

# 3D Change Detection at Street Level Using Mobile Laser Scanning Point Clouds and Terrestrial Images

Rongjun Qin <sup>a\*</sup>, Armin Gruen <sup>a</sup>

<sup>a</sup> Singapore-ETH Center, Future Cities Laboratory, 1 CREATE Way, #06-01 CREATE Tower, Singapore 138602.

\*Corresponding author at: Singapore-ETH Center, Future Cities Laboratory, 1 CREATE Way, #06-01 CREATE Tower, Singapore 138602. Tel: +65 9341 5509. Email: [rjin@student.ethz.ch](mailto:rjin@student.ethz.ch) (Mr. Qin)

**ABSTRACT:** Automatic change detection and geo-database updating in the urban environment are difficult tasks. There has been much research on detecting changes with satellite and aerial images, but studies have rarely been performed at the street level, which is complex in its 3D geometry. Contemporary geo-databases include 3D street-level objects, which demand frequent data updating. Terrestrial images provides rich texture information for change detection, but the change detection with terrestrial images from different epochs sometimes faces problems with illumination changes, perspective distortions and unreliable 3D geometry caused by the lack of performance of automatic image matchers, while mobile laser scanning (MLS) data acquired from different epochs provides accurate 3D geometry for change detection, but is very expensive for periodical acquisition. This paper proposes a new method for change detection at street level by using combination of MLS point clouds and terrestrial images: the accurate but expensive MLS data acquired from an early epoch serves as the reference, and terrestrial images or photogrammetric images captured from an image-based mobile mapping system (MMS) at a later epoch are used to detect the geometrical changes between different epochs. The method will automatically mark the possible changes in each view, which provides a cost-efficient method for frequent data updating. The methodology is divided into several steps. In the first step, the point clouds are recorded by the MLS system and processed, with data cleaned and classified by semi-automatic means. In the second step, terrestrial images or mobile mapping images at a later epoch are taken and registered to the point cloud, and then point clouds are projected on each image by a weighted window based z-buffering method for view dependent 2D triangulation. In the next step, stereo pairs of the terrestrial images are rectified and re-projected between each other to check the geometrical consistency between point clouds and stereo images. Finally, an over-segmentation based graph cut optimization is carried out, taking into account the color, depth and class information to compute the changed area in the image space. The proposed method is invariant to light changes, robust to small co-registration errors between images and point clouds, and can be applied straightforwardly to 3D polyhedral models. This method can be used for 3D street data updating, city infrastructure management and damage monitoring in complex urban scenes.

**Keywords:** 3D Change Detection; Terrestrial Images; Image-based Mobile Mapping System; Mobile Laser Scanner; Topographic LiDAR; Graph Cuts; Superpixel;

## 1. INTRODUCTION

Applications for geographical information system (GIS) are increasingly being found at the street level, e.g. in 3D car navigation systems ([Cornelis et al., 2008](#)) and infrastructure management, such as in telegraph poles, billboards. GIS elements such as bus stops, traffic signs are gradually being incorporated with the newly developed 3D GIS for smart city management ([Soheilian et al., 2010](#)), and the emerging popularity of street modeling is showing great potential for applications ([Haala and Kada, 2010](#); [Xiao et al., 2009](#)). However, How to keep these models in pace with reality in a fast and cost-efficient way remains an open question, Since the scene is more complex and contains more disturbances from unwanted changes than in the aerial view. A naïve re-modeling of these objects is both costly and time consuming due to the irregularity of street objects. Currently, the most common method of updating consists of manual checks on-site and rebuilding of models which are specified as changed, and this still requires a large amount of labor to identify the changes.

Beyond the purpose of the updating, the change itself is also of much interest. In the case of monitoring of visible damage to public infrastructures such as street benches, dustbins, telegraph poles, etc., the public facility department also wants to monitor whether any of these has malfunctioned and needs to be repaired ([LTA, 2014](#)). In this case, the 3D model or point clouds can be built right after these infrastructures are installed and then change detection is performed periodically to monitor these infrastructures.

Although there exists a large number of solutions ([Hussain et al., 2013](#); [Murakami et al., 1999](#); [Radke et al., 2005](#)) for change detection by using data sources taken from top views, such as satellite images and aerial images, few related works for conducting this at street level have been discussed. For traditional 2D/2.5D change detection from top views, the roof landscape is a major region of concern for change, and the depth is subdivided into limited layers such as buildings, trees and ground, which makes it easy to perform raster-based analysis. While in the ground view, the façade of buildings, road signs, bus stops, high and low vegetation constitute a rather complex scene, resulting in the failure of the traditional raster-to-raster (either Digital Surface Models (DSM) or satellite/aerial images) comparison. At the street level, the problems of traditional image to image comparison become magnified and these problems include: 1. Illumination and shape differences in shadow areas caused by the sun's position; 2. Perspective distortions of images taken from different angles and 3. Disturbances from cars, pedestrians, and unwanted changes like seasonally varying vegetation. One idea is to use the images to generate 3D geometry, and then compare the geometry between two epochs. However, automatic image dense matching is still a relevant research topic, and matchers can hardly perform well when there are a lot of multi-layer objects ([Gruen, 2012](#)).

The mobile laser scanning (MLS) system is mainly used for high-precision dense point cloud acquisition for measurement at the street level, as well as street object modeling. Operational MLS products provide cleaned point clouds (where pedestrians and cars are erased). Point clouds of different objects such as roads, trees and buildings, etc. are classified with semi-automated/manual approaches with commercial point cloud processing software such as [Terrasolid \(2013\)](#) to get reliable classification. By using overlays of accurately geo-referenced laser scanning point clouds from different epochs, it is possible to detect differences in 3D. However, periodical monitoring is both costly and storage-consuming, since laser scanners require high cost equipment and easily produce hundreds of Giga-bytes

of data per mission for processing. Compared to laser scanning point clouds, terrestrial images are much cheaper in terms of equipment and usually require less storage for a typical oblique photogrammetric configurations for change detection purpose (16 Megapixels, 80% in-strip overlap with 0.2-5 cm pixel footprint and normal JPEG compression rate), which makes it possible for frequent data acquisition. The idea behind the current work is to perform change detection by combining the two different data sources from different epochs to achieve both accuracy and space/cost-efficiency, which has rarely been explored before.

The proposed method of this paper first compares the geometrical differences between the point clouds and the terrestrial images by re-projecting between images with point cloud data to obtain the raw changes (initial cost), and then propagating the changes in the image space according to the similarity of various sets of information such as image color, depth, and pixel classes. In this work, we study the change detection directly from the point cloud instead of the 3D polyhedral model, because of the following reasons: 1) the automatic construction of a complete and correct 3D polyhedral model from point clouds or photogrammetric images is still in the research phase. Operational 3D modeling techniques still require heavy post-editing and manual drawing, while MLS data is recorded from a standard operational procedure and can be acquired efficiently. 2) For our procedure, objects do not need to be reconstructed as 3D models. 3) The method can be transformed directly onto 3D models. Change-detection using point clouds and terrestrial images contributes in a more general case, since 3D point clouds can be derived directly from 3D models.

In summary, this paper proposes a new approach for geometric change detection based on MLS point clouds taken in an earlier epoch and terrestrial images in a later epoch, or vice versa. The proposed method is invariant with respect to illumination conditions, since only images in one epoch are used. The proposed method provides change masks with clear boundaries by fully integrating the color, depth and class information for 3D street model updating and public facility monitoring. The rest of the sections are organized as follows: In section 2, we give a literature review of the state-of-the-art approaches for change detection under different scenarios; in section 3, the proposed algorithm is introduced in detail; in section 4, three experiments are performed and results are demonstrated; in section 5, the experiment results are analyzed to validate the methodology; section 6 discusses the tunable parameters of the proposed algorithm; in section 7, we conclude by discussing the pros and cons and possible further improvements of the proposed method.

## **2. LITERATURE REVIEW**

A spate of research has been conducted on change detection with satellite images and aerial images for various applications such as urban monitoring and map updating, where the objects of interest such as buildings and roads are represented at a medium or small scale ([Knudsen and Olsen, 2003](#); [Matikainen et al., 2010](#); [Rottensteiner, 2008](#)). Complexity and difficulties increase as the image resolution increases. There is limited research focusing on change-detection at street-level – images taken from a ground view, where the image contents focus on the façade of the objects on the ground, and the diameter of the pixel footprint ranges from 0.2 cm – 5 cm. Since change-detection in the top-

view scenario provides a good reference for developing methods in the street view scenario, we shall introduce the state-of-art change detection methods in both scenarios in this section.

## *2.1 Change detection in the top-view scenario*

### *2.1.1 Two-dimensional image-to-image comparison*

Two-dimensional image-to-image comparison methods have mainly been investigated in low resolution satellite images at the landscape level, where it is relatively easy to register images from different epochs. Changes in forest, water, soil and urban area are detected between two epochs for the analysis and monitoring of urban growth and its environmental impact. At the urban scale, using existing geo-databases, [Walter \(2004\)](#) proposed an object-based method that utilized the existing object class in the geo-information database and the corresponding multi-spectral image data as training sets to perform supervised classification. He then compared the result of the classification with the existing database to get the changed areas. [Bouziani et al. \(2010\)](#) used the semantic information of building footprints and remote sensing indicators such as NDVI (Normalized Difference Vegetation Index) to conduct a set of rules as constraints to detect changes between a single satellite image and building footprints. [Doxani et al. \(2010\)](#) proposed an object-based method by detecting changes in the classification result and used a morphological operator to filter out the insignificant results. These methods assume that only affine transformations exist between images from different epochs for co-registration, so they have succeed only for low resolution images of flat areas in the top view scenario, since the height variations of the landscape are not significant enough to cause perspective distortion. While for high-resolution true ortho-photos, such methods may suffer shadow problems and illumination changes.

### *2.1.2 2.5D to 2.5D geometry comparison*

A straightforward and robust way to avoid illumination and shadow problems is to compare the 3D geometry directly. The change detection techniques for the top view scenario, such as geometry comparison, was reduced to a 2.5D comparison, i.e. DSM raster comparison as well as the point cloud comparison in a top view where it was processed in a similar way as image to image comparisons. [Peng Gong et al. \(2000\)](#) subtracted two DSMs for evaluating height differences for the detection of changes. Other similar approaches can be found in ([Murakami et al., 1999](#); [Waser et al., 2008](#)). [Chaabouni-Chouayakh and Reinartz \(2011\)](#) argued that simple subtraction would give rise to false alarms due to data noise and inaccurate co-registration, and they proposed to adopt the shape information of the changed patches such as elongation and eccentricity to perform SVM (Support Vector Machine) classification to distinguish the building changes from uninteresting changes such as cars and trees. However, their method required high quality DSM for co-registration, and the threshold of height difference needed to be tuned according to different cases.

### *2.1.3 3D line based change detection*

[Niederöst \(2003\)](#) used stereo pairs to update 2D vector maps by using the detected building outline in the aerial images to improve existing 2D vectors. New buildings were detected and reconstructed using blob shapes and 3D lines, but complex structures cannot be handled correctly. [Champion et al. \(2010\)](#) proposed a two-step method: first 3D line primitives were extracted to verify the existing buildings, and the nDSM (Normalized Digital Surface Model) was used for new building extraction. However, the result contained a lot of false alarms, which was mainly due to the erroneous 3D line extraction of the small buildings. Therefore, 3D lines have served as robust features as compared to point features, whereas algorithms relying on 3D lines are sensitive to image noise and occlusions.

## *2.2 Change detection in the street view scenario*

### *2.2.1 3D to 3D comparison*

[Kang and Lu \(2011\)](#) detected disappearing changes by using terrestrial point cloud 3D building models from earlier epochs; [Girardeau-Montaut et al. \(2005\)](#) adopted an octree structure to divide 3D spaces for geometry comparison. The same strategy was adopted for detecting changes in areas of pipe work at high accuracy for security control but was based purely on terrestrial point clouds from different epochs. Both of the abovementioned works used the terrestrial laser scanning point clouds to detect geometrical changes on a single building or small objects. However the drawback of such methods lied in the high cost of the data, which created the barrier for periodical acquisition.

For the 3D geometric comparison methods, if the geometric information from different epochs is obtained from reliable sources such as laser scanning, the result will be stable and errors can be controlled within a certain range. However, if the geometric information is obtained from automatic image matching, especially at the street level, the results will be less reliable due to the large uncertainties of the matching results ([Taneja et al., 2011](#)).

### *2.2.2 Change detection with oblique images*

[Eden and Cooper \(2008\)](#) proposed a 3D line based approach to reduce the pixel-wise computation of change detection and to increase the robustness of the algorithm as well. A set of 3D lines were extracted from the image set from an earlier epoch using a multi-view line matching algorithm, and then a set of threshold-based rules were formulated for labeling the changed lines to “changed”, “unchanged”, “occluded” and “removed”. However, the method suffered the same problem of the line-based change detection algorithm mentioned in section 2.1.3, that is, shadows and unreliability of line detection due to image noise.

[Taneja et al. \(2011\)](#) proposed a voxel-based method to detect changes in a street environment. The probability of each voxel was computed by color consistency from multi-views, and changes were propagated in 3D spaces. However, the voxels would always bound the actual change, which could not depict changes following the natural boundaries of the objects. [Pollard and Mundy \(2007\)](#) proposed a method to treat changes between two sets of oblique images in a 3D environment. The algorithm used voxels as basic elements for labeling whether changes occur, and adopted a space carving method

([Broadhurst et al., 2001](#)) on the image sets from an earlier epoch to compute the probability of visibility of a voxel due to its color distribution. Then the probability of each voxel being visible on the surface was updated with the image sets from a later epoch. Changed pixels and geometry would result in low probabilities in surface voxels, and these voxels could be indicated as changes. Nevertheless, the method was very sensitive to alignment errors in object boundaries, where high color variances occurred inside voxel covering boundaries.

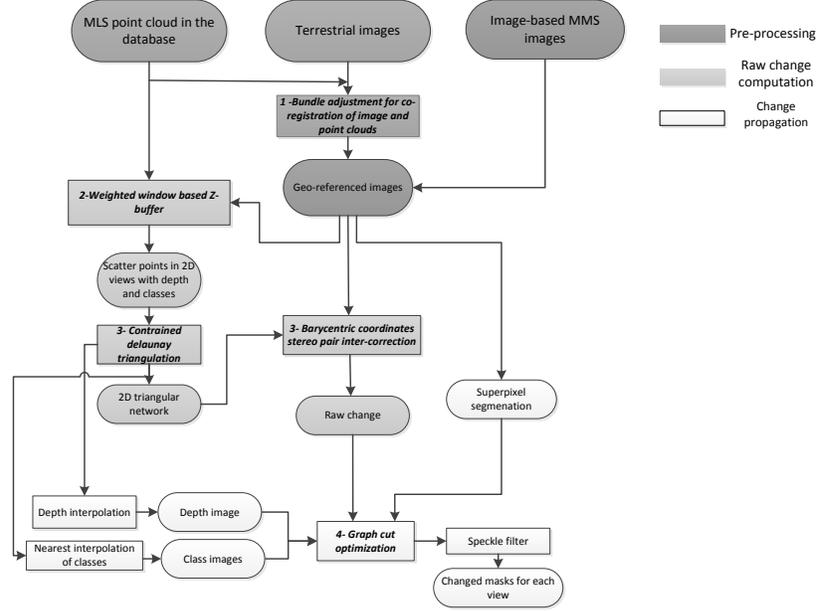
These methods implicitly adopt a multi-matching model within the images where a large number of artifacts occur due to unreliable geometry and voxel sampling. This results in methods which are only able to detect changes that are large (e.g. taking more than 5% of the image content) and occurring in relatively simple scenes (e.g. less than 10 objects in the image). Moreover, it does not take into account the relationships of neighboring pixels (e.g. color similarity, depth similarity), which serves as a crucial factor to propagate changes following the natural object boundary ([Taneja et al., 2011](#)).

### 3. METHODOLOGY

The proposed method in this paper compares point clouds and terrestrial images with their implicit geometric consistency to obtain the raw changed areas: images of a stereo pair are correlated from one to the other using the point clouds. Raw changes in the two images are then computed based on the pixel color difference, and finally, the raw changes are propagated in each view of the image while taking into account the coherence of neighboring pixels with depth, color and class.

The adopted strategy has several advantages: (1) the images are taken in a short time, so the illumination changes of the sun and environment can be ignored; (2) explicit geometry will not be generated from the images which avoids matching artifacts, and saves computational time as well and (3) pixel coherence in image space gives more reliable information than propagating fragmental changes in 3D space. Supposing the point cloud was scanned at time  $t_1$ , and the terrestrial images are taken at time  $t_2$  (where  $t_2 > t_1$ ), the method will automatically mark out the changed area in each view with binary masks; this could save a large amount of time when updating 3D data, by limiting images only with changes such that the operator can decide which places in the street should be updated.

The proposed method consists of three main parts: (a) Pre-processing—from point cloud to view dependent 2D mesh; (b) raw change computation by comparing geometrical consistency with multi-stereo image pairs and (c) change propagation by using graph cut optimization, with a speckle filter ([Davies, 2004](#)) which searches and deletes very small segments (less than 200 pixels). The flow chart of the algorithm is shown in Fig. 1.



**Fig. 1.** Workflow of the proposed algorithm: the numbered parts with bold italic font correspond to subsections in the following.

### 3.1. Co-registration of images and point clouds

There are many ways to co-register point clouds with images, and some automatic methods have been proposed in the literature ([González-Aguilera et al., 2009](#); [Makadia et al., 2006](#)). Most of the existing methods are largely dependent on point cloud intensity and RGB values of the images, but they fail either in homogenous areas which lack feature correspondence, or they fail due to the noise of the intensity coming from the z-buffer problem ([Girardeau-Montaut et al., 2005](#)), where intensity of the points from the back-face are projected as the front-face points. For images taken from image-based mobile mapping system (MMS), they are geo-tagged with relatively accurate positions (a few centimeters to decimeters) and orientations (a few tens of seconds) ([POSLV, 2013](#)) as a result of the GPS/IMU system, so the images are co-registered to the point cloud in the first place. For terrestrial images, the geo-referencing can be done by bundle adjustment, through either additional surveys for control points, or manually/semi-automatically selected control points from the point cloud.

In this work, the free-network bundle adjustment is firstly done on the terrestrial images, and the control points are manually collected from feature points in the point cloud such as roof corners and road signs, and the image orientation parameters are then optimized with control points. After bundle adjustment, the re-projection error in the image space was about 3 pixels in each direction. Since recognizing corresponding points between the point cloud and image at sub-pixel level is difficult, another option which is also more accurate is to use ground control points. The proposed algorithm is robust to small co-registration errors and will be demonstrated in the experiment.

After the point clouds and the images are co-registered, we introduce the following terms: given a point cloud set  $\Omega = \{P_i\}$  and a set of geo-referenced images  $\{I_i\}$ , the resulting image coordinate of

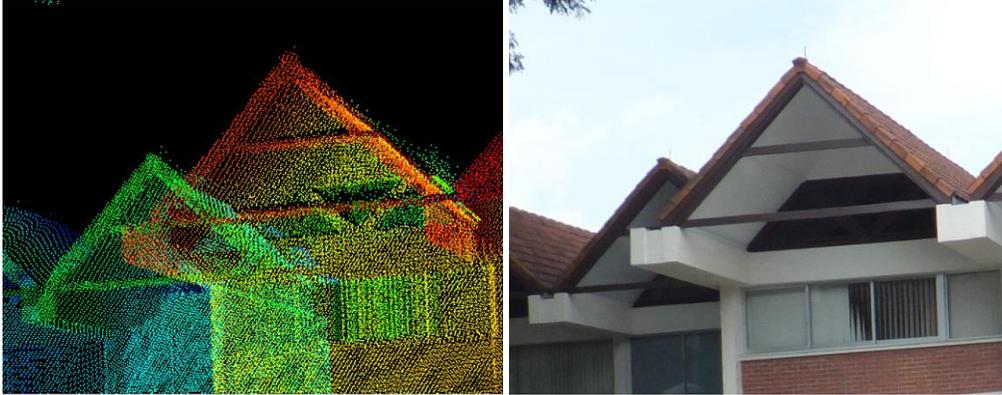
projecting point  $P_t$  on image  $I_i$  is  $m_t^i$ , with depth value  $z_t^i$ , where  $(m_t^i, z_t^i) = \mathcal{P}_i(P_t)$ , and  $\mathcal{P}_i$  is the projection function of the  $i_{th}$  image which project a 3D point onto the  $i_{th}$  image plane.

### 3.2 Weighted window based image z-buffer for decimating points

The relation between image pixels and the 3D object points can be computed by using the co-linearity equation, via shooting rays from the image to the 3D geometry. However, for geometry represented by point clouds, some rays may not shoot at any point or at some points on the back surface which should be occluded by the front face. In this case, surface reconstruction is needed. However, automatic surface reconstruction from unstructured point clouds is still in the research phase; holes and incorrect patches occur quite often, even for dense MLS point clouds. Therefore, we perform the surface reconstruction in 2D image space for each view by creating a 2D triangular mesh based on the projected points from the point clouds with depth values on each image plane. The projection of point clouds onto an image plane leads to the well-known z-buffer problem; that is, points from the back face may be projected through the interspace of the front-face points (Fig. 2) (Zhou, 2013). Since the MLS point clouds are dense and distributed regularly with similar point intervals on the object space, block window patches could serve as a basic z-buffer unit to filter out points projected from the back face. Suppose a square window of size  $w \times w$  is used, and the image divided into  $\left(\frac{width}{w}\right) \times \left(\frac{height}{w}\right)$  units. For each unit  $j$ , its projected point  $l_j$  is:

$$l_j = m_i, z_i = \min(z_t), t \in \text{pixels within unit } j \quad (1)$$

This block-window-based Z-buffer algorithm (BWZB) with a large window will filter out most of the back-face points but it will also decimate a lot of front-face points, whereas with a small window size, more back-face points will remain. Therefore, it is difficult to choose an appropriate window size for filtering the back-face points.



(a) Depth coded point clouds

(b) Corresponding image

**Fig. 2.** Z-buffer problem to project point clouds onto images: points in the back face are projected onto the front through interspaces of the front faces.

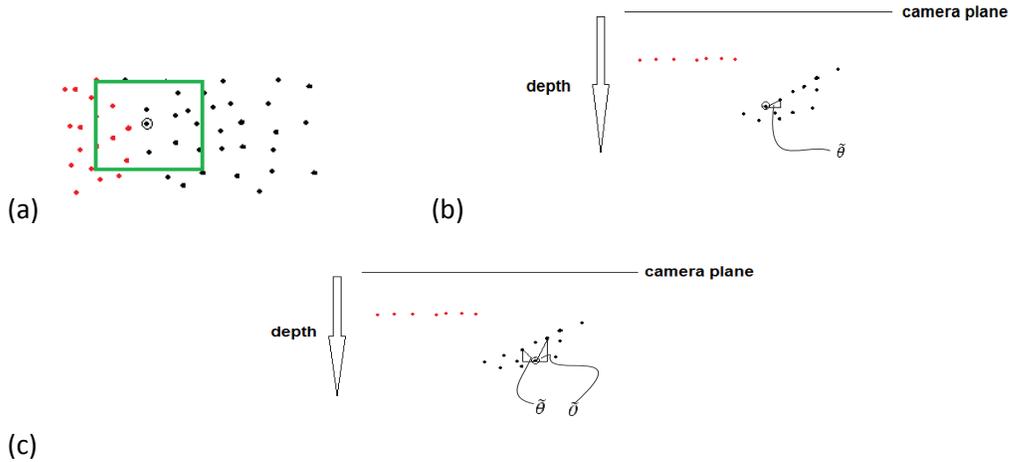
A weighted window based filtering strategy which takes into account the color information and angular constraint is carried out for filtering the back-face points. It first filters the back-face points using BWZB with a small window size (e.g.  $5 \times 5$ ) to keep the front-face points as dense as possible. Then for each projected 3D point on the image after BWZB, a weighted window with size  $w_1 \times w_1$  centered at the point is generated as:

$$W(p) = \exp\left(-\left(\frac{\Delta d}{\gamma} + \frac{\Delta c}{\delta}\right)\right), \text{ where } \Delta d = \|x_p - x_q\|, \Delta c = \|I(p) - I(q)\| \quad (2)$$

$q$  is the central pixel of the window with pixel coordinate  $x_q$ , and this implies pixels with higher color proximity and high 2D geometric proximity have higher probability of being on the same surface, and this can prevent decimating points in depth discontinuities, since most of the depth discontinuity will result in color differences.  $\gamma$  and  $\delta$  control the weights of 2D geometric proximity and color proximity, and in our experiment, they are fixed at  $\gamma = 100, \delta = 10$  for normal RGB images (24-bit depth).  $W(p)$  ranges from 0 to 1, and a threshold  $\tau = 0.5$  is given to determine which pixels in a window are activated to perform the angular constraint. Let the depth of the central pixel be  $z_c$ , and  $x_c, y_c$  be the planimetric coordinates on the focal plane, and  $z_i, x_i, y_i$  be the depth, and planimetric coordinates of any point inside the window. The largest angle in this window can be computed as follows:

$$\theta = \max\left\{\arctan\left(\frac{z_c - z_i}{\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}}\right) \mid z_i < z_c \text{ and } W(i) > \tau\right\} \quad (3)$$

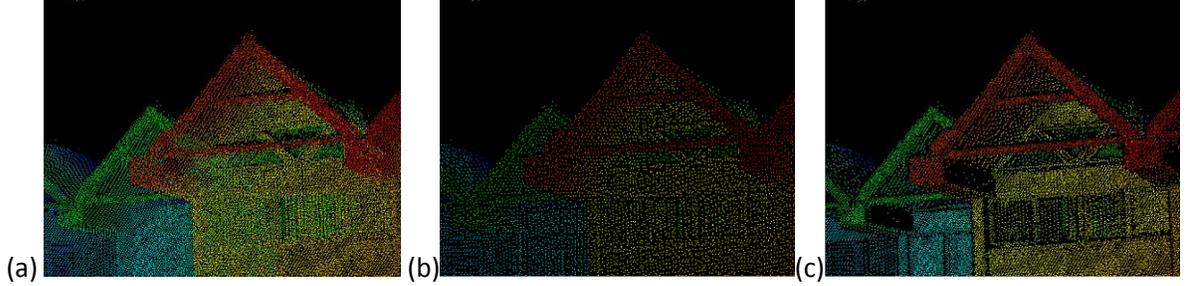
Points with  $\theta$  that is larger than a given threshold (we fixed this parameter as 45 degree in our experiments) will be seen as a back-face point and will be filtered out. An intuitive illustration of the process is shown in Fig. 3.



**Fig. 3.** Intuitive illustrations of the weighted z-buffer strategy: (a) Visualization of the projected points in the perspective image, we use only two colors for simplicity. The green window shows the local area for computation and the point with black circle is the central point. (b) A profile of the projected points, where the central point is not erased (c) A profile of the projected points, where the central point is erased.

Fig. 3 shows the projected points on the perspective image, and we only use two colors for illustrative purpose. In the first step (as shown in Fig. 3(a)), points that have the similar color and small 2D image distance to the central point (marked in circle) is considered as the supporting points, and this is done by

thresholding (with  $\tau$ ) the similarity computed by equation (2). In the second step, the angular constraint is applied (as shown in Fig. 3(b, c)).  $\tilde{\theta}$  is computed for supporting points that have smaller depth than the central point as, and  $\theta$  is computed as  $\theta = \max(\{\tilde{\theta}\})$ . If the central point with  $\theta$  is larger than 45 degree, it will be erase as a back-face point. The weighted window size  $w_1$  should be larger than  $w$  to cover more points for computation. The weighted window based Z-buffer algorithm (WWZB) keeps more front-face points and prevents the edge points from being smoothed by the windows as well (Fig. 4).



**Fig. 4.** (a) BWZB algorithm with  $w = 5$  (17840 points); (b) BWZB  $w = 10$  (5799 points); (c) WWZB algorithm with  $w = 5$ ,  $w_1 = 15$  (11518 points with clearer edges).

Fig. 4(a) shows that with a small  $w$  for BWZB, there are many back-face points left, while with a larger  $w$  shown in Fig. 4(b), the back-face points are reduced while most of the front-face points are also filtered. Fig. 4(c) shows that using a larger weighted window ( $w_1 = 15$ ) will keep more points with clear edges, due to the incorporation of the color information. These projected points will be further triangulated for extrapolating the depth and classes pixel-wise.

In our experiments, we set  $w_1$  to 15 (pixels) as an empirical value, and the WWZB could filter more than 90% of the back-face point.  $w_1$  is related to point density of the laser scanning data on the objects, which should be adjusted accordingly for point clouds with different density. It should be large enough to cover at least one front-face point, but small enough to reduce the computation time.

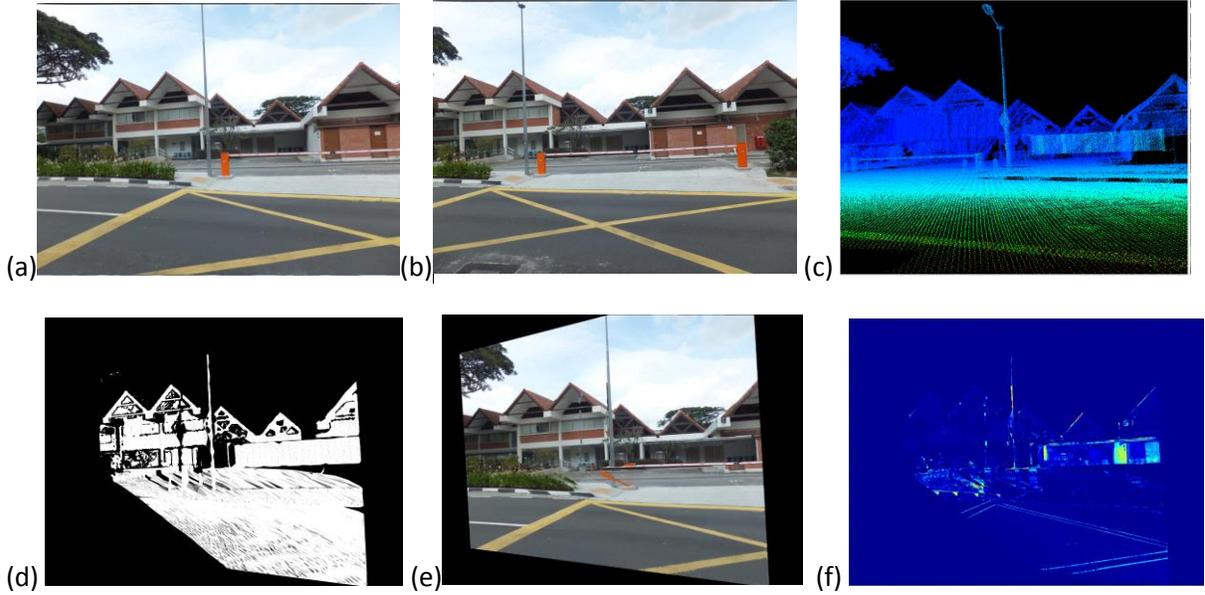
### 3.3 Multi-stereo image inter-correlation for raw change computation and ambiguity elimination

In order to evaluate the geometrical inconsistency between the reference Image  $I_p$  and the point clouds, neighboring images  $I_q$  ( $q \in N(p)$ ,  $N(p)$  is the neighboring set of  $p$ ) are correlated onto image  $I_p$  as  $I_{qp}$  (inter-correlation in stereo pairs) based on the point clouds, color differences of each pair are computed in pixel-wise as raw changes. To compute the inter-correlation  $I_{qp}$  in stereo pair  $\langle I_p, I_q \rangle$ , the stereo pair is first rectified as epipolar pairs  $\langle \bar{I}_p, \bar{I}_q \rangle$  so that the pixel correspondence of a stereo pair can be depicted with disparity  $L = \{l\}$ , where  $\bar{I}_q(x, y)$  and  $\bar{I}_p(x + l, y)$ , are projections of the same 3D object point, but the disparity value is only available for pixels which are projections of the point clouds. A 2D Delaunay triangulation is then performed on the discrete points obtained in section 3.2 on  $\bar{I}_q$ . The triangles are further filtered by the depth values of the vertices and the length of the edges: triangles with large depth difference among the three vertices are likely to be across depth discontinuities, which might be affected by occlusions. Moreover, since points lying on a surface should be distributed evenly, triangles with sharp angles and long edges in the 2D image plane are likely to be

wrong connections between separated surfaces. In our experiments, we adopt 0.5 meters as the threshold for depth value and 30 pixels for the edge length to remove the irregular triangles. After filtering the triangles, the disparities of pixels on  $\bar{I}_q$  are interpolated with barycentric interpolation, and the inter-correlation is computed as:

$$\bar{I}_{qp}(x, y) = \bar{I}_q(x - l, y), d \in L \quad (4)$$

$I_{qp}$  stands for the inter-correlated image from  $I_q$  to  $I_p$  (Fig .5(e)), computed by inversely rectifying  $\bar{I}_{qp}$ . Since depth information and class information are available only for a part of the pixels on the image plane, depth of each pixel is interpolated with barycentric interpolation and the class label for each pixel is extrapolated by a nearest neighborhood interpolation.



**Fig. 5.** Inter-correlation between stereo pairs based on point clouds. (a) The reference image  $I_p$ ; (b) the neighboring image  $I_q$  to be correlated; (c) the point clouds; (d) filtered 2D triangular mesh on  $I_q$ ; (e) the inter-correlated image  $I_{qp}$  (occluded pixels are filled with pixels from  $I_p$ ); (f) color differences  $|I_p - I_{qp}|$  by direct subtraction coded with Jet color map.

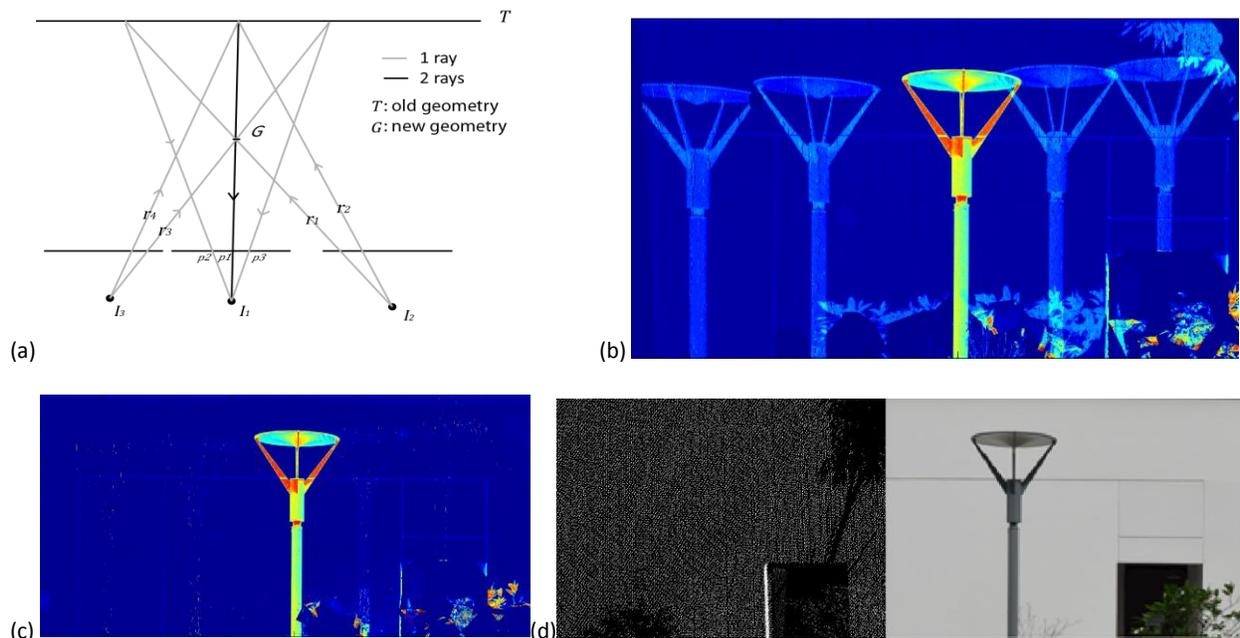
A single stereo pair is able to show the geometrical inconsistency. However, ambiguity exists and Fig. 6(a) shows a simple example. Assuming  $G$  and  $T$  have distinguishable colors where  $T$  denotes the old geometry existing in the database at an earlier time, and  $G$  denotes a new object that is captured by images at a later time which is not present in the old geometry.  $I_1$  is the reference view, and the neighboring view  $I_2$  projects each pixel onto the old geometry, and then re-projects the object point onto  $I_1$ .  $I_2$  shoots ray  $r_1$  and  $r_2$  to  $T$  and they are re-projected onto the image plane of  $I_1$  in point  $p1$  and  $p2$ . Both of them are then projected through the new object  $G$ , and they show color differences, while only  $p1$  is the real change.

With multiple overlaps, there are multiple stereo pairs referring to the reference image  $I_1$ , and this gives redundant information for the geometric inconsistency to eliminate the ambiguity. As shown in Fig. 6(a), the inter-correlation from  $I_3$  to  $I_1$  contributes one more ray for the color inconsistency on  $p1$ , and

this implies that the real changes can be distinguished from the false changes by counting the rays of the pixels which have color inconsistency. Fig. 6(b) shows the ambiguity by summing up the color differences of all the stereo pairs related to the reference image. Therefore, with multiple stereo pairs, we compute the raw change in the following way.

$$C_p = \min_{q \in N(p)} \{|I_p - I_{qp}|\} \quad (5)$$

where “min” is operated for every pixel. For computational convenience,  $C_p$  is normalized to  $[0, 1]$ . The resulting raw change is shown in Fig. 6(c).



**Fig. 6.** A simple case showing the ambiguity of changes eliminated by multi-stereo images. (a) Real changes receive more rays that contain color differences; (b) sum of color difference of all the pairs, the real changed area shows higher values whereas ambiguity exists with lower values; (c) raw change computed with equation (5) where ambiguity is eliminated by sorting the number of rays contributing to the changes; (d) the reference image and the point cloud, the street light pole is a new object.

The environmental illumination is assumed to be invariant due to the short time interval between adjacent image acquisitions. However, there may exist different micro-environments for each view which renders different digitized grey values for the same point. Therefore, we adopted a classic histogram stretching algorithm (Zhang and Main, 1990) to balance the digitization difference of different views. Changes are computed by selecting the minimal color differences by comparing different stereo models. To avoid disturbances from trees and cars, we simply assign a small weight to the raw changes that reside in the class of the trees and roads.

### 3.4 Graph cut optimization of changed area based on the superpixel over-segmentation

The raw changes shown in Fig. 5(f) are affected by artifacts that come from co-registration errors or triangulation errors, or the irregular shapes of trees. Moreover, color consistency is not an absolute

factor for changes, since for homogenous areas, geometrical changes may not lead to color differences. Therefore, these areas need to be propagated by similarity measurements so that a complete-change mask can be computed from fragmental color inconsistencies. We adopt the graph cut optimization to propagate the possible changes by taking the color, depth and class similarity into account. The classic graph cuts algorithm ([Vicente et al., 2008](#)) tries to minimize the energy (cost) function in the following form:

$$E(X) = \sum_{p \in V} D_p(x_p) + \beta \sum_{(p,q) \in E} S_{pq}(x_p, x_q) \quad (6)$$

Where  $(V, E)$  is a general graph;  $V$  is the vertex set and  $E$  is the edge set,  $x_p = 0$  or  $1$  for each node.  $D_p(x_p)$  is the node energy, which defines the energy for each node when  $x_p$  is  $0$  or  $1$ ;  $S_{pq}(x_p, x_q)$  is the smooth term that defines neighborhood relationship for each edge, and  $\beta$  controls its weight, and the minimization of the energy function can then be transformed into a max-flow/min-cut problem ([Boykov and Kolmogorov, 2004](#)). We define every pixel in the reference image as a node, and 4-neighborhood connectivity in the image grid as edges. For each pixel  $p$  in the reference images,  $x_p = 1$  represents a changed pixel, while  $x_p = 0$  represents an unchanged pixel. We define the node energy as:

$$D_p(x_p = 0) = \left( \frac{\exp(aC_p - b)}{\exp(aC_p + b)} - m \right) / s$$

$$D_p(x_p = 1) = 1 - D_p(x_p = 0) \quad (7)$$

where

$$m = (1 - b)/(1 + b), \quad s = \frac{\exp(a - b)}{\exp(a + b)} - m,$$

$C_p$  is the normalized raw change computed from Equation (5) in section 3.3. In equation (7),  $\frac{\exp(aC_p - b)}{\exp(aC_p + b)}$  maps the raw change into a "S" shaped curve which controls the sensitivity of the node energy to the color differences,  $a$  controls the steepness of the curve and  $b$  controls the curve shifts,  $m$  and  $s$  are normalization parameters so that the domain of the node energy is  $[0, 1]$ . In practice,  $a = 45$ ,  $b = 11$  is a reasonable value for interpreting the real changes and at the same time, it also suppresses the illumination difference. If the raw change of one pixel is high, it means a high cost for it to be unchanged and a low cost to be changed. The similarity measurement is defined by considering the color similarity, depth similarity and class similarity:

$$S_{pq}(x_p, x_q) = \begin{cases} \min\{\exp(-||I_p - I_q||), \exp(-||z_p - z_q||)\} \cdot \delta(G_p - G_q), & \text{if } x_p = x_q \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where  $z_p$  denotes the depth of the pixel  $p$ , and  $G_p$  denotes the class index of the pixel  $p$ .  $\delta(\cdot)$  is impulse function which is defined as:

$$\delta(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{otherwise} \end{cases} \quad (9)$$

This enables changed pixels to extend the possibility of the change to the neighboring pixels in the same class with high color similarity or depth similarity. Since the image dimension can easily go up to 10 mega pixels, it is computationally inefficient to take every pixel as the node. Therefore, we adopt a speed-up strategy by using superpixels ([Achanta et al., 2012](#)) as nodes. Superpixel segmentation is an over-segmentation technique that groups small patches of pixels with similar colors. The superpixel segmentation adopts simple K-mean clustering methods for initially well-distributed cell centers by iteratively grouping pixels with the Euclidean distance of a 5D vector which considers color ( $R, G, B$ ) as well as 2D geometric distance (image coordinates  $X, Y$ ):

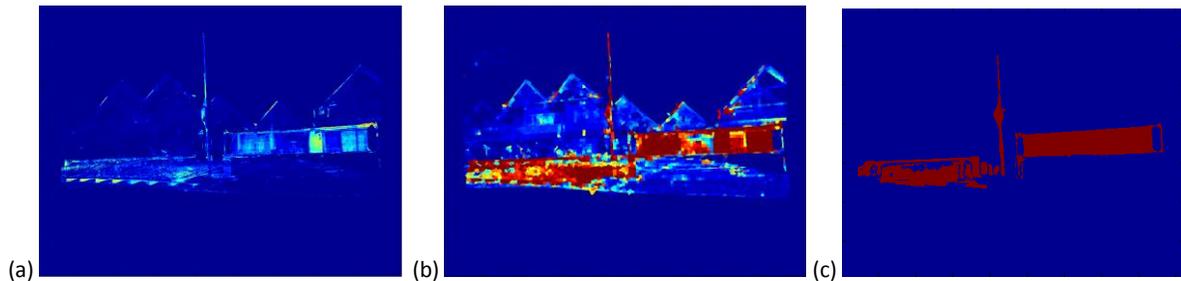
$$D_s = ||Lab_k - Lab_i|| + \frac{M}{S} ||x_k - x_i|| \quad (10)$$

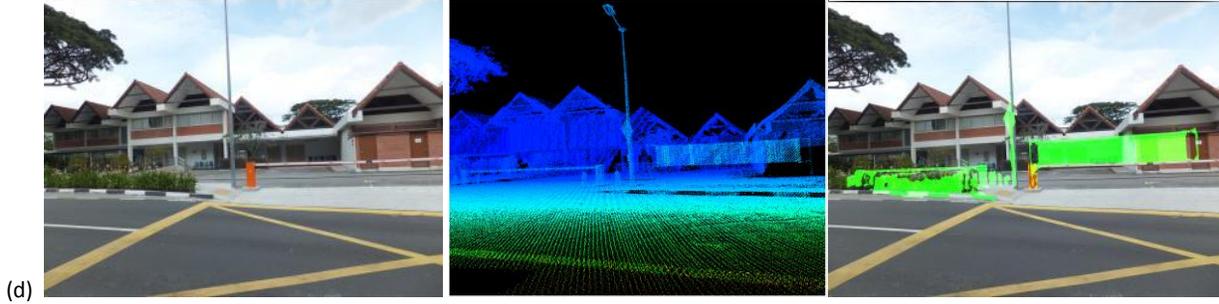
where  $Lab_k$  and  $Lab_i$  represents the *CIELAB* color space of the  $k_{th}$  center and any of the pixel in the local region, while  $x_k$  and  $x_i$  denote the respective pixel coordinates.  $M$  and  $S$  are compact factors which control the compactness of the resulting superpixels. We modify this method by incorporating the pixel classes by:

$$D_s = ||Lab_k - Lab_i|| + \frac{M}{S} ||x_k - x_i|| + \delta(G_k - G_i) \cdot \alpha \quad (11)$$

where  $\alpha$  is a large penalty which enforces pixels in each superpixel segment to be in the same class. Therefore, the resulting superpixels lie along the class boundaries which ensure that each superpixel segment represents a meaningful unit of change. Thus, for each superpixel, the color value and depth value are assigned as the average value over all the pixels in the superpixel.

The size of superpixel should be large enough to reduce the computational time, but smaller than significant objects. If the superpixel is too large, small color edges might be averaged in the cell size, which results in smoothed boundaries of the detected segments. Moreover, the compactness parameters  $M, S$  control the extent that each cell follows the image edges. If  $M/S$  is large, the superpixels tend to be isotropic rather than following image edges, thus image edges tend to be averaged in the superpixels, which results in smoothed boundary of the detected segments. While if  $M/S$  is too small, the superpixels becomes irregular and can be affected by insignificant edges, which may results in irregular shapes of the detected changes. In our experiments, the cell size is  $50 \times 50$  pixels, and the  $M/S$  is set to be 20.





**Fig. 7.** Process from change to the final determination of changes. (a) Raw Change  $C_p$ ; (b) initial probability of changed area  $D_p(x_p = 1)$ ; (c) changed area optimized by graph cuts; (d) change detection result: left: terrestrial images in later epoch; middle: depth coded MLS point cloud; right: change detection result (marked in Green).

Fig. 7(a-d) show the process from raw change to the final determination of changes by graph cut optimization based on superpixel segmentation. By visual comparison of point clouds and the images, we found that change occurred mainly in three parts: a billboard was removed (in the right part of the image); a small diamond-shaped board in the street light pole was removed (in the middle); a hedge was built, replacing the parking railing and bare ground (in the left part of the image). In Fig. 7(a, b), there are some unchanged areas that show high probability of change (at the edge of the buildings) which mainly comes from co-registration errors. In some areas where real change occurs, the probability of it being a changed area is low because these areas are homogenous areas, and even though the correlated area in  $I_{qp}$  is not the same physical area in the reference image  $I_p$ , the color of these different areas might be the same, e.g., walls, grounds, etc. As described in equation (8), those isolated superpixels with different probabilities of change are strongly connected with the neighborhoods by similarity in color, depth and classes. Graph cut optimization is performed and the results are shown in Fig. 7(c, d). The result shows that the optimization process absorbs small segments and enlarges the changed homogenous regions following the boundaries of the meaningful segments, which renders results at a finer detail than state-of-the-art change detection methods in complex environments. There might still be some isolated small segments resulting from artifacts with very high energy. In the final step, we simply filter out segments which are smaller than a given threshold using a speckle filter. The proposed algorithm can be easily applied to cases where 3D models of the street objects are available by directly performing stereo pair inter-correlation on 3D models without dealing with the z-buffer problem of a discrete point cloud.

#### 4. EXPERIMENT

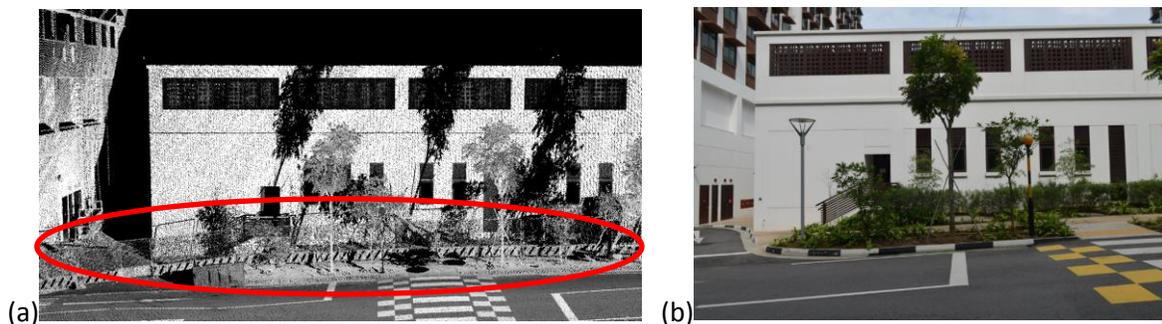
In order to validate the methodology, we performed three experiments in the street with terrestrial images of different street environments and complexity. The point clouds were acquired by Riegl-VMX 250 MLS ten months before the terrestrial images were taken. Point clouds were geo-referenced with ground control points and classified in a semi-automatic way with Terrasolid software. The point density on the objects ranged from 200 points/m<sup>2</sup> to 900 points/m<sup>2</sup> depending on distance of the object to the MLS, with an average of 500 points /m<sup>2</sup> on objects that are less than 20 meters away from the street border.

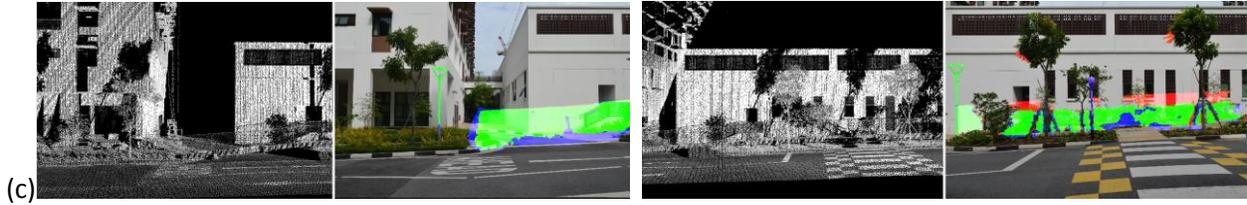
The terrestrial Images were taken by the off-the-shelf camera Fujifilm FinePix F500EXR for the first dataset and Nikon D7000 for the second and third dataset. Images were taken along the pedestrian way as one strip, with the distance ranging from 10 to 30 meters to the targeting objects, and the average in-strip overlap is about 70-80%, with 0.2 cm to 1 cm pixel footprint. The geo-referencing was done by Apero software ([Deseilligny and Clery, 2011](#)) by manually selecting control points on the MLS point clouds.

The three experiments were performed under different street environment, which contains typical urban facilities such as light poles, billboard, carport, parking railings, building facades, and underbrush, etc. These objects lie on different layers in the image depth direction, which constitute a medium to high scene complexity. The same set of parameters is adopted in the three experiments, and the results are demonstrated in Fig. 7 - 9, respectively.

The first dataset contains 27 images covering a 200 m × 50 m area of the street. Objects such as billboards and parking railing were removed during the time period, and new planting areas were built. Examples are shown in Fig. 7 where the most significant changes are detected by the proposed algorithm; false detections are shown on the upper part of the telegraph pole, which result from a significant co-registration error as well as the connection to the real changes on the diamond-shaped billboard in the middle of the telegraph pole.

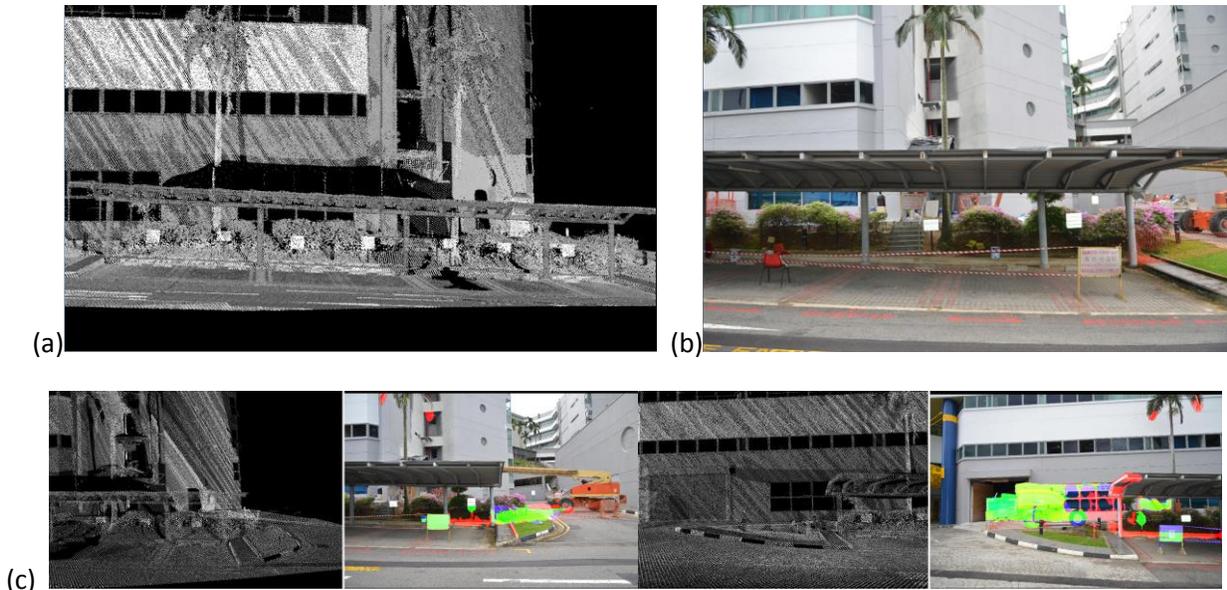
The second dataset contains 18 images which cover a 150 m × 50 m area of the street (Fig. 8). The point clouds showed that the area was under construction and a large amount of fence was found along the street (shown in the red ellipse in Fig. 8(a)) with no street light poles available yet, while the terrestrial image shows the completion of the construction, where the fences are taken out and the plantation areas are built, as well as the light poles, and this experiment shows the case of monitoring a construction site. All the newly built street light poles were correctly detected with only one missing, which is mainly attributed to the disturbance of trees very close to it. Dense fences in the lower part of the image are also correctly detected with some missing because of the sparse points on the fences.





**Fig. 8.** Experiment on the second dataset. (a) Point clouds acquired at an earlier time; (b) one of the terrestrial images; (c) examples of the detection result (green indicates the TP (true positive) pixels, red indicates FP (false positive) pixels, blue indicates FN (false negative) pixels).

The third dataset contains 15 images covering a small area in front of a large building which has experienced a fire accident, and warning signs appear to indicate repair works; this shows the application of recognizing changes in disaster management. Shifts of several sign boards were detected; new warning boards were detected, and the construction vehicles were also detected (in the right pair of Fig. 9(c)). This dataset contains many thin structures which creates more disturbances to the 2D triangulation than the other dataset. The construction vehicle in the left pair of Fig. 9(c) was not detected because of the lack of point clouds in the corresponding area.



**Fig. 9** Experiment on the third dataset. (a) Point clouds acquired at an earlier time; (b) terrestrial images acquired at a later epoch; (c) examples of the detection result (green indicates the TP pixels, red indicates FP pixels, blue indicates FN pixels).

## 5. EVALUATION

The purpose of the view dependent change detection is to: 1) inform the operator where change occurs over a large amount of images which are taken along the street by massive image data-acquisition systems such as image-based MMS and 2) detect geometric changes at a finer level of detail for the purpose of automatic data updating. Therefore, to evaluate the proposed method, we manually interpreted the changes by comparing the point cloud and the images, and masked them onto each view, taking into account the most important changes (appearing, disappearing and deformation) of the

public facilities and man-made objects such as billboards, road signs, bus stops, street light poles, etc. We verify the performance of the proposed algorithm with the following measures: 1) good detection rate (GDR): if a detected segment takes 50% of the ground truth segment, we count it as a good detection; 2) False detection rate (FDR): if a detected segment appears in an area where there is no change, it is counted as a false detection; a higher false detection rate will increase the workload of the operator; and 3) pixel-wise evaluation of true positives (TP), false positives (FP), and the corresponding kappa coefficient (KC), which are computed as follows:

$$TP = \frac{NTP}{PG}, \quad FP = \frac{NFP}{PD}, \quad KC = \frac{(NTP+NTN)/N-M}{1-M}, \quad M = (PD * PG + ND * NG)/(N * N) \quad (12)$$

where  $NTP, NFP, NTN$  stand for the number of true positive pixels, false-positive pixels and true negative pixels, respectively.  $PD, PG, ND, NG$  stand for the number of positives detected, number of positives in ground truth mask, number of negatives detected, and number of negatives in ground truth.  $N$  stands for the total number of pixels.

GDR indicates the ability of the algorithm in recognizing the changes, while FDR indicates the disturbances created by the algorithm. TP and FP evaluate the result pixel-wise to show the completeness of the detected changes as well as their shapes and sizes and the quantity of false alarms. The KC shows in total how the detected result agrees pixel-wise with the ground truth. Fig. 8(c) and Fig. 9(c) show examples of the TP, FP and FN (false negative) pixels of the experiments, and the GDR, FDR, TP, FP and the KC are computed in Table 1:

**Table 1.** Change detection results

Dataset	GDR (%)	FDR (%)	TP (%)	FP (%)	KC
1	80.0	14.2	82.5	5.8	0.86
2	64.0	27.7	80.2	11.9	0.79
3	76.4	26.0	82.6	38.2	0.67
Total	73.5	22.6	81.6	18.6	0.78

To be more detailed, Table 2 shows the change detection results of different object types:

**Table 2.** Change detection results of each object type

Dataset 1	Parking Railing	Boards	Light pole	House segment (approx.)	Hedge	Total
NS	9	4	4	40	4	61
NGT	8	4	2	0	1	15
NCDC	6	3	2	0	1	12
NFDC	0	0	0	1	1	2
Dataset 2	Fence	Hedge	Light pole	House segment (approx.)	Total	
NS	23	4	4	18	49	

NGT	21	0	4	0	25		
NCDC	12	0	4	0	16		
NFDC	0	2	0	4	6		
<b>Dataset 3</b>	Fire hydrant	Boards	Warning tape	Crane	Chair & Bucket	House segment (approx.)	Total
NS	3	10	3	1	5	30	52
NGT	0	8	3	1	5	0	17
NCDC	0	6	2	1	4	0	13
NFDC	1	1	0	0	0	4	6

(**NS**: number of segments in the scene; **NGT**: number of ground true segments that has changed; **NCDC**: number of correctly detected changes; **NFDC**: number of falsely detected changes)

The results of the three experiments show that most of the changes are detected, and it is particularly effective for changes of planar objects such as boards. High GDR are also obtained on objects that have high depth differences such as light poles and parking railing to those at the background (walls, continuous bushes). Missing detections mainly come from occlusions by trees or vehicles, and lack of points for thin objects. Wrongly detected changes appear on the house segments, which are mainly due to the disturbances of the trees. Since the urban scene is very complex, similar works in the literature consider GDR but not the shape of the mask, while it is important for applications such as automatic 3D model updating. We compute the TP and FP pixel-wise to indicate the agreement between the detected change mask and the ground truth mask: three experiments show high TP which means that the changed mask and the ground truth mask agree well in the changed area; experiments on dataset 1 and 2 have high KC which show the shape of the mask also agrees well. In dataset 3, the false positive is higher, which is due mainly to occlusions and thin objects resulting from the scene complexity.

Moreover, we have checked the influence of the registration error to the change detection accuracy, which is also an important factor that affects detection. The registration error mainly causes high a value of the raw changes, and these raw changes are strongly connected to their neighborhoods that have the same color, depth and class labels. Therefore, the robustness of the algorithm to the registration errors is dependent on the object size. If large registration errors (e.g. 10 pixels) occur to a large unchanged object with homogenous textures (e.g. a house with monochromatic color on the wall) which causes a high value of the raw change, the graph cuts will absorb these raw changes with the larger unchanged part. While small objects such as chairs and light poles, are more sensitive to registration errors. We have found that during our experiments, the algorithm is robust to 3 pixels of registration errors for almost all the objects in our tests, but when the registration errors are more than 5 pixels, there will be detection errors on thin objects such as light poles and fences.

## 6. DISCUSSION

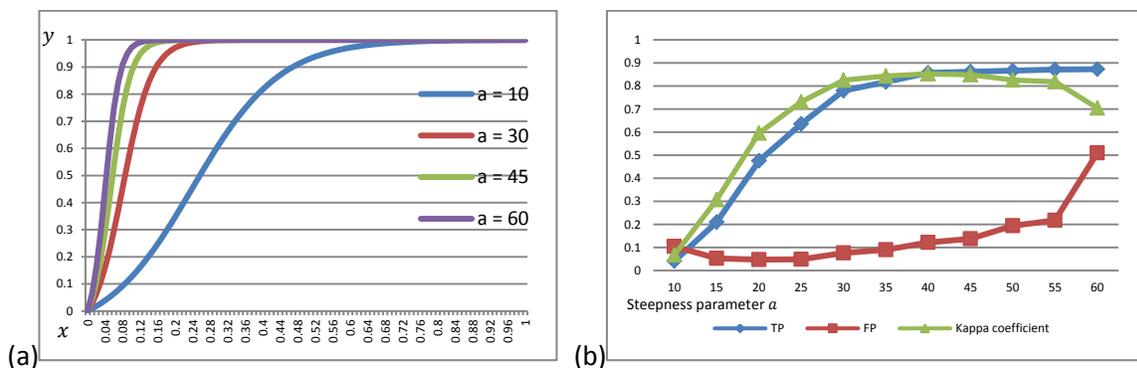
It is worth noting that there are several tunable parameters in the proposed algorithm, and these parameters are described in Table 3.

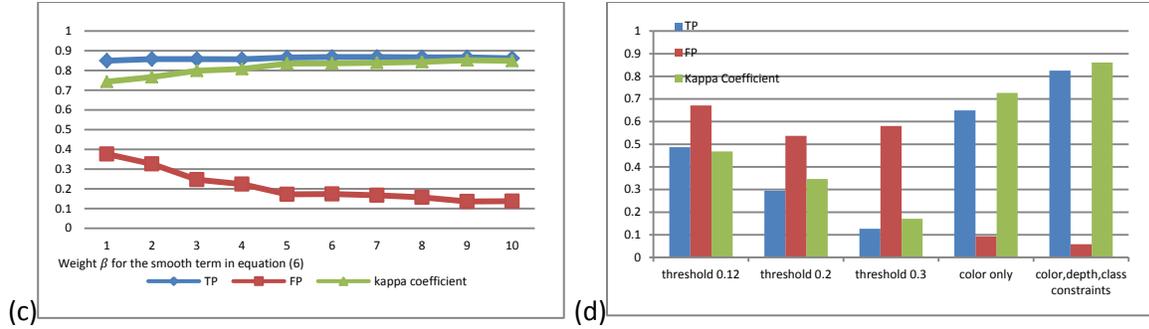
**Table 3.** List of the tunable parameters of the proposed algorithm

Parameter	Function
Window size $w_1$ for WWZB	It decimates back-face points, and depends on the point density

Depth difference threshold and triangle edge threshold	They filter irregular 2D triangles for depth interpolation, the depth threshold is fixed at 0.5m. The edge threshold is 30 pixels, and it varies according to the image resolution
Curve response parameter $a$ and $b$ in (7)	They scales the color difference; $b$ fixed at 11, and $a$ varies for adjusting the sensitivity between changes and color differences.
$M/S$ and $\alpha$ in (11)	$M, S$ are compactness parameter of superpixel; $M/S$ fixed at 20 for RGB images, and more information can be found in ( <a href="#">Achanta et al., 2012</a> ); $\alpha$ is fixed as a large number (400)
Smooth parameter $\beta$ in (6)	It controls the connectivity of superpixels in the graph cuts optimization.
Cell size of the superpixel	It controls the average size of the superpixels, the smaller, and usually depends on the image resolution.

The most important ones are the steepness parameter  $a$  for computing the initial change cost in equation (7), and the weight  $\beta$  of the smooth term in equation (6). These parameters are chosen based on the empirical studies in the experiment. Fig. 10 shows the relationship between these two parameters and the final TP, FP and KC in the first experiment. Fig. 10(a) shows that the shape of the “S” curve adjusts the sensitivity response between the color difference and the initial cost with different  $a$ . Fig. 10(b) shows  $a$  greatly affects the final detection rate, with a small  $a$ , the curve assigns large initial cost of being changed to real changes, which results in decrease of TP, while with a large  $a$ , pixels with small luminance difference are assigned with very low cost to the real changes, which leads increase of FP. The similarity weight  $\beta$  determines how strong the connectivity between pixels are; in Fig. 10(c), it shows small  $\beta$  will result high FP but is relatively stable with TP. By incorporating multi-constraints, the similarity measurement of pixels can be better defined. Fig. 10(d) shows the comparative results obtained by 1) a simple threshold on raw changes (three thresholds 0.12, 0.2 and 0.3 are used, and the raw change is scaled to  $[0, 1]$ ); 2) a constraint of only the color and 3) constraints of color, depth, and class. Poor TP and FP are obtained by simply giving a threshold to the raw changes; the results are further optimized by constraints of color, depth and class. A final KC of 0.78 is obtained for all the three experiments.





**Fig. 10.** (a) The color value mapping curves with different steepness value in equation (7),  $y = \exp(ax - b) / \exp(ax + b)$ ,  $b = 11$ ; (b) the relationship between the parameter  $a$  in equation (7) and the TP, FP and KC; (c) the relationship between the similarity weight  $\beta$  in equation (6) and the TP, FP and KC; (d) TP, FP, KC of simple threshold on raw changes compared to TP, FP, KC of proposed method with color, depth and class constraints.

The proposed method assumes terrestrial images are taken early in the morning, when there are much fewer disturbances from pedestrians and cars. The elimination of disturbances is always the key issue in achieving a good detection rate, and this is largely dependent on image understanding techniques such as pedestrian detection, car detection, and vegetation detection. In the proposed method, we focus on the algorithm of detecting changes in the environment with complex geometry, while addressing the problems of disturbances by simple weighing of the known classes. The elimination of cars is done by assigning a small weight on the road area projected on the image, as well as trees. For pedestrian detection, we propose to adopt a HOG-LBP (Histograms of Oriented Gradients - Local Binary Pattern) feature based human detecting algorithm ([Wang et al., 2009](#)) to mask out possible pedestrian rectangular areas.

There are two steps that require human interaction in our procedure: point clouds classification and the registration of terrestrial images to the point clouds. It is possible to avoid the classification step, but it will result in decrease of the change detection accuracy (Fig. 10(d)). The point clouds classification of the whole scene is a one-time effort, and class information only needs to be updated for changed part after the change detection is performed, so for periodical monitoring and updating, it is worth to perform the classification in the very beginning. As automatic classification methods are increasingly proposed with decreasing false alarms ([Arastounia, 2012](#)), it is also possible to automate our method by designing more robust strategy to handle the false alarms of the classification. Moreover, The current process of registering point clouds to the terrestrial images relies on the manually selected control points, and this process can be possibly automated by robust 3D/2D feature extraction and matching techniques.

## 7. CONCLUSIONS

We have proposed a new method for change detection at street level by using MLS point clouds and terrestrial image in a 3D complex urban environment. The idea is to compare the differences between the 3D point clouds and the implicit scene geometry to effectively avoid illumination differences as well as matching uncertainty, and also to propagate the probability of changes in the image space with multi-

constraints by graph cut optimization. In general, our contribution is four-fold: 1) the proposed method performs the change detection on data from different sources, and it effectively addresses the problem of purely image-based change detection (problems with illumination change, perspective distortion, unreliable geometry) and purely point cloud-based change detection (high cost, high volume of storage compared to the photogrammetric images for our change detection purpose). 2) The proposed method combines the explicit 3D geometry of point clouds and the implicit geometry of images to overcome the uncertainty brought about by automatic image matching. 3) The proposed method establishes the changed area in finer detail than given in the literature by extending the changes in the image space with multi-constraints which creates the potential for further automatic updating. 4) This paper performs experiments with real datasets with acceptable detection rates and this proves the feasibility of the proposed method.

Experimental results show the applicability of the proposed methods in terms of visual inspection and the statistical analysis with the manually marked ground truth. This algorithm is primarily considered to be a selector and change marker which selects a number of images that have high probability of containing geometric changes out of thousands of images, as well as the changed mask. The proposed algorithm makes use of MLS data and photogrammetric techniques as means of monitoring city infrastructure and progress on construction sites. It could be easily extended to change detection of 3D models which is simpler, because the z-buffer problem in the point cloud is internally solved, since polyhedral models are represented as piece-wise continuous surfaces. The research demonstrates the potential and feasibility of such applications and proposes a way to make good use of MLS data together with terrestrial images for performing the change detection at the street view level in complex urban environments. The propagation of the changes by color, depth and class information shows that the resulting changed mask fits naturally to the objects' boundaries.

The proposed method is able to detect most of the significant changes while suppressing artifacts and false alarms. In the experiment, most of the false positives were mainly from the following circumstances: 1) incomplete point clouds which violates the assumption that points on objects are distributed evenly; 2) branches of trees that grow or are moved by wind which are inconsistent with the point cloud in absolute geometry and 3) large homogenous areas showing only a very small amount of color inconsistency on the border and these changes are absorbed by the homogeneous areas with very low initial cost. Future improvements of the algorithm will focus on image understanding, acquiring more semantic information of the scene, e.g. cars and trees, to better tackle the aforementioned problems of change-detection in a very complex urban environment. Moreover, as the proposed method requires some manual operation, our future work will also look for automatic solution for change detection at the street level.

## **ACKNOWLEDGEMENTS**

This work was established at the Singapore-ETH Centre for Global Environmental Sustainability (SEC), co-funded by the Singapore National Research Foundation (NRF) and ETH Zurich. The authors would like to thank the anonymous reviewers for their precious comments and suggestions.

## REFERENCE

- Achanta, R., A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Susstrunk, 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (11), 2274 - 2282.
- Arastounia, M., 2012. Automatic Classification of LiDAR Point Clouds in A Railway Environment. University of Twente, Netherlands, 83.
- Bouziani, M., K. Goïta and D.-C. He, 2010. Automatic change detection of buildings in urban environment from very high spatial resolution images using existing geodatabase and prior knowledge. *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (1), 143-153.
- Boykov, Y. and V. Kolmogorov, 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (9), 1124-1137.
- Broadhurst, A., T. W. Drummond and R. Cipolla, 2001. A probabilistic framework for space carving. In: *Proceedings of IEEE International Conference on Computer Vision*, Vancouver, British Columbia, Canada, 7-14 July, pp. 388-393.
- Chaabouni-Chouayakh, H. and P. Reinartz, 2011. Towards automatic 3D change detection inside urban areas by combining height and shape information. *Photogrammetrie-Fernerkundung-Geoinformation* 2011 (4), 205-217.
- Champion, N., D. Boldo, M. Pierrot-Deseilligny and G. Stamon, 2010. 2D building change detection from high resolution satellite imagery: A two-step hierarchical method based on 3D invariant primitives. *Pattern Recognition Letters* 31 (10), 1138-1147.
- Cornelis, N., B. Leibe, K. Cornelis and L. Van Gool, 2008. 3d urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision* 78 (2-3), 121-141.
- Davies, E. R., 2004. *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann, 934 p.
- Deseilligny, M. P. and I. Clery, 2011. Apero, an open source bundle adjustment software for automatic calibration and orientation of set of images. In: *Proceedings of ISPRS International Workshop on 3D Virtual Reconstruction and Visualization of Complex Architectures*, Trento, Italy, 2-4 March, pp. 269-276.
- Doxani, G., K. Karantzalos and M. Tsakiri-Strati, 2010. Automatic change detection in urban areas under a scale-space, object-oriented classification framework. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII (Part4/C7)*, 7 p. (on CDROM).
- Eden, I. and D. B. Cooper, 2008. Using 3D line segments for robust and efficient change detection from multiple noisy images. In: *10th European Conference on Computer Vision*, Marseille, France, 12-18, October, pp. 172-185.
- Girardeau-Montaut, D., M. Rouxa, R. Marcob and G. Thibaultb, 2005. Change detection on point cloud data acquired with a ground Laser scanner. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36 (Part 3/W19), pp. 30-35.
- González-Aguilera, D., P. Rodríguez-González and J. Gómez-Lahoz, 2009. An automatic procedure for co-registration of terrestrial laser scanners and digital cameras. *ISPRS Journal of Photogrammetry and Remote Sensing* 64 (3), 308-316.
- Gruen, A., 2012. Development and status of image matching in photogrammetry. *The Photogrammetric Record* 27 (137), 36-57.
- Haala, N. and M. Kada, 2010. An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (6), 570-580.

Hussain, M., D. Chen, A. Cheng, H. Wei and D. Stanley, 2013. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing* 80 (2013), 91-106.

Kang, Z. and Z. Lu, 2011. The change detection of building models using epochs of terrestrial point clouds. In: *IEEE International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping* Xiamen, China, 10-12 January, pp. 1-6.

Knudsen, T. and B. P. Olsen, 2003. Automated change detection for updates of digital map databases. *Photogrammetric engineering and remote sensing* 69 (11), 1289-1296.

LTA, 2014. Maintaining Our Roads and Facilities, <http://www.lta.gov.sg/content/ltaweb/en/roads-and-motoring/road-safety-and-regulations/maintaining-our-roads-and-facilities.html>. (Accessed 14 January, 2014)

Makadia, A., A. Patterson and K. Daniilidis, 2006. Fully automatic registration of 3D point clouds. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 17-22 June, pp. 1297-1304.

Matikainen, L., J. Hyyppä, E. Ahokas, L. Markelin and H. Kaartinen, 2010. Automatic detection of buildings and changes in buildings for updating of maps. *Remote Sensing* 2 (5), 1217-1248.

Murakami, H., K. Nakagawa, H. Hasegawa, T. Shibata and E. Iwanami, 1999. Change detection of buildings using an airborne laser scanner. *ISPRS Journal of Photogrammetry and Remote Sensing* 54 (2), 148-152.

Niederöst, M., 2003. Detection and reconstruction of buildings for automated map updating. *Mitteilungen- Institut für Geodäsie und Photogrammetrie an der Eidgenössischen Technischen Hochschule Zurich*

Peng Gong, Greg.S. Biging and R. Standiford, 2000. Technical Note: Use of Digital Surface Model for Hardwood Rangeland Monitoring. *Journal of Range Management* 53 (6), 622-626.

Pollard, T. and J. L. Mundy, 2007. Change detection in a 3-d world. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, 18-23 June, pp. 1-6.

POSLV, 2013. POSLV-Specifications, [http://www.applanix.com/media/downloads/products/specs/poslv\\_specifications12032012.pdf](http://www.applanix.com/media/downloads/products/specs/poslv_specifications12032012.pdf). (Accessed 07 November, 2013)

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

Rottensteiner, F., 2008. Automated updating of building data bases from digital surface models and multi-spectral images: Potential and limitations. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 37 (Part B3a), pp. 265-270.

Soheilian, B., N. Paparoditis and D. Boldo, 2010. 3D road marking reconstruction from street-level calibrated stereo pairs. *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (4), 347-359.

Taneja, A., L. Ballan and M. Pollefeys, 2011. Image based detection of geometric changes in urban environments. In: *Proceedings of IEEE International Conference on Computer Vision*, Barcelona, Spain, 6-13 November, pp. 2336-2343.

Terrasolid, 2013. Terrasolid - Software for Airborne and Mobile LiDAR and Image Processing, <http://www.terrasolid.fi/>. (Accessed 11 September, 2013)

Vicente, S., V. Kolmogorov and C. Rother, 2008. Graph cut based image segmentation with connectivity priors. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 24-26 June, pp. 1-8.

Walter, V., 2004. Object-based classification of remote sensing data for change detection. *ISPRS Journal of Photogrammetry and Remote Sensing* 58 (3), 225-238.

- Wang, X., T. X. Han and S. Yan, 2009. An HOG-LBP human detector with partial occlusion handling. In: Proceedings of IEEE International Conference on Computer Vision, Miami, Florida, USA, 20-25 June, pp. 32-39.
- Waser, L., E. Baltsavias, K. Ecker, H. Eisenbeiss, E. Feldmeyer-Christe, C. Ginzler, M. Kuchler and L. Zhang, 2008. Assessing changes of forest area and shrub encroachment in a mire ecosystem using digital surface models and CIR aerial images. *Remote Sensing of Environment* 112 (5), 1956-1968.
- Xiao, J., T. Fang, P. Zhao, M. Lhuillier and L. Quan, 2009. Image-based street-side city modeling. *ACM Transactions on Graphics (TOG)* 28 (5), 114.
- Zhang, K. and P. Main, 1990. Histogram matching as a new density modification technique for phase refinement and extension of protein molecules. *Acta Crystallographica Section A: Foundations of Crystallography* 46 (1), 41-46.
- Zhou, L., 2013. Fusing laser point cloud and visual image at data level using a new reconstruction algorithm. In: *Intelligent Vehicles Symposium (IV)*, 2013 IEEE, pp. 1356-1361.