required to record the excavation progress on a regular basis. Similarly, tracking of changes can be useful to contrast vandalism in a cultural heritage site, to prevent natural damages and reduce hydro-geological risks, and for building construction planning and management (Mani et al., 2009).

A simple strategy to change tracking requires that photos or videos of the scene are acquired and inspected regularly. However, a manual checking of all the produced data would be time consuming and prone to human errors. To solve this task in a fully automatic way, 2D and 3D change detection methods based on computer vision can be employed.

Vision-based change detection is a broad topic that includes very different methods. They can be distinguished by the used input data—pairs of images, videos, image collections or 3D data—and by their application scenarios, that can require different levels of accuracy. In any case, all methods have to consider some sort of registration to align the input data before detecting the changes. Through the years, several methods have been proposed (Radke et al., 2005). The first approaches were based on bi-dimensional data, and used to work with pairs of images acquired at different times. In order to detect actual scene changes, geometric (Brown, 1992) and radiometric (Dai and Khorram, 1998; Toth et al., 2000) registration of images were implemented to avoid detection of unrelevant changes due to differences in point of view or lighting conditions. Video-surveillance applications (Collins et al., 2000) were also considered: In this case a video of a scene acquired from a single point of view is available, and change detection is obtained by modelling the background (Toyama et al., 1999; Cavallaro and Ebrahimi, 2001)—typically using mixture-of-Gaussian models. Successively, solutions exploiting three-dimensional information were introduced, in order to obtain better geometric registration and mitigate problems related to illumination changes. In (Pollard and Mundy, 2007), the authors try to learn a world model as a 3D voxel map by updating it continuously when new images are available—a sort of background modelling in 3D. Then change detection is performed by checking if the new image is congruent with the learned model. However, camera poses are supposed to be known and lighting conditions are kept almost constant. Other methods exploit instead a 3D model pre-computed from image collections (Taneja et al., 2011; Palazzolo and Stachniss, 2017) or depth estimates (Sakurada et al., 2013) of the scene so as to detect structural changes in new images. After registering the new image sequence on the old 3D model—using feature matching and camera resectioning (Taneja et al., 2011; Sakurada et al., 2013) or exploiting also GPS and inertial measurements (Palazzolo and

Stachniss, 2017)—change detection is obtained by re-projecting a novel image onto the previous views by exploiting the 3D model, so as to highlight possible 2D misalignments, indicating a change in 3D. However, these solutions find not only structural changes but also changes due to moving objects (i.e. cars, pedestrians, etc.) that can appear in the new sequence: in order to discard such nuisances, object recognition methods have been trained and used to select changed areas to be discarded (Taneja et al., 2011). In (Taneja et al., 2013), a similar solution is adopted, using instead a cadastral 3D model and panoramic images. Differently, in (Qin and Gruen, 2014) the reference 3D point cloud is obtained with an accurate yet expensive laser scanning technology; changes are then detected in images captured at later times by re-projection. Note that, in order to register the laser-based point cloud with the images, control points have to be selected manually. More recently, even deep network have been used to tackle change detection (Alcantarilla et al., 2016) using as input registered images from the old and new sequences.

Differently from the state-of-the-art, in this paper we propose a simple yet effective solution based on the analysis of 3D reconstructions computed from image collections acquired at different times. In this way, our method focuses on detecting structural changes and avoids problem related to difference in illumination, since it exploits only geometric information from the scene. Moreover, 3D reconstruction methods such as Structure from Motion (SfM) (Szeliski, 2010), Simultaneous Localization and Mapping (SLAM) (Fanfani et al., 2013) or Visual Odometry (Fanfani et al., 2016) build 3D models of fixed structures only, thus automatically discarding any moving elements in the scene. By detecting differences in the 3D models, the system is able to produce an output 3D map that outlines the changed areas. Our change detection algorithm is fully automatic and is composed by two main steps: (i) initially, a rigid registration at six degrees of freedom has to be estimated in order to align the temporally ordered 3D maps; (ii) then, the actual change detection is performed by comparing the local 3D structures of corresponding areas. The detected changes can also be transported onto the input photos/videos to highlight the image areas with altered structures. It is worth noting that our method is easy to implement and can be sided with any SfM software—as for example VisualSFM (Wu, 2013) or COLMAP (Schönberger and Frahm, 2016), both freely available—to let even non expert users build their own change detection system.

# 2 METHOD DESCRIPTION

Let $I_0$ and $I_1$ be two image collections of the same scene acquired at different times $t_0$ and $t_1$. At first, our method exploits SfM approaches to obtain estimates for the intrinsic and extrinsic camera parameters and a sparse 3D point cloud representing the scene, for both $I_0$ and $I_1$. We also retain all the correspondences that link 2D points in the images with 3D points in the model. Then, the initial point clouds is enriched with region growing approaches (Furukawa and Ponce, 2010) to obtain more dense 3D data. Note that camera positions and both the sparse and dense models obtained from $I_0$ and $I_1$ are expressed in two independent and arbitrary coordinate systems, since no particular calibration is used to register the two collections.

Hereafter we present the two main steps of the change detection method: $(i)$ to estimate the rigid transformation that maps the model of $I_0$ onto that of $I_1$, implicitly exploiting the common and fixed structures in the area, $(ii)$ to detect possible changes in the scene by comparing the registered 3D models.

## 2.1 Photometric Rigid Registration

Since 3D models obtained through automatic reconstruction methods typically include wrongly estimated points, before using a global registration approach—such as the Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992)—our system exploits the computed correspondences among image (2D) and model (3D) points, to obtain an initial estimate of the rigid transformation between the two 3D reconstructions.

Once computed the sparse reconstructions $S_0$ and $S_1$, from $I_0$ and $I_1$ respectively, for each 3D point we can retrieve a list of 2D projections and, additionally, for each 2D projection a descriptor vector based on the photometric appearance of its neighbourhood is also recovered. Exploiting this information, we can establish correspondences among images in $I_0$ and $I_1$, as follows. Each image in $I_0$ is compared against all images in $I_1$ and putative matches are found using the descriptor vectors previously computed. More in detail, let $f_0^i = \{\mathbf{m}_0^i, , \mathbf{m}_N^i\}$ be the set of $N$ 2D features in image $I_0^i \in I_0$ that have an associated 3D point, and $f_1^j = \{\mathbf{m}_0^j, , \mathbf{m}_M^j\}$ the set relative to $I_1^j \in I_1$. For each 2D point in $f_0^i$ we compute its distance w.r.t. all points in $f_1^j$ by comparing their associated descriptor vectors. Then, starting from the minimum distance match, every point in $f_0^i$ is put in correspondence with a point in $f_1^j$.

Since we know the relation between 2D and 3D points in $S_0$ and $S_1$, once obtained the matches between the 2D point sets, we can promote these relationships to 3D: Suppose the point $\mathbf{m}_n^i \in f_0^i$ is related to the 3D vertex $\mathbf{X}_0 \in S_0$, and similarly the point $\mathbf{m}_m^j \in f_1^j$ is related to the 3D vertex $\mathbf{X}_1 \in S_1$, then if $\mathbf{m}_n^i$ matches $\mathbf{m}_m^j$, $\mathbf{X}_0$ and $\mathbf{X}_1$ can be put in correspondence with each other. In this way, all the 2D matches found can be transformed into correspondences of 3D points.

3D correspondences obtained by comparing all images in $I_0$ with every image in $I_1$ are accumulated into a matrix $Q$. Since erroneous matches are possible, inconsistent correspondences could be present in $Q$. To extract a consistent matching set $\hat{Q}$, we count the occurrences of a match in $Q$ (note that, since a 3D point can be the pre-image of several 2D points, by comparing all images, we can find multiple occurrences). Starting from the most frequent, a match is selected and copied into $\hat{Q}$, then all matches in $Q$ that include points already present in $\hat{Q}$ are removed. Once completed this analysis, and emptied $Q$, $\hat{Q}$ will define a consistent 3D matching set without repetitions or ambiguous correspondences.

An initial rigid transformation between $S_0$ and $S_1$ is then computed using (Horn, 1987), by exploiting the 3D matches in $\hat{Q}$. Since wrong correspondences could still be present in $\hat{Q}$, we include a RANSAC framework by randomly selecting three correspondences per iteration. Inliers and outliers are found by observing the cross re-projection error. For example, if $T_{1,0}$ is a candidate transformation that maps $S_1$ onto $S_0$, the system evaluates re-projection errors between the transformed 3D $\tilde{S}_1 = T_{1,0}(S_1)$ and the 2D points of the images in $I_0$ and vice-versa, using 3D points from $\tilde{S}_0 = T_{1,0}^{-1}(S_0)$ and 2D points from $I_1$. The best transformation $T_{1,0}^*$ is estimated using the largest inlier set. Now, let $D_0$ and $D_1$ be the dense reconstructions obtained respectively from $S_0$ and $S_1$ using a region growing approach. As final step, our system runs an ICP algorithm using as input $D_0$ and $\tilde{D}_1 = T_{1,0}^* D_1$ to refine the registration.

Note also that, once the registration is completed, the 3D maps can be easily overlapped so as to visually observe the environment evolution and produce a 4D map (space + time).

## 2.2 Structural Change Detection

Once the 3D models are registered, surface normal vectors are computed for both $D_0$ and $D_1$: for each vertex, a local plane is estimated using neighbouring 3D points, then the plane normal is associated to the 3D vertex.

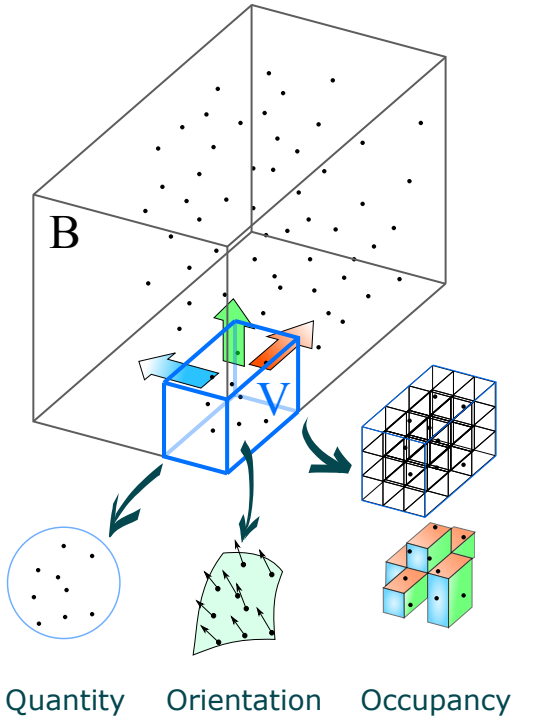



Figure 1: Graphical representation of our change detection method. Top: the volume $B$ and the voxel $V$ that shifts across the point cloud. Bottom: the three criteria used to detect changes: *Quantity*, that takes into account the number of 3D points enclosed in $V$, *Orientation*, that is related to the local 3D shape, and *Occupancy*,that describes the spatial point distribution in $V$.

Change detection works by shifting a 3D box over the whole space occupied by the densely reconstructed model. A bounding volume $B$ is created so as to include all points in $D_0$ and $D_1$, then a voxel $V$ is defined whose dimensions $(V_x, V_y, V_z)$ are respectively $(\frac{B_x}{10}, \frac{B_y}{10}, \frac{B_z}{10})$. $V$ is progressively shifted to cover the entire $B$ volume with an overlap of $\frac{3}{4}$ between adjacent voxels. Corresponding voxels in $D_0$ and $D_1$ are then compared by evaluating the enclosed 3D points with three criteria named *Quantity*, *Orientation* and *Occupancy* (see Fig. 1).

**Quantity Criterion.** The quantity criterion compares the effective number of 3D points in $V$ for $D_0$ and $D_1$. If their difference is greater than a threshold $\alpha$, the criterion is satisfied. Note that, even if counting the 3D points easily provides hints about a possible change, this evaluation can be misleading since $D_0$ and $D_1$ could have different densities—i.e. the same area, without changes, can be reconstructed with finer or rougher details, mostly depending on the number and the resolution of the images used to build the 3D model.

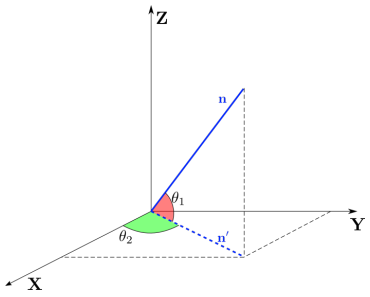


Figure 2: The angles used to describe normal vector orientation.

**Orientation Criterion.** To evaluate local shape similarity, the normal vectors of points included in $V$, both for $D_0$ and $D_1$, are used. For each normal $\mathbf{n}$ we define its orientation by computing the angles $\theta_1$, between $\mathbf{n}$ and its projection $\mathbf{n}'$ onto the $XY$-plane, and $\theta_2$, between $\mathbf{n}'$ and the $X$ axis (see Fig. 2). Quantized values of both $\theta_1$ and $\theta_2$ for all considered 3D points are accumulated in a 2D histogram to obtain a descriptor of the surfaces enclosed in $V$, both for $D_0$ and $D_1$. These descriptors are then vectorized by concatenation of their columns, and the Euclidean distance (spanning the dimensions of the histogram bins) is employed to evaluate their similarity. If the distance is greater than a threshold $\beta$, the orientation criterion is satisfied. Note that the reliability of this criterion is compromised if too few 3D points (and normal vectors) are included in $V$; hence, we consider this criterion only if the number of 3D points in $V$ is greater than a threshold value $\mu$ for both $D_0$ and $D_1$.

**Occupancy Criterion.** This last criterion is used to evaluate the overall spatial distribution of points in $V$. The voxel $V$ is partitioned into $3^3 = 27$ sub-voxels. Each sub-voxel is labeled as "active" if at least one 3D point falls into it; again, for both $D_0$ and $D_1$ we construct this binary occupancy descriptor. If more than $\gamma$ sub-voxels have different labels, then the occupancy criterion is satisfied.

If at least two out of these three criteria are satisfied, a token is given to all points enclosed in $V$, both for $D_0$ and $D_1$. Once $V$ has been shifted so as to cover the entire $B$ volume, a 3D heat-map can be produced by considering the number of tokens received by each 3D point in $B$, referred to as *change score*.

## 2.3 2D Change Map Construction

Since it could be difficult to appreciate the change detection accuracy just by looking at the 3D heat-map (examples of which are in Fig. 6), results will be presented in terms of a 2D change map. This is constructed by projecting the 3D points onto one of the input images, and assigning a colour to each projected point related to the local change score. Although better than the heat-map, the 2D map thus obtained is sparse and usually presents strong discontinuities (see e.g. Fig. 8). This is mainly due to the use of a sparse point cloud as 3D representation. Since any surface information is missing, we cannot account for correct visibility of 3D points during the projection. As a consequence, some points falling on scene objects actually belong to the background plane.

In order to improve the 2D map, we split the input image into superpixels, using the SLIC method (Achanta et al., 2012) (see Fig. 10a and 10c) and then, for each superpixel, we assign, to all the pixels in it, the mean change score value of the 3D points that project onto the superpixel. The resulting 2D change maps is denser and smoother w.r.t. the previous one (see Fig. 10b and 10d).

# 3 EXPERIMENTAL EVALUATION

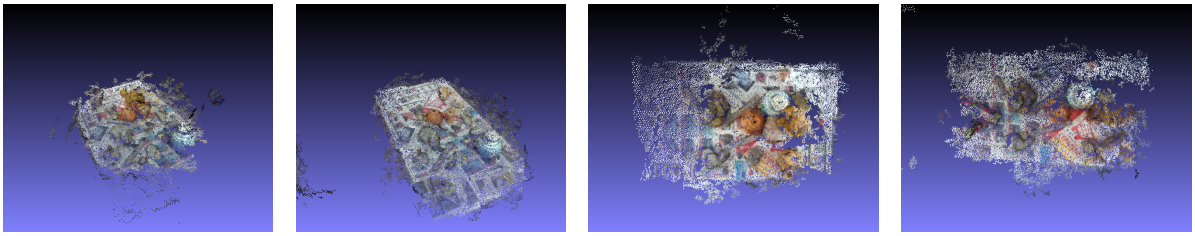
To evaluate the accuracy of the proposed method we ran two different tests with datasets recorded in our laboratory. In the first test ("Object removal"), we built a scene simulating an archaeological site where several artefacts are scattered over the ground plane and we acquired a first collection of images $I_0$. Then we removed two objects (the statue and the jar) and acquired a second collection $I_1$ (see Fig. 3).

For the second test ("Object insertion and displacement"), using a similar scene with objects positioned according to a different setup, we recorded the collection $I_2$. The original setup was then changed by inserting two cylindrical cans on the left side, by laying down the statue on the right side and by displacing in a new position the jar, the rocks, and the bricks. A new collection $I_3$ was then acquired (see Fig. 4).


(a)


(b)

Figure 3: Example frames of the two sequences. (a) $I_0$ sequence; (b) $I_1$ sequence. Note that the statue in the middle and the jar in the top left corner have been removed.



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

Figure 5: Dense 3D reconstructions obtained respectively from (a) $I_0$, (b) $I_1$, (c) $I_2$, and (d) $I_3$.

(a) (b)

Figure 4: Frames from the sequences $I_2$ (a) and $I_3$ (b). Two cylindrical cans were added in the left side, the statue on the right was laid down, and the jar, the rocks, and the bricks were moved to a different position.

All images were acquired with a consumer camera with resolution 640x480; for $I_0$ and $I_1$, 22 and 30 images were recorded respectively, while $I_2$ and $I_3$ are made of 22 and 18 images. Parameters were selected experimentally as follows: $\alpha$ is equal to the average number of points per voxel computed over $D_0$ and $D_1$, $\beta = 0.5$, $\gamma = 10$, and $\mu = 75$.

## 3.1 Qualitative Results

Figure 5 shows the dense 3D models obtained with SfM for the four image collections described before. To visually appreciate the detected changes, in Fig. 6 we present the obtained 3D heat-maps for the first ($I_0$ *vs* $I_1$) and the second ($I_2$ *vs* $I_3$) test setups. Hotter areas indicate a higher probability of occurred change.

As clear from the inspection of Fig. 6a, the removed jar and statue correspond to the hottest areas of the heat-map for the "Object removal" test. Similarly, all the relevant objects of the "Object insertion and displacement" test are correctly outlined by red areas in Fig. 6b. However, some false positives are present in both the 3D heat-maps: This is probably due to errors in the 3D reconstruction. Indeed, these false positives appear mostly on peripheral areas of the reconstruction, related to background elements that are under-represented in the image collection (the acquisition was made circumnavigating the area of interest) and thus reconstructed in 3D with less accuracy.

In order to better assess the performance of the proposed method, and also to observe the impact of the false positives visible in the heat-maps, we complemented the above qualitative analysis with a quantitative one.

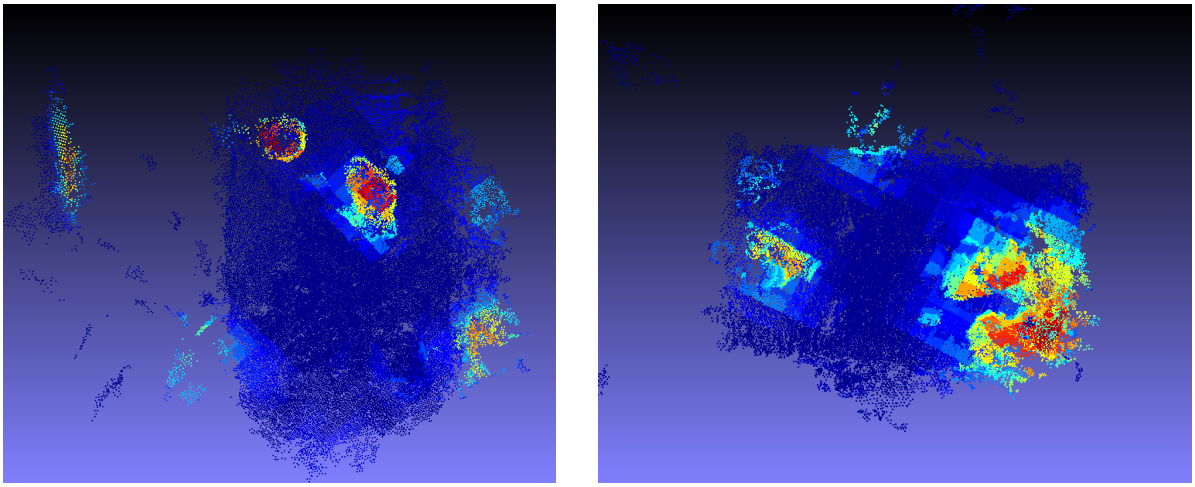
## 3.2 Quantitative Results

Ground-truth (GT) masks highlighting the changes occurred between the two image sequences were manually constructed. Fig. 7 reports two example of GT masks: Fig. 7b shows the changes between $I_0$ and $I_1$, reported in the reference system of $I_0$, while Fig. 7d depicts the comparison of $I_2$ and $I_3$, in the coordinate frame of $I_2$.

The GT masks were used to evaluate the Receiver Operating Characteristic (ROC) curve and assess the performance of our method for both the tests at hand.

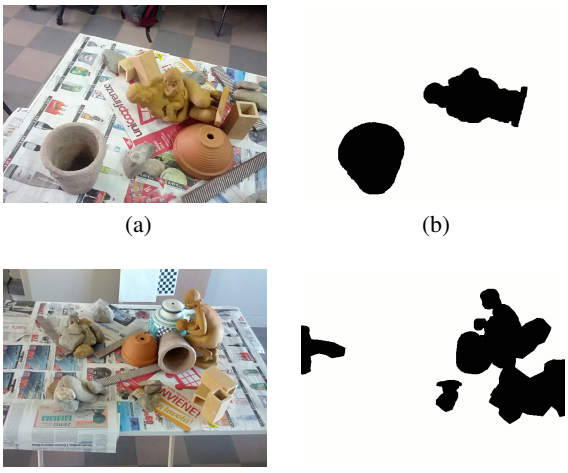
Fig. 8 shows the performance obtained with the sparse 2D change maps. In Figure 9 ROC curves obtained for $I_0$ *vs* $I_1$ and $I_2$ *vs* $I_3$ are reported. The values of the Area Under the ROC Curve (AUC) are respectively 0.76 and 0.89.

Improving the 2D change maps as described in Sect. 2.3 by exploiting superpixel segmentation (see Figs. 10a and 10c), yields denser and smoother maps—see Figs. 10b and 10d. The corresponding ROC curves are shown in Figs. 11a and 11b respectively. AUC values are 0.96 for $I_0$ *vs* $I_1$, and 0.92 for $I_2$ *vs* $I_3$. With the improved change maps, the AUC for the "Object insertion and displacement" test increases only by 3%, while in the "Object removal" test the AUC increases by almost 20% (0.19). This is due to the fact that the reconstructed 3D maps from $I_2$ and $I_3$ are denser that those obtained from $I_0$ and $I_1$ (see again Fig. 5). As a result, the sparse 2D change map for the "Object insertion and displacement" test is of better quality than its homologous for the "Object removal" test. For completeness, we report in Table 1 True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values obtained at the ROC cut-off point that maximizes the *Younden's*



(a) (b)

Figure 6: Heat-maps obtained from (a) $I_0$ *vs* $I_1$ and (b) $I_2$ *vs* $I_3$.



(a) (b)

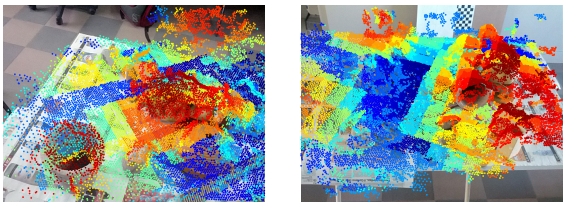
(c) (d)

Figure 7: Example of ground-truth masks: (a) an image from $I_0$ and (b) its GT mask, (c) an image from $I_2$ and (d) its GT mask. It is worth noting that, while the GT mask of $I_0$ *vs* $I_1$ is simply obtained by highlighting the removed object only, in the mask for $I_2$ *vs* $I_3$ highlighted areas have to account for removed, displaced or inserted objects. For this reason, the GT map (d) was obtained by fusing the masks of $I_2$ with those of $I_3$—after having performed view alignment based on the registered 3D models.

*Index*, i.e.

$$\max\{Sensitivity + Specificity - 1\} = \max\{TPR - FPR\} \quad (1)$$

where *TPR* and *FPR* are the True Positive Rate and the False Positive Rate, respectively.



(a) (b)

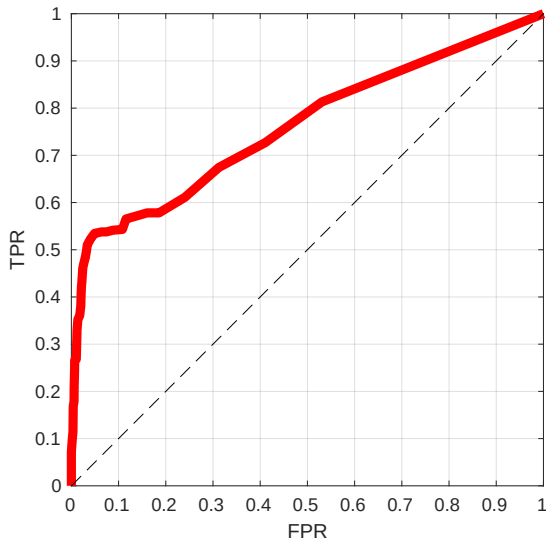
Figure 8: Sparse 2D change maps obtained by projecting the computed 3D heat-map onto a reference frame of $I_0$ (a), and $I_2$ (b).

Table 1: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) percentage obtained at the ROC cut-off point for each test sequence, plus the related AUCs. The subscript $_{SPX}$ indicates the improved change maps using superpixel segmentation.

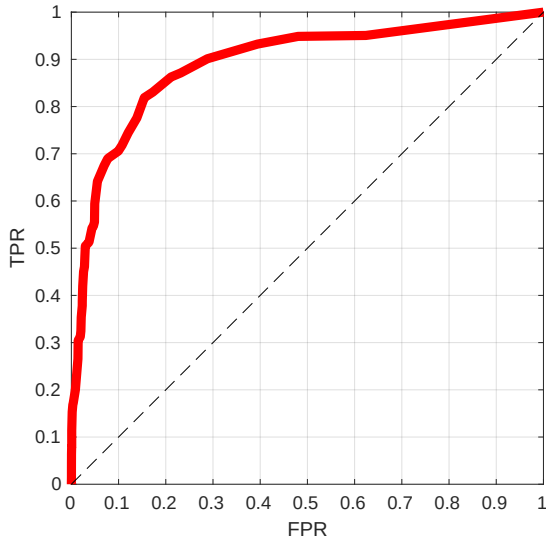
| Scene | TP | FP | TN | FN | AUC |
|---|---|---|---|---|---|
| $I_0 vs I_1$ | 0.12 | 0.04 | 0.74 | 0.10 | 0.76 |
| $I_0 vs I_{1SPX}$ | 0.13 | 0.10 | 0.76 | 0.01 | 0.96 |
| $I_2 vs I_3$ | 0.23 | 0.11 | 0.61 | 0.05 | 0.89 |
| $I_2 vs I_{3SPX}$ | 0.15 | 0.10 | 0.72 | 0.03 | 0.92 |

# 4 CONCLUSIONS AND FUTURE WORK

In this paper, a vision-based change detection method based on three-dimensional scene comparison was presented. Using as input two 3D reconstructions of the same scene obtained from images acquired at different times, the method estimates a 3D heat-map that outlines the occurred changes. Detected changes are also highlighted into the image space using a 2D change map obtained from the 3D heat-map. The



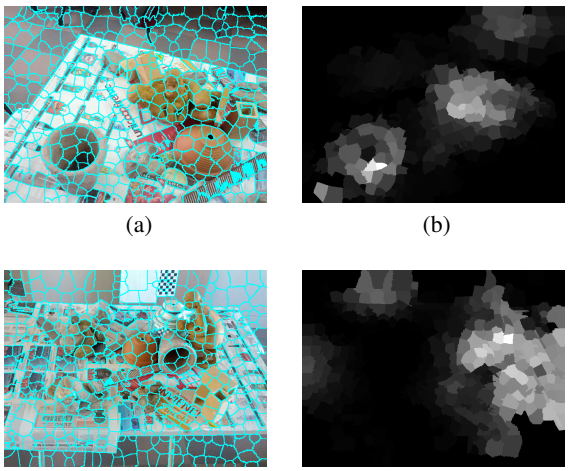

(a)




(b)

Figure 9: ROC curves obtained from the sparse 2D change map. In (a) ROC for $I_0$ *vs* $I_1$ achieving an AUC of 0.76. In (b) ROC for $I_2$ *vs* $I_3$ achieving an AUC of 0.89.

method works in two steps. First, a rigid transformation to align the 3D reconstructions is estimated. Then, change detection is evaluated by comparing locally corresponding areas. Three criteria are used to assess the occurrence of a change: a *quantity* criterion based on the number of 3D points, an *orientation* criterion that exploits the normal vector orientations to assess shape similarity, and an *occupancy* criterion to evaluate the local spatial distribution of 3D points. As a by-product, a 4D map (space plus time) of the environment can be constructed by overlapping the



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

Figure 10: Superpixel segmentation examples and improved 2D change maps.

aligned 3D maps. Qualitative and quantitative results obtained from tests on two complex datasets show the effectiveness of the method, that achieves AUC values higher than 0.90.

Future work will address a further improvement of the 2D change map, based on the introduction of surface (3D mesh) information into the computational pipeline.

## REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Ssstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.

Alcantarilla, P. F., Stent, S., Ros, G., Arroyo, R., and Gherardi, R. (2016). Street-view change detection with deconvolutional networks. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan.

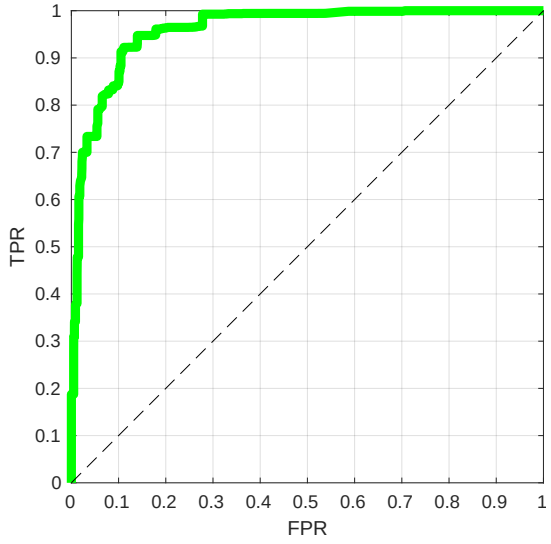
Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Brown, L. G. (1992). A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376.

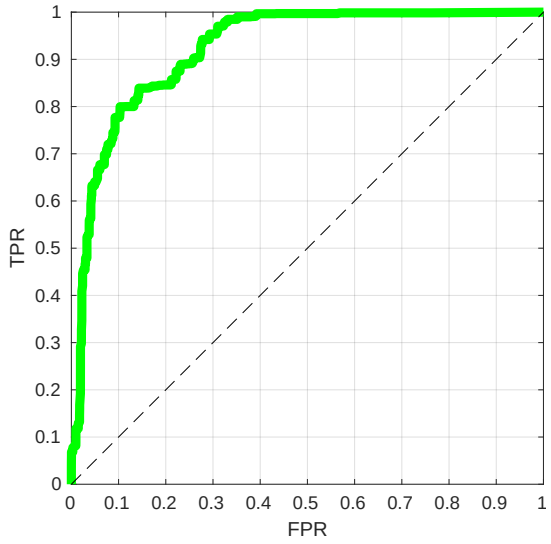
Cavallaro, A. and Ebrahimi, T. (2001). Video object extraction based on adaptive background and statistical change detection. In *Proc.SPIE Visual Communications and Image Processing*, pages 465–475.

Collins, R. T., Lipton, A. J., and Kanade, T. (2000). Introduction to the special section on video surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):745–746.

Dai, X. and Khorram, S. (1998). The effects of image misregistration on the accuracy of remotely sensed change




(a)




(b)

Figure 11: ROC curves obtained from the improved 2D change maps. In (a) ROC for $I_0$ *vs* $I_1$ achieving an AUC of 0.96. In (b) ROC for $I_2$ *vs* $I_3$ achieving an AUC of 0.92.

detection. *IEEE Transactions on Geoscience and Remote Sensing*, 36(5):1566–1577.

Fanfani, M., Bellavia, F., and Colombo, C. (2016). Accurate keyframe selection and keypoint tracking for robust visual odometry. *Machine Vision and Applications*, 27(6):833–844.

Fanfani, M., Bellavia, F., Pazzaglia, F., and Colombo, C. (2013). Samslam: Simulated annealing monocular slam. In *International Conference on Computer Analysis of Images and Patterns*, pages 515–522. Springer.

Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.

Mani, G. F., Feniosky, P. M., and Savarese, S. (2009). D$^4$AR – A 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. *Electronic Journal of Information Technology in Construction*, 14:129 – 153.

Palazzolo, E. and Stachniss, C. (2017). Change detection in 3D models based on camera images. In *9th Workshop on Planning, Perception and Navigation for Intelligent Vehicles at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

Pollard, T. and Mundy, J. L. (2007). Change detection in a 3-d world. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6.

Qin, R. and Gruen, A. (2014). 3D change detection at street level using mobile laser scanning point clouds and terrestrial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 90:23 – 35.

Radke, R. J., Andra, S., Al-Kofahi, O., and Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307.

Sakurada, K., Okatani, T., and Deguchi, K. (2013). Detecting changes in 3D structure of a scene from multiview images captured by a vehicle-mounted camera. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 137–144. IEEE.

Schönberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://colmap.github.io/.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition.

Taneja, A., Ballan, L., and Pollefeys, M. (2011). Image based detection of geometric changes in urban environments. In *2011 International Conference on Computer Vision*, pages 2336–2343.

Taneja, A., Ballan, L., and Pollefeys, M. (2013). City-scale change detection in cadastral 3D models using images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 113–120.

Toth, D., Aach, T., and Metzler, V. (2000). Illumination-invariant change detection. In *4th IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 3–7.

Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: principles and practice of background maintenance. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 255–261 vol.1.

Wu, C. (2013). Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision - 3DV 2013*, pages 127–134. http://ccwu.me/vsfm/.