# LiDAR-guided dense matching for detecting changes and updating of buildings in Airborne LiDAR data

Zhou, K.; Lindenbergh, R.; Gorte, Ben; Zlatanova, S.

**DOI**
10.1016/j.isprsjprs.2020.02.005

**Publication date**
2020

**Document Version**
Accepted author manuscript

**Published in**
ISPRS Journal of Photogrammetry and Remote Sensing

# LiDAR-guided dense matching of airborne VHR images for detecting changes in and updating of buildings in LiDAR data

Kaixuan Zhou[a,*], Roderik Lindenbergh[a], Ben Gorte[b], Sisi Zlatanova[b]

[a] *Department of Geoscience and Romote Sensing, Delft University of Technology, Stevinweg 1, Delft, the Netherlands*
[b] *Faculty of Built Environment, University of New South Wales, Kensington NSW 2033, Australia*

**Abstract**

As more complete 3D information can be extracted from a stereo pair, using stereo images can provide better changes detection from LiDAR data than using shadows in a single image. However, the quality problem of 3D information in both LiDAR data and stereo images will result in false alarms and directly affect the accurate change detection. LiDAR point clouds are sparse and irregularly spaced, and have mixed returns near building edges, while 3D information extracted from stereo images are affected by shadow and low texture. This chapter proposes LiDAR-guided edge-aware dense matching (LEAD-Matching) to address these problems explicitly for detecting accurate building changes. Data sparsity and irregular spacing is addressed by densifying LiDAR points in a form of a digital surface model (DSM). Instead of applying interpolation with associated edge problems due to mixed returns, three candidate DSMs are created by linking each DSM pixel to up to three planes as identified in segmented and triangulated LiDAR data. The candidate DSMs limit the disparity search space for dense matching, addressing low texture and shadow problems in images. Through edge-aware dense matching, the detailed building edges in stereo pairs determine the optimal heights to address LiDAR edge problem. Changes are detected where corresponding pixels from dense matching have large color differences. Due to homogeneous surroundings and shadows, only partial changes are initially detected. A second hierarchical dense matching step is employed to complete changes and update 3D information by propagating initial partial changes iteratively. The proposed method is applied on data from two cities, Amersfoort and Assen, the Netherlands, with around 1200 existing buildings. In both areas, the method successfully verifies unchanged buildings while detecting minimum changes of $2 \times 2 \times 2\ m^3$. The experiments also show that the proposed method outperforms two well-known change detection methods in terms of verifying unchanged buildings and detecting small changes simultaneously.

*Keywords:* Change detection, up-to-date 3D city model, building, LiDAR data, VHR images,

*Corresponding author
*Email address:* k.zhou-1@tudelft.nl (Kaixuan Zhou )

## 1. Introduction

Up-to-date 3D city models are needed for many applications, e.g. water management, city climate assessment and urban planning. Airborne LiDAR data is widely used for constructing 3D city models, however, acquiring LiDAR data is expensive so the updating rate is low at national level. For example, open source nationwide point clouds are available for the whole of the Netherlands, but a complete update takes around 7-10 years [1]. Airborne Very High Resolution (VHR) images are often available every year due to their lower cost. Thanks to the development of dense image matching [2, 3], there is an opportunity to use point clouds from stereo images—photogrammetric point clouds [4] for creating up-to-date 3D city models. However, point clouds extracted from stereo images are often outlying and incomplete in shadow and low texture areas [5]. As urban areas do not change dramatically every year, stereo images are actually suitable for change detection and updating changed areas. In past decades, spectral information has been widely used for deriving changes [6, 7, 8]. However, especially for building change detection, the accuracy and effectiveness of using 3D geometric information is considered higher [9, 10]. Therefore, a method for detecting changes and updating buildings in LiDAR data using 3D information from stereo images is proposed. Our goal is to preserve LiDAR data in unchanged areas and update it with photogrammetric point clouds in changed areas. The updated point clouds can be used by any 3D reconstruction method or software to create up-to-date 3D models.

3D information extracted from both data sources have pros and cons. Airborne LiDAR data has high vertical accuracy, so accurate plane information extracted from LiDAR data has been used for decades for 3D building modelling. However, airborne LiDAR data is often sparse and irregularly spaced. If LiDAR point cloud is densified by interpolating to a DSM with uniform spatial spacing for change detection, the interpolated heights are not always reliable, especially near building edges due to mixed returns [11]. For example, LiDAR points from ground, wall and roof near an overhanging roof get mixed from top view. Wrong heights directly result in false alarms in change detection. On the other hand, VHR stereo images with higher ground sampling distance (GSD) have detailed building edges and can extract dense point clouds. However, 3D points extracted from stereo images are missing or outlying in shadow and low texture areas, also resulting in false alarms in change detection.

These false alarms resulting from the quality problems so far prevented the development of an accurate 3D change detection and updating workflow that could meet requirements for a large scale 3D map, which often requires buildings with small size, for example $2 \times 2$ m$^2$, to be included. Several state-of-the-art methods [10, 9, 12] only detect building changes over 100 m$^2$, 200 m$^2$ and 50 m$^2$

respectively.

In this chapter, we propose LiDAR-guided edge-aware dense matching (LEAD-Matching) on an stereo pair to address these quality problems. The method starts from densifying LiDAR data by employing the plane information extracted from sparse LiDAR points, instead of using interpolation. As buildings often consist of planar surfaces, an accurate DSM height can be calculated, as long as a correct plane is assigned to each DSM pixel. LiDAR point clouds are first segmented and then triangulated in 2D with plane information assigned to each vertex. Each DSM pixel receives up-to three different planes from the three vertices of the 2D triangle the pixel locates in. For example, a DSM ground pixel near a building may fall in a triangle with three vertices namely ground, wall or roof, where in the most cases the correct plane is included. Three DSMs are created and then transformed to disparity images to guide the dense matching by limiting the disparity search space (DSS) to address the shadow and low texture problems in dense matching. The edge-aware dense matching method uses detailed building information from images to determine the optimal heights from three candidates to address the building edge problems in the LiDAR data. Note that, no priori classification of LiDAR data to ground, wall and roof classes is required. If corresponding pixels chosen by dense matching still have large color differences, these pixels indicate that the 3D information in the LiDAR data is outdated. However, changes are not detected completely in homogeneous areas. A second hierarchical dense matching step is applied to propagate partial changes iteratively and update 3D information simultaneously.

The scientific contributions are as follows. (1) To our knowledge, we are the first to propose LEAD-Matching method to detect accurate changes by integrating accurate plane information from LiDAR data and detailed building edges from a stereo pair to address quality problems in both data sources. (2) A novel two-step dense matching framework is proposed for accurate 3D building change detection and updating. (3) The proposed method is shown to obtain a successful building verification rate, while detecting unprecedented minimum changes of $2 \times 2 \times 2$ m$^3$ for updating large scale 3D maps.

## 2. Related work

Several articles [9, 10, 13] confirmed that detecting building changes from geometric information is more accurate than from spectral information. [14, 10, 9] discussed the fusion of geometric and spectral information for change detection. However, tuning parameters for fusion on different datasets is difficult. Therefore, this chapter only discusses three possible approaches for deriving geometric change between LiDAR data and images.

### 2.1. Single image change detection

Shadow is an indicator of the 3D geometry of a scene in a single image. By using geometric information to reconstructed shadows in an image, changes are detected when the reconstructed shadows do not match the actual shadows in the acquired image. Shadow information has been used for detecting buildings in VHR images in [15]. However, shadows are rarely discussed in change detection. To reconstruct shadows, a watertight model like a DSM or a surface mesh needs to be constructed from the LiDAR data. However, an accurate model is difficult to obtain automatically due to quality problem of LiDAR data. More importantly, many lower buildings positioned next to a high building may not cast shadows, so only partial geometric change can be derived.

### 2.2. Direct geometric change detection

Detecting changes directly between two point clouds seems straightforward. Easiest is to subtract two DSMs interpolated from point clouds to detect changes. However, false alarms near building edges occur in both data sources described above. False alarms can be bigger than those resulting from the misalignment of the two sources. In [16], a window-based approach, considering minimum height differences between two DSMs converted from photogrammetric point clouds, is applied to filter false alarms near building edges. But choosing an effective window size is critical, otherwise small building changes would be discarded. A similar approach [17] uses morphological filters to reduce building edge effects of DSM errors. Instead of converting point clouds to DSMs, 3D Euclidean distances are used to reduce false alarms near building edge by searching for corresponding points between two point clouds [13]. The critical step in this approach is to find correct corresponding points and estimate their 3D distances. Iterative closest point (ICP) [18] is often used to register point clouds. Instead of using distances of nearest points as the change metric, the Hausdorff distance of local neighbors of nearest points is tested to be a better metric to deal with roughness and varying point density in point clouds. A surface based distance metric is provided by the multi-scale model to model cloud comparison (M3C2) algorithm [19] that further mitigates the effects of roughness and point density variation in point clouds. A least squares 3D surface matching algorithm (LS3D) [20] registers two point clouds by minimizing distances from local surfaces instead of points. An outlier removal procedure is conducted in each minimization iteration when the distances between local surfaces are above a threshold. These outliers are considered as changes. However, none of the above methods addresses quality problems of photogrammetric point clouds explicitly. For example, in the LS3D algorithm, the outliers extracted from shadow and low texture regions will cause false alarms as they will be considered as change.

### *2.3. Projection-based geometric change detection*

Instead of extracting point clouds from stereo images, the projection-based geometric change approach projects existing 3D information from LiDAR data to stereo images to find corresponding pixels. The color similarity of the corresponding pixels is used for validating the 3D information [13]. The advantage of this approach is that no correspondences need to be identified from stereo images, especially in shadow and low texture areas. However, finding accurate corresponding pixels from stereo images for each sparse LiDAR point is not easy. As terrain relief causes relief displacement and occlusions in airborne images [21], several LiDAR points can get projected to the same image pixel. A DSM interpolated from sparse LiDAR points is often required to determine which LiDAR point can be seen from the camera. Given a DSM and known intrinsic and extrinsic camera parameters, true ortho-rectification is applied to rectify displaced pixels to their correct place. If the given height is correct, corresponding pixels from different ortho-photos should have similar color. However, the quality of the ortho-photos relies on the quality of the DSM used in the ortho-rectification process. Instead of making true ortho-photos, dense matching is applied to compensate for misalignment of data sources in [9], to better find the correct corresponding pixels from a stereo pair. The DSM converted from an existing and accurate 3D model is transformed to a disparity image which constrains the disparity search space (DSS) by only allowing a small search range to guide semi-global matching [3] for compensation of misalignment between two data sources. However, the method strongly relies on the accuracy of the 3D model [13]. If a DSM is obtained by interpolation from LiDAR data, the edge problem of the DSM will directly affect the quality of change detection. The method described in [22] uses planar surfaces from sparse points extracted from a feature detector to provide guidance for dense matching. Every image pixel receives not only one, but several candidate disparities, estimated from candidate planes the pixel may belong to, limit the DSS for improving the quality of dense matching in problematic areas.

## 3. Methodology

Our LEAD-Matching densifies height information to create candidate DSMs from planar surfaces extracted from sparse LiDAR points. Three candidate heights estimated from three adjacent planes are generated for each DSM pixel. The candidate DSMs are transformed to candidate disparities to guide dense matching. This approach avoids searching for correspondences from stereo images, especially in shadow and low texture areas, but also uses image information to determine heights for LiDAR data in uncertain areas, especially near edges. By addressing these quality problems, or data limitations, in both data sources explicitly, accurate partial changes are detected. A second dense matching step is performed to complete changes and update 3D information simultaneously. The workflow uses a

two-step dense matching approach as shown in Figure 1, followed by a post-processing step to filter irrelevant changes caused by trees, cars and other small objects.

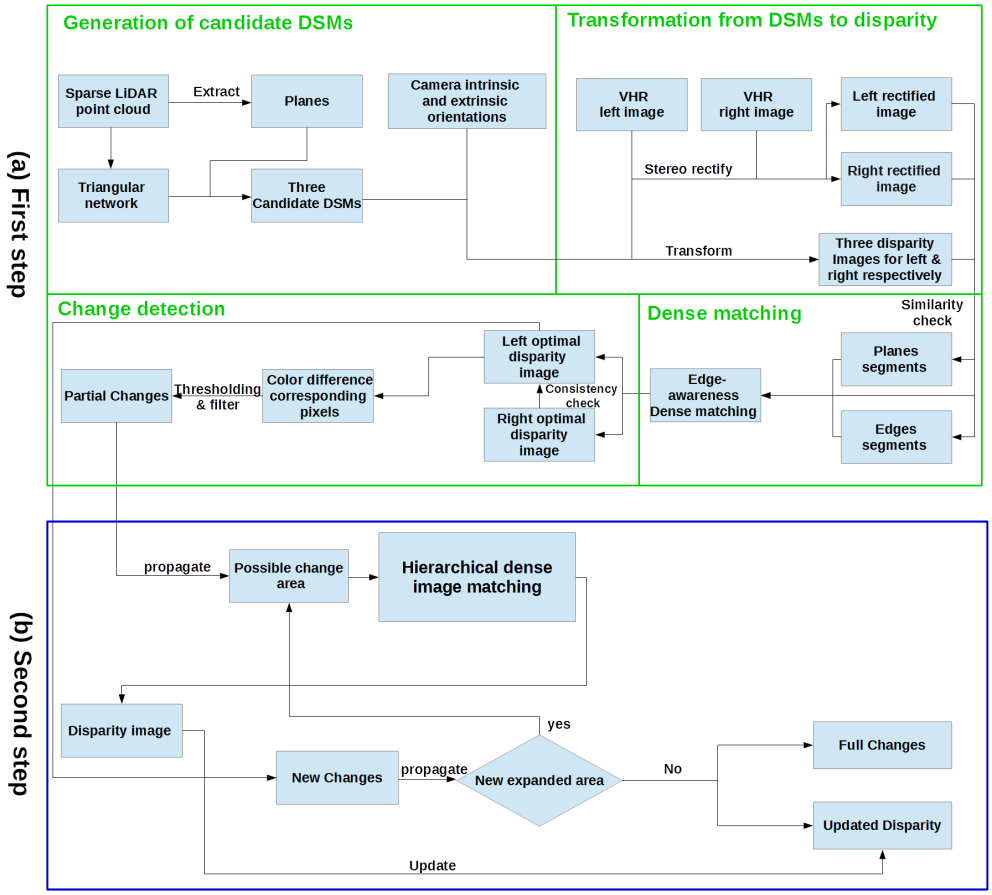


Figure 1: The workflow of the two-step dense matching algorithm to detect changes and update of LiDAR data using a stereo pair.

### *3.1. Partial change detection—LiDAR-guided dense matching*

As shown in Figure 1a, the proposed LiDAR guided edge-aware dense matching step consists of four parts as follows.

#### *3.1.1. Generation of candidate DSMs*

As described in pseudo-code in Algorithm 1, the planar patches are first segmented from sparse LiDAR points from a successful plane growing algorithm proposed in [23]. The planar information is associated to the sparse LiDAR points. In this step, we aim to find the adjacent planes for each DSM pixel by searching the neighboring LiDAR points. However, the sparsity and mixed return problems of

LiDAR data make it sometimes difficult to ensure that the correct plane is present among the adjacent planes, especially in the edge areas. To enforce that the correct plane is almost always present, the two problems, sparsity and mixed return, are elaborated and addressed as follows.

**Algorithm 1** Generation of Candidate DSMs from LiDAR Data

**INPUT:** *Pcs* - LiDAR point cloud;

**OUTPUT:** *vDs* - Vector of three DSMs;

1: **procedure** DSMGeneration(*Pcs*)
2: *Pls* ← *RegionGrow (Pcs)* ▷ Use Region Growing for plane segmentation
3: *Tris* ← *Delaunay (Pcs)* ▷ Obtain 2D Delaunay triangulation of LiDAR
4: *Plim* ← *Interppl (Tris,Pls)* ▷ Assign three planes for each DSM pixel using Delaunay triangles
5: *Edgesegments* ← *Segmentation (Plim)* ▷ Use plane id in each pixel to get edge segments
6: *Edgesegments* ← *Fillholes (Edgesegments)* ▷ Fill holes in edge segments
7: **for** each pixel *p* in *Edgesegments* **do**
8: *Plim* ← *Update(Plim,window,p)* ▷ Update planes for edge pixel
9: *vDs* ← *CreateDSMs (Plim)* ▷ Create DSMs from three planes in each pixel

Due to the sparsity problem, the adjacent points selected for the DSM pixel on the ground may all lay on the roof, when the several nearest LiDAR points in 2D are selected as shown in Figure 2b. A 2D Delaunay triangulation addresses this sparsity problem by defining adjacent LiDAR points for each pixel as the three vertices of the 2D Delaunay triangle in which the pixel is located. Therefore, the triangle allows a maximum of three different planar segments found for each pixel, which often provides enough candidate planes for DSM pixels near edges. Indeed, most DSM pixels near building edges are adjacent to at most three planes, corresponding to roof, wall and ground planes respectively. Note that classification of the planes is not necessary. The probability of a pixel belonging to a candidate plane is estimated according to the distance of the corresponding point of the plane to the pixel. These steps correspond to lines 2-4 in Algorithm 1.

Due to the mixed return problem of LiDAR points, it is still possible that a DSM pixel is not linked to its correct plane as shown in Figure 2c and d. Two DSM pixels on the roof shown in Figure 2c and d, do not have the roof plane as a candidate. In Figure 2c, the DSM pixel has adjacent vertices of two colors, purple and red, corresponding to wall and ground segments respectively. As this situation only happens near edges, edge areas are identified before addressing this problem. If a DSM pixel is located in a triangle with all three vertices assigned to the same plane, the pixel is labeled as planar, otherwise it is labeled as edge pixel. However, the DSM pixel in Figure 2d is located near an edge but

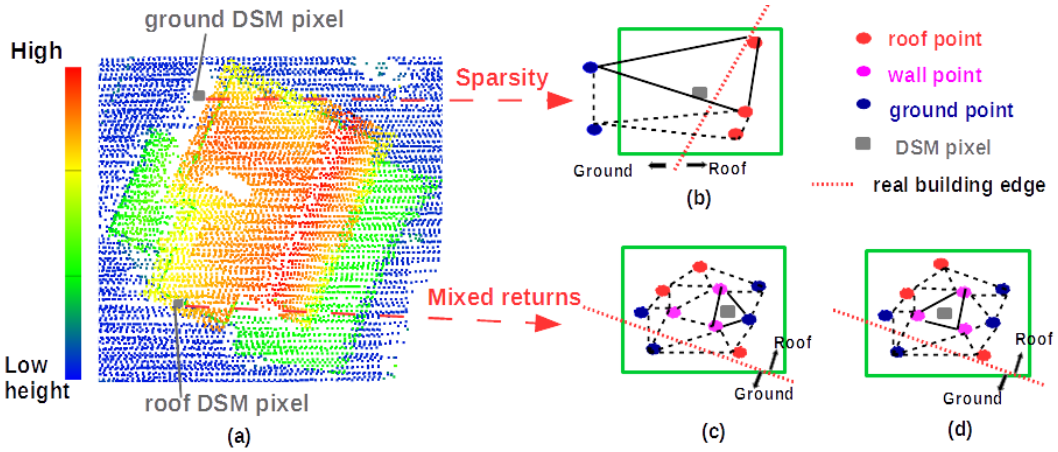


Figure 2: (a) Sparse LiDAR point clouds with mixed points, shown in different colors, near building edge areas. (b) Sparse LiDAR points around a DSM pixel. The black triangle defines the points adjacent to the pixel. (c) The DSM pixel, on the roof, in a triangle does not get a height from the roof plane as a candidate. The vertices of the triangle only include wall and ground points. (d) The DSM pixel, on the roof, in a triangle only get heights from wall planes as candidates.

still marked as a planar pixel with all three pixels assigned to the wall plane. This effect results in many holes near edges. The size of such holes is often less than the average GSD of the LiDAR points. Typically, LiDAR data comes with a specification of its point density as X points per square meters (ppm). The GSD is approximated by $\frac{1}{\sqrt{X}}$. A closing filter of the GSD size is applied to fill such holes. In these edge areas, a window with the same size is applied to every edge pixel to get robust candidate planes from neighboring pixels. The probabilities of the pixel belonging to each candidate plane are used to choose and order the top three plane candidates. Finally, three heights for each DSM pixel are estimated from the three candidate planes. Therefore, three candidate DSMs are derived. Planar pixels are simply assigned three times the same height while edge pixels have a maximum of three different heights. These steps correspond to lines 5-9 in Algorithm 1.

### *3.1.2. Transformation from DSM to disparity*

In this part, the candidate DSMs created from LiDAR are transformed to candidate disparities to guide dense matching by limiting the disparity search space for a stereo pair, which consists of a left and right image, represented as $I_l$ and $I_r$ respectively. First, the perspective projection, using camera intrinsic and extrinsic parameters of a stereo pair, is applied to project each DSM pixel, in a form of a 3D point, to left and right image respectively to find the corresponding pixels. As described above, several DSM pixels can be projected to the same image pixel. A depth image of the DSM from the optical center of the left and right image respectively is created to select the correct DSM pixel, i.e.

the pixel that is actually seen by the image. Second, stereo rectification using camera intrinsic and extrinsic parameters [24] is applied to rectify the stereo pair in order to align the corresponding pixels to the same rows. Therefore, the positions of the corresponding pixels are simplified to a single value, called disparity, which is obtained by simply subtracting the column indices of the corresponding pixels in the rectified images. Third, as adjacent DSM pixels may be projected to the same image pixel or to image pixels that are not adjacent, resulting in salt and pepper effects, a 3 by 3 median filter is applied to clean the disparity values. Many facade pixels in images don't get disparity values as DSMs do not contain accurate facade heights. They are set to a very small negative constant to indicate occlusions. Finally, all three candidate DSMs are transformed using the three steps above to candidate disparity images for $I_l$ and $I_r$ respectively. This means that both left and right images are applied with dense matching, as the consistency check is applied latter to detect occlusions.

The guidance of LiDAR data is set by using the three disparities to limit the disparity search space (DSS) for each image pixel. A small range of $[-T, +T]$ is set to each disparities for adding flexibility to compensate for possible misalignment between data sources. The pixels in the rectified image are easily classified as edge or plane pixels by simply checking whether three disparities are the same. The image is then split into planar, edge and occlusion segments by connected component labeling. The DSS consists of $2 * T + 1$ candidate disparities for each planar pixel, and consists of $3 * (2 * T) + 1 = 6 * T + 3$ candidate disparities for each edge pixel. Each segment, except occlusion segments, is used for dense matching independently. As each segment is used for dense matching, the less computation and storage is needed.

Even though each disparity image has been cleaned by a median filter, still segments from candidate DSMs are not transformed exactly to disparity images. The segments in the boxes in Figure 3b are small and isolated and should be cleaned with the corresponding disparities updated. In the Figure, white, gray and black pixels represent planar, edge and occlusion pixels. Several steps are applied to remove small and isolated edge segments, and fill small holes as shown in the blue boxes in Figure 3b. A opening filter with a size of 3 is applied to remove isolated pixels. Also edge segments with area below $1m^2$ are removed. Another closing filter of a bigger size, 7, is applied to fill holes in the edge segments to better fill holes in the edges while reducing expansion effects. The same process is applied to remove small and isolated occlusion segments as shown in the red boxes in Figure 3b. The result is shown in Figure 3b bottom. The candidate disparity of pixels is updated according to the class changes introduced by the cleaning steps. For new planar pixels, a corresponding $3 \times 3$ window is applied to select the top one disparity as the candidate. For new edge pixels, a corresponding $7 \times 7$ window is applied to select the top three disparities as candidates.

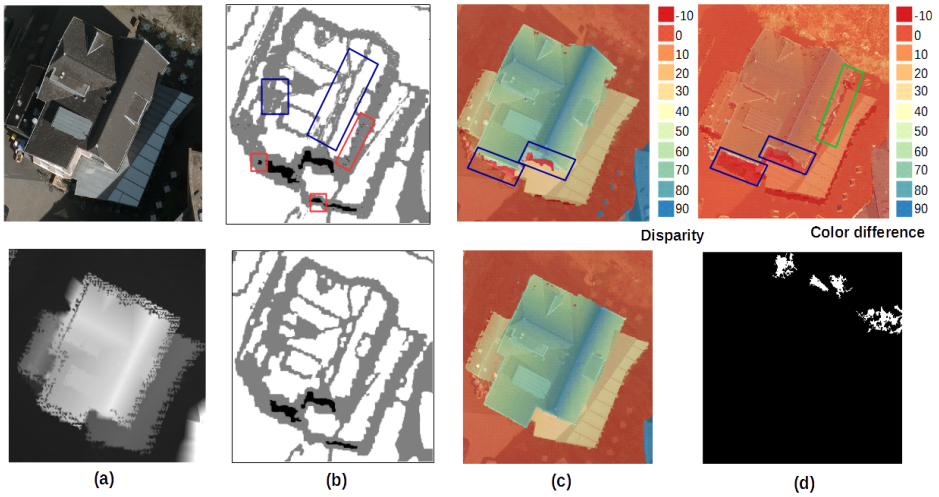


Figure 3: LiDAR-guided dense matching applied to a building without changes between LiDAR and image acquisition. (a) The building, shown in the image (top) and LiDAR data (bottom), did not change. (b) Top: many small isolated segments obtained after initial segmentation in blue and red boxes. Bottom: the segments after filtering small isolated segments for dense matching. (c) Optimized disparity for left image (top) and right image (bottom). (d) Top: the color difference of corresponding pixels from dense matching with occlusions indicated. Some changes, like speckle, are left and indicated in the green box. Bottom: changes detected after thresholding and filtering. Indeed, no change is detected in the building area.

### *3.1.3. Dense matching*

Dense matching is often defined as a maximum a posterior Markov Random Field (MAP-MRF) problem, which combines with global optimization to estimate optimal disparity for each pixel [22]. The MRF formulation aims to estimate optimal disparity by considering two aspects: (1) color similarity of corresponding image pixels, and (2) disparity similarity among neighbors. The second aspect improves the estimation especially in low texture or shadow areas. An equivalent framework is energy minimization as used in [3]. A MRF is defined by an underlying graph $G = (V, E)$ where each vertex $v \in V$ denotes a pixel in the disparity image, and each undirected edge $e \in E$ connects the pixel to one of its neighbors $W \in V$. The disparity of pixels is given by $\mathbf{d} = \{d_i\}_{i \in V}$, while disparity probability depends on the color information from two rectified images, represented by $\mathbf{I} = \{I_l, I_r\}$. $c_{p_i}$ and $c_{p'_i}$ store the color information of corresponding pixels from the two images given $d_i$. A MRF assumes that $d_i$ obeys the Markov property which states that probability is also conditional on its neighbors $W_i$ [25]. The posterior probability over $\mathbf{d}$, given two rectified images $\mathbf{I}$ is given as:

$$P(\mathbf{d} \mid \mathbf{I}) = \frac{1}{Z}\Big(\prod_{i \in V} \varphi(c_{p_i}, c_{p'_i}, d_i) \cdot \prod_{i \in V} \prod_{j \in W_i} \psi(d_i, d_j)\Big), \tag{1}$$

where Z is a normalization constant known as partition function, independent from disparity. $\varphi$ and $\psi$ are association (AP) and interaction potentials (IP), respectively. The two terms are defined as follows.

The association potential is a local term which encodes the link between a given disparity value and color similarity of the resulting corresponding pixels. The association potential , $\varphi$, is modeled as an exponential function [26] of the posterior probability for $d_i$ given $c_{p_i}$ and $c_{p'_i}$ :

$$\varphi(c_{p_i}, c_{p'_i}, d_i) = \exp(P(d_i|c_{p_i}, c_{p'_i})). \tag{2}$$

The probability of disparity given color information of the corresponding pixels is defined as the normalized cross-correlation (NCC) score in $[-1, 1]$ between two $3 \times 3$ matrices constructed from the intensity of $3 \times 3$ image patches centered at each corresponding pixel. As negative correlation does not give more confidence on the difference between two image patches, the NCC is restricted to $[0, 1]$ using a max function, $max(0, \mathrm{NCC}(c_{p_i}, c_{p'_i}))$. Shadows are often casted near building edges. Textures in shadows are seriously affected, so the NCC probability is not reliable. Another way of LiDAR guidance is applied when pixel intensity is low. We then define:

$$P(d_i|c_{p_i}, c_{p'_i}) = \begin{cases} \max(0, \mathrm{NCC}(c_{p_i}, c_{p'_i})) + \epsilon \cdot \omega(\overline{c}_{p_i}), & \text{if } \mathrm{Ord}(d_i) = 1 \\ \max(0, \mathrm{NCC}(c_{p_i}, c_{p'_i})), & \text{otherwise.} \end{cases} \tag{3}$$

In case the first ranked disparity is considered, as indicated by the condition $\mathrm{Ord}(d_i) = 1$, the term $\epsilon \cdot \omega(\overline{c}_{p_i})$ adds guidance from the LiDAR data. As described above, the probabilities of the candidate heights given to each DSM pixel are estimated. When the heights are transformed to disparities, disparity values are ranked in descending order based on these probabilities. The candidate disparity ranked first is the one with highest probability according to the LiDAR data. $\epsilon$ is a constant to add guidance from the LiDAR data when a pixel is affected by shadow. The logistic function $\omega(\overline{c}_{p_i})$ defines the probability of a pixel affected by shadow and depends on the average intensity of the $3 \times 3$ image patches $\overline{c}_{p_i}$ from which the disparity is extracted:

$$\omega(\overline{c}_{p_i}) = \frac{1}{1 + a(\exp^{\frac{\overline{c}_{p_i} - b}{\theta}})}, \tag{4}$$

where $a, b$ and $\theta$ are parameters defining the shifting and decay speed of the possibility function. The logistic function is defined to let the LiDAR guidance decrease when the intensity increases. In this way, the LiDAR information is incorporated in designing the association potential, especially for improving dense matching in shadow areas..

The interaction potential is a term preferring similar disparity in neighborhoods. The interaction potential, $\psi$, is modeled as an exponential function [26] of the joint probability of $(d_i, d_j)$:

$$\psi_{i,j}(d_i, d_j) = \exp(P(d_i, d_j|e_{ij})). \tag{5}$$

The interaction potential prefers disparities of neighboring pixels to be similar. However, in edge areas, disparities are more likely to be different. Therefore, an edge-aware IP is designed conditionally on edge features $e_{ij}$ from colors differences of neighboring pixels. The probability $P(d_i, d_j|e_{ij})$ conditional on $e_{ij}$ is defined as:

$$P(d_i, d_j|e_{ij}) = \begin{cases} 1 & \text{if } d_i = d_j \\ \max(0.7, \omega(\Delta g_{ij})) & \text{if } |d_i - d_j| = 1 \\ 1 - \omega(\Delta g_{ij}) & \text{otherwise.} \end{cases} \quad (6)$$

where $\Delta g_{ij}$ is an edge feature representing the absolute difference of the gradient of pixel i and its neighbor j. $\omega(\Delta g_{ij})$ is a logistic function similar to Eg.(4). Therefore, $1 - \omega(\Delta g_{ij})$ defines that if $\Delta g_{ij}$ is high when an edge probably exists between pixel i and j, the probability will be high when the difference of disparity between pixel i and j is larger than 1. The AP and IP are set to the equal weights. The optimization algorithm, alpha expansion [27], is selected due to its ability of global optimization with a good efficiency.

*3.1.4. Partial change detection*

After dense matching, an optimal disparity is selected for each pixel as shown in Figure 3c. The example indicates that LiDAR-guided dense matching provides effective results, especially near building edges. However, many pixels in one image, especially facade pixels, are occluded in the other image as shown in the blue box in Figure 3c. These disparity images are applied for consistency checking [3]. Each pixel $p$ in the left image uses its disparity to find its corresponding pixel $p^{'}$ in the right image. If the disparity difference between $p$ and $p^{'}$ is larger than 2 pixels, $p$ is labeled as occlusion pixel, as shown in blue boxes in Figure 3d. Changes are detected by subtracting the color difference between $p$ and $p'$, which is defined as the root mean square of the differences in RGB value of corresponding pixels. A threshold is set to get a binary change image.

Then, an important step is to remove irrelevant changes or speckle as shown in the green boxes in Figure 3d and Figure 5b and c. Our goal is to find all height changes of at least 2 m. The relief displacement in pixel unit corresponding to 2 m of height above the ground defines the size of the filter. If camera is nearly nadir, a quick estimation of the displacement can be estimated from Figure 4a assuming the stereo pairs are acquired in stereo-rectified position ('normal case') without considering the stereo-rectification. An accurate estimation of the displacement of 2 m height above the ground is performed later to confirm this assumption. If the average ground height is known, the linkage between ground disparity $d_g$ and average ground height $h_g$ is a function of the baseline $B$ between two images, focal length $f$ in the unit of pixels and flight height $H$ [28]: $H - h_g = \frac{B \cdot f}{d_g}$. The relation

between relief displacement $\Delta d$ and height with respect to ground $\Delta h$ is:

$$\Delta d = B \cdot f\left(\frac{1}{H - h_g - \Delta h} - \frac{1}{H - h_g}\right). \tag{7}$$

An example from our research data provides the graph of this relation in Figure 4b which shows that 2 m of height change from the ground corresponds to a displacement of around 14.2 pixels approximately. Accurate displacements using the same data are estimated by considering the stereo-rectifications as shown in Figure 4c. The figure shows the displacement of all pixels with a height of 2 m above the average ground level in an area of 512 m by 240 m. All displacements are around 14.5 pixels which only differs 0.3 pixel from the quick estimation. When the size of displacement is estimated, connected component labelling is applied to group components and an erosion filter with a size slightly smaller than displacement size, e.g. 10 pixels, is then applied to remove small components. After this filtering, the unchanged building is verified effectively as no changes are detected in the building area as shown in Figure 3d.

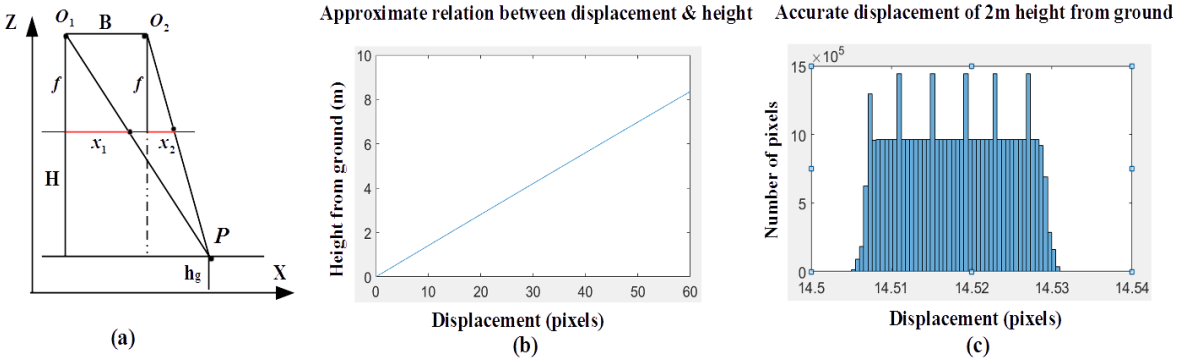


Figure 4: (a) Geometric explanation of the relation between disparity and height in 'normal case'. (b) Relation between the approximate displacement from the function and height above the ground. (c) The accurate displacement calculated for 2 m heights above the ground.

However, in homogeneous areas, corresponding pixels found using LiDAR data would still have similar colors. In Figure 5a and b, when many small buildings with homogeneous roof colors are newly built,the detected changes only occur near building boundaries. The same problem happens for shadow areas. In Figure 5c and d, when a new building stands in the shadow. only partial change is detected. More difficultly, if a building would have been completely removed and replaced by homogeneous ground, even no partial change would be found by our method. This problem is solved by using the ground disparity in buildings surrounding to find corresponding pixels for the pixels in building areas indicated by LiDAR data. If all these corresponding pixels are similar, all pixels in building areas are marked as change.

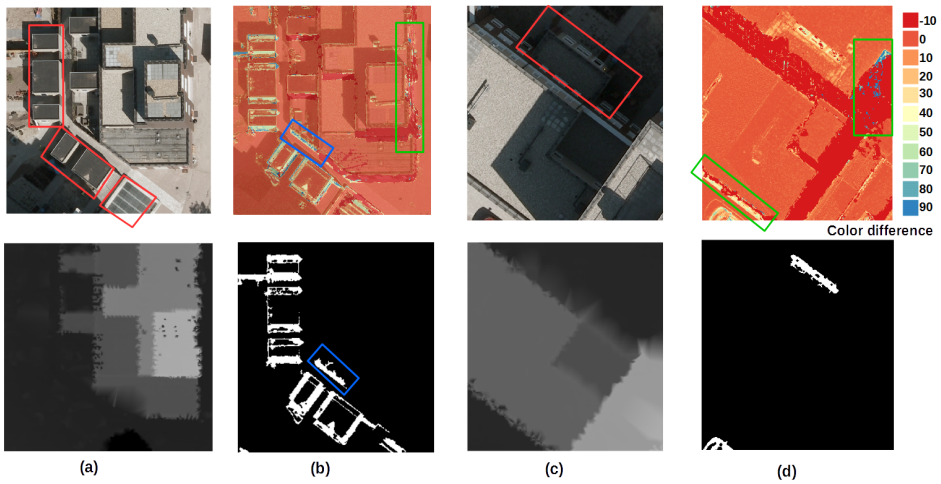


Figure 5: (a) LiDAR-guided dense matching applied to two examples where buildings were built between LiDAR and image acquisition. (b) The color difference (top) and binary partial changes (bottom) shows the effect of homogeneous roofs. Examples of speckle and irrelevant changes from a fence are shown in the green and blue box respectively. (c) One building is newly built between the old LiDAR data (bottom) and new image (top) acquisition. (d) The color difference (top) and binary partial changes (bottom) show the effects of shadow.

### *3.2. Change propagation and update*

#### *3.2.1. Hierarchical dense matching*

The second step of dense matching is to iteratively propagate the partial changes. The disparity image estimated in the first LiDAR-guided dense matching step represents the state of the geometric information in the LiDAR data. The second step aims to estimate disparities, from images only, in areas propagated from where partial changes were detected. If these disparities estimated from images are different from the disparities from the first step, changes and updated 3D heights are derived simultaneously. However, the disparity search space (DSS) is large without guidance from LiDAR data, requiring a large computation and storage. A hierarchical dense matching method borrowed from the SURE algorithm [28] is implemented to estimate disparities to overcome the computation and storage issues. The same MAP-MRF framework as above is applied with the difference that in this case no information from LiDAR data is used. The same alpha expansion algorithm is used for optimization.

#### *3.2.2. Iterative change propagation*

First, hierarchical dense matching is applied in areas surrounding partial changes. A dilation filter of 1 m is applied on the partial change areas. The reason of choosing this small size of expansion is that many changes detected in first step of dense matching are walls and fences as shown in the blue

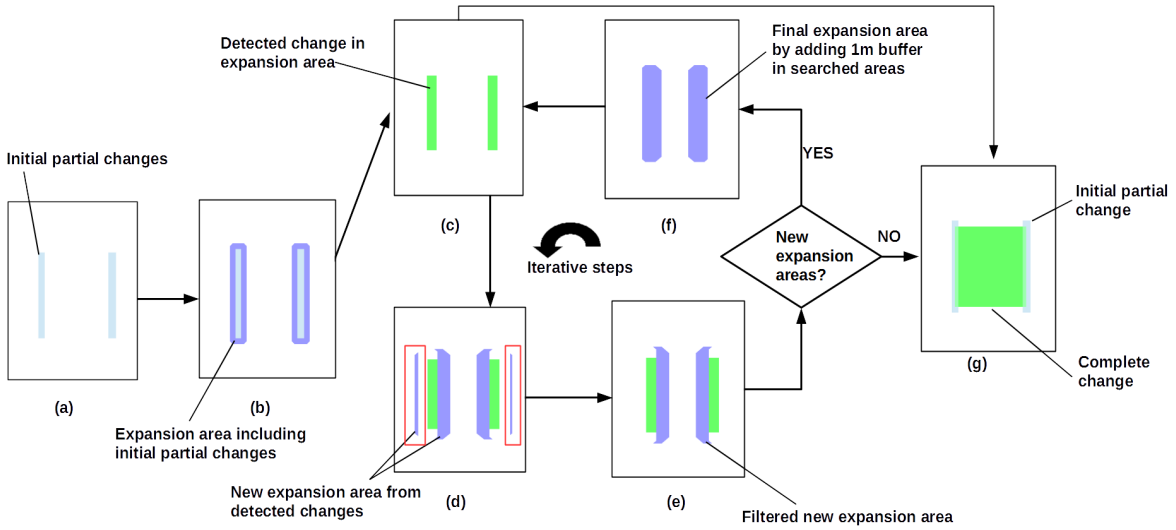


Figure 6: Iterative change propagation work flow. (a) Partial changes identified in the first dense matching step. (b) Areas propagated from partial changes. (c) Detected new changes in propagation areas shown in green. (d) Areas propagated from new changes. (e) Removed propagation areas not adjacent to the new changes. (f) Final propagation area for further disparity estimation including the 1 m search areas. (g) Full change propagated from partial changes.

box in Figure 5b. Propagation at a larger size would introduce unnecessary areas for processing. A connected component labeling is applied to the dilated changes to create segments, which are applied with dense matching independently. In Figures 6a,b, the propagation is shown for two segments identified in the partial change step, similar to the changes shown in Figure 5b (bottom).

When disparities for the left image are estimated, corresponding pixels in the right image are queried. The disparities from the right image are used to verify the disparities extracted from the left image through a similar consistency check as described in Section 4.3.1–Partial change detection. After the check, the disparities of pixels in expansion areas are compared to the disparities resulted from the first step of dense matching to detect 2 m height changes. If the difference is larger than the filter size, e.g. 10, as described in Section 4.3.1–Partial change detection, pixels are marked as changed as shown in Figure 6c. The irrelevant change and outliers of with a size below $1 \times 1$ m$^2$ is removed.

A larger propagation area of $2 \times 2$ m$^2$ is applied to create new expansion areas from the detected changes in order to increase the speed of propagation in the iterative steps as shown in Figure 6. The new expansion areas are subtracted from the searched areas, which are the expansion areas shown in Figure 6b. The remaining new expansion area is shown in Figure 6d. If the expansion areas is not adjacent to the detected changes as shown in the red boxes in Figure 6d, these areas are removed

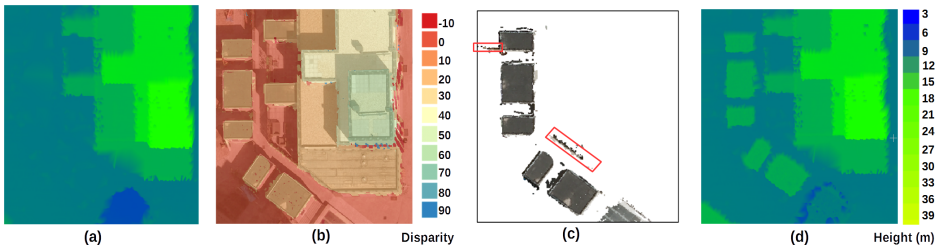


Figure 7: Transformation of updated disparity and changes in images to point cloud data. (a) 3D mesh of old LiDAR data. (b) Updated disparities for the new buildings overlaid on the VHR image. (c) 3D points extracted from the updated disparities. A wall connected to a newly built building and a separate fence are detected as new objects shown in a red and blue box respectively and are filtered as being too small. (d) 3D mesh of the updated LiDAR data using 3D points extracted from images after filtering.

as shown in Figure 6e. Dense matching relies on a smoothness constraint to estimate disparities in shadows and low texture areas. In order to get disparities with smooth transition following estimated disparities in the previous step, the final expansion areas as shown in Figure 6f also include 1 m areas in the searched areas. The same hierarchical dense matching is applied to derive new changes iteratively until no more propagated areas are added as shown in Figure 6e. Using this approach, complete changed areas are detected as shown in Figure 6g , while simultaneously, corresponding disparities are extracted that are used for updating height information.

### *3.3. Post-processing*

As changes and updates are only obtained in the image domain, these changes and updates should be transformed to the changes and 3D updates on the LiDAR point cloud.. Changes are separated into new objects and (partially) removed objects by evaluating the differences between updated disparity and disparity guided by the LiDAR data. If updated disparity is higher, a newly built object is identified. Otherwise, a removed object is identified. The updated disparity, as shown in Figure 7b, is transformed to a 3D point cloud as shown in Figure 7c. Then irrelevant changes, such as trees, cars and small walls, should be filtered to obtain final building changes. Therefore, the point cloud is rasterized to a DSM and cleaned by the following three filtering steps: (1) filter vegetation using NDVI (normalized difference vegetation index). NDVI is calculated from additional near-infrared images with coarse spatial resolution. If the NDVI of a pixel is above 0.1, the pixel is marked as vegetation in a vegetation mask. This mask is used for filtering new objects including many trees and bushes detected as newly built objects. (2) Filter changed objects on roads and rivers. The focus of this chapter is to detect building changes, therefore, changes on roads and in rivers are disregarded. Many cars detected as new objects are parked on the side of the road. An existing vector topographic map with information of streets and rivers is used for this purpose. (3) Walls, fences, as shown in

the blue and red boxes in Figure 7c, and remaining cars, are filtered based on their size. A filter component step is applied to filter change components with a size below $2 \times 2$ m$^2$. Finally, the LiDAR points in the remaining change areas are replaced by newly extracted 3D points from images. The original and updated 3D mesh are shown in Figures 7a and d, respectively.

### *3.4. Evaluation*

Both pixel-wise and object-wise evaluation is performed. Three metrics are selected: completeness ($Comp$), correctness ($Corr$) and F1 score.

$$\begin{aligned} Comp &= \frac{TP}{TP+FN}, \quad Corr = \frac{TP}{TP+FP}, \\ F1 &= 2 \times \frac{Comp \times Corr}{Comp + Corr}. \end{aligned} \tag{8}$$

TP, FP and FN denote the true positives, false positives and false negatives respectively. The completeness and correctness of change detection is denoted by $Comp$ and $Corr$ respectively [29]. The F1 score is a quality measurement combining these two metrics. In pixel-based evaluation, $TP$ and $FP$ are calculated based on the intersection of building pixels between detection result and ground truth. In object-based evaluation, the intersection between segmented building objects is calculated. An object is defined to be correctly detected if the detected object has at least 50% area overlap with the ground truth as suggested by [29]. TP and FP also denote the number of buildings correctly detected and falsely detected respectively. The ground truth of building changes is represented by GT.

## 4. Experiment and discussion

### *4.1. Data specification and pre-processing*

#### *4.1.1. LiDAR data and BAG building map*

The study area consists of two urban areas located in Amersfoort and Assen, the Netherlands, covering 512 m by 240 m and 854 m by 386 m respectively. LiDAR data is from AHN2, an open source dataset for the Netherlands which was collected between 2007 and 2012 [1]. The point clouds acquired for Amersfoort and Assen in 2010 and 2012 are shown in Figure 8a and Figure 11a respectively. The average horiontal and vertical accuracy of the LiDAR data are around 5 cm and an average of 10 points per square meter (ppm) is available [30]. The approximate spatial resolution is around 30 cm. BAG is an open-source Dutch building map. Trees always change between LiDAR and image data acquisition. Tree effects in the LiDAR data are reduced using BAG building polygons, while trees effects on the images are reduced using NDVI. Note that, effectiveness of NDVI depends on the image acquisition season. In general, NDVI is also affected by shadows. In Assen experiment, still, three trees were falsely detected as new buildings .

*4.1.2. Aerial VHR image*

VHR images were taken by the same Microsoft Vexcels UltraCam-Xp in 2010 and 2018 for Amersfoort and Assen respectively. The bundle adjustment is performed using ground control points to provide a file with camera intrinsic and extrinsic parameters. The back-projection error of using these parameters is less than 1 pixel. The data channels are RGB and the image size per frame is 11310×17310 for Amerfoort and 8720×13340 for Assen respectively. The Assen data is pan-sharpened, resulting in a image of $13080 \times 20010$ pixels. The images are taken from near-nadir position and have 60% forward and 40% lateral overlap. For Amersfoort, flight and ground height are around 620m and 5m respectively, and baselines are around 160m and 281m in forward and lateral direction respectively. Combined with the focal length of the camera, 16750 (in unit of pixels), the GSD on the ground is approximately 3.5 cm. For Assen, flight and ground height is around 970 m and 12 m, and the baselines are around 260 m and 700 m in forward and lateral direction respectively. The focal length of the camera is 19326.923 pixels, so the GSD on the ground is approximately 5 cm. Objects of about 2 meter height on the ground have 14 and 11 pixel displacements in two data sources respectively according to Equation 7, which is suitable for our proposed method. The stereo pair with 60% overlap is chosen for both areas for change detection and updating. As most of the buildings in the data are not very high, occlusions are limited in the stereo pair. One VHR image in the overlapping area with its stereo pair is shown for Amersfoort and Assen in Figure 8b and Figure 11b respectively. Another open-source Dutch near-infrared ground ortho-image with a coarse GSD of 25 cm in 2010 and 2018 is obtained for Amerfoort and Assen respectively. The acquisition time is not exactly the same as the VHR images described above, but they perform reasonably good to remove trees.

*4.1.3. Existing and ground truth of updated building point clouds*

In our research, the existing LiDAR data and BAG maps are used to present the status of buildings in the past, while the new VHR images are used to update building point clouds. In the Amersfoort dataset, the LiDAR and image data are obtained in the same year, 2010. There are no significant building changes. Therefore, we simulate changes by using BAG map from 2008. Comparison to the BAG map shows that many buildings are newly built between 2008 and 2010. BAG building polygons are used to clip LiDAR data to create a building point cloud for 2008. In order to create ground truth for validating our method, the changed building polygons are drawn on the LiDAR data in 2010. The minimum size of changed buildings drawn is $2 \times 2 \times 2$ m$^3$. After the building polygons in 2010 are manually delineated, the original LiDAR data from 2010 are clipped for use as ground truth to evaluate the updated point clouds from our method.

In Assen, the LiDAR and image data are obtained in different years, 2012 and 2018 respectively. An up-to-date BAG building map is available for 2018. Manual editing of the 2018 BAG map is

performed to create a 2012 BAG map using the LiDAR data of 2012. Building point clouds for 2012 are created in the same way as for Amersfoort. For Assen, an updated LiDAR point cloud, AHN3, collected between 2014 and 2019 [1], is available. The ground truth is created using 2018 building polygons to clip AHN3 LiDAR data. The ground truth of building changes for Amersfoort and Assen is shown in Figure 8c and Figure 11c.

The LiDAR point clouds are rasterized to DSMs for evaluation by interpolating heights from LiDAR points in each polygon. Due to the limited resolution of LiDAR data, the boundaries are not exactly correct. We removed an area of 20 cm buffer around the boundaries from the ground truth. This step is suggested by the ISPRS benchmark evaluation which removed a buffer area of 24 cm to reduce the impact of uncertain borders on the evaluation [31].. The programs are written in C++ using opencv, gdal and liblas libraries except for two sections: (1) the surface growing algorithm to find planes from LiDAR data from Twente University [23]; (2) the optimization algorithm for the MRF model, alpha expansion, from the UGM [32].

### *4.2. Results*

#### *4.2.1. Amersfoort results*

The change detection and update results are shown in Figure 8d. As shown in Table 1, all 253 unchanged buildings are verified. One removed building has been correctly detected. There are 16 new buildings detected correctly, while 4 new buildings are missed, and 3 objects different from buildings were falsely-detected. The completeness and correctness of the new building detection are 0.80 and 0.84 respectively. The missed buildings are mainly due to shadow and tree effects. For example, in Figure 9a, a small new building is correctly detected. However, half of the building is covered by trees. After filtering trees, the size of remaining changes is less than $2 \times 2$ m$^2$. Another new building is barely visible in the image due to shadows and occlusions casted by trees. The reconstructed building points are not complete and the area is also less than $2 \times 2$ m$^2$. Two false-detections were caused by cars in the image. One example is shown in Figure 9b. A truck with a size large than $2 \times 2$ m$^2$ is identified as a new building. One false-detection is caused by problems in the LiDAR data. As shown in Figure 9c, LiDAR points penetrate through the window on the ground, so the heights from the LiDAR data are lower than the terrain height. The 3D mesh from the LiDAR data shows that the height of the window is lower than the ground. The ground heights are captured by images and an object change is detected as shown in the lower right image. One removed building is correctly detected as shown in Figure 9d. The building is shown in the 3D mesh of the LiDAR data in Figure 9d left, while the building is removed in the updated 3D mesh in Figure 9d right.

The proposed method also manages to detect many small buildings. Three newly built buildings are shown in Figure 9e. The smallest building found is around $2 \times 2 \times 2$ m$^3$. The examples are given

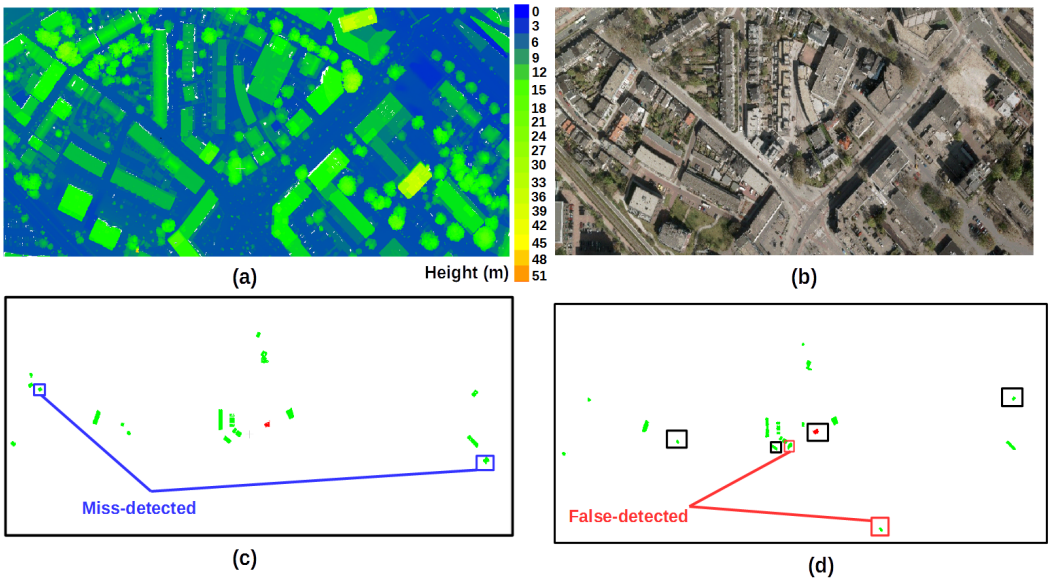


Figure 8: Building change detection results in Amersfoort. (a) AHN2 LiDAR data for Amersfoort. (b) VHR image of the same area. (c) Ground truth for building changes detection. Removed buildings are shown in red and new buildings are shown in green. Missed buildings are indicated by Blue boxes. (d) Changes detected from the proposed method. False-detected buildings are indicated by red boxes. Black boxes indicate examples of correctly detected buildings. The details in the boxes are shown in Figure 9.

Table 1: Change detection result for Amersfoort.

| Pixel-based | *Comp* | *Corr* | F1 | Object-based | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | GT | TP | FP | *Comp* | *Corr* | F1 |
| Building verification | | | | 253 | 253 | | | | |
| New building | 0.85 | 0.86 | 0.86 | 20 | 16 | 3 | 0.80 | 0.84 | 0.82 |
| Removed building | 0.99 | 0.99 | 0.99 | 1 | 1 | 0 | 1 | 1 | 1 |

by the upper two buildings in Figure 9e. The bottom image shows that the method manages to detect a new building even in the shadow. In this case, partial changes were found in the first step of image matching, while change propagation in the second image matching step is effective by estimating disparities only for the area of interest. Pixel-based evaluation in Table 1 shows that the completeness of new building detection is 0.86, while the correctness is similar with a value of 0.86. It indicates that the proposed method manages to keep both completeness and correctness high, resulting in a high F1 score with a value of 0.86. As only one removed building exists in the experiment, this high detection rate with a F1 score of 0.986 is not meaningful. But the effectiveness of the proposed method to detect removed buildings is shown below in the Assen experiment. The height difference between updated

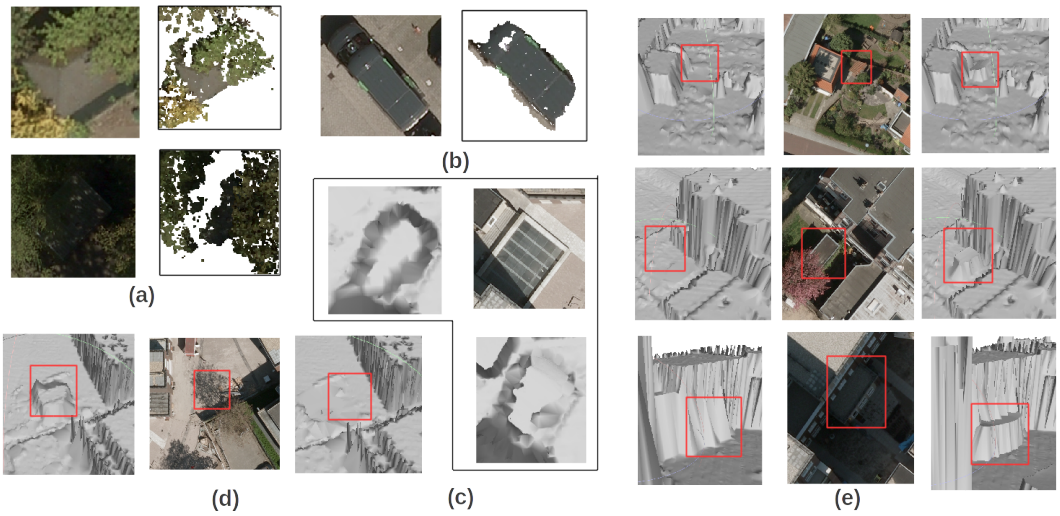


Figure 9: Examples of (a) missed, (b, c) falsely detected, and (d, e) correctly detected building changes. (a) Missed buildings due to trees or shadows casted by trees. (b) a truck wrongly detected as a building. (c) An over-detected object due to a ground window. (d) Correctly detected removed buildings. The original 3D mesh, a new VHR image and a updated 3D mesh are shown from left to right respectively. (e) Three examples of correctly detected buildings that are small or in the shadow. The original 3D mesh, new VHR image and updated 3D mesh for three cases are shown from left to right.

point clouds from the proposed method and ground truth is calculated from the overlapping area. The average height difference in the changed areas is 0.20 m.

*4.2.2. Amersfoort comparative results*

The detection results are compared qualitatively to two types of methods, a projection-based geometric approach (projection–geometry) [9] and robust DSM subtraction (robust–dDSM) [10] as described in Section 4.2.3 and 4.2.2. The results of the three methods on the three cases introduced in Figures 3 and 5 are compared. In Figure 10a, as projection–geometry uses the interpolated LiDAR DSM for guiding dense matching to detect changes, the error in edge areas propagates to the final change detection result as shown in the red boxes. Another problem shown in the blue boxes is on the building facades. As the DSM does not have heights on facades, the corresponding pixels extracted by DSM guidance are wrong. These false alarms with a size similar to the real building changes, as shown in the green boxes, are difficult remove. The proposed method rely on the guidance from three candidate DSMs instead of one interpolated DSM. The false alarms from building edges and facade are largely reduced. In addition, the changes obtained by projection–geometry are not complete due to homogeneous areas.

In order to apply robust–dDSM, a photogrammetric point cloud is constructed from four-view

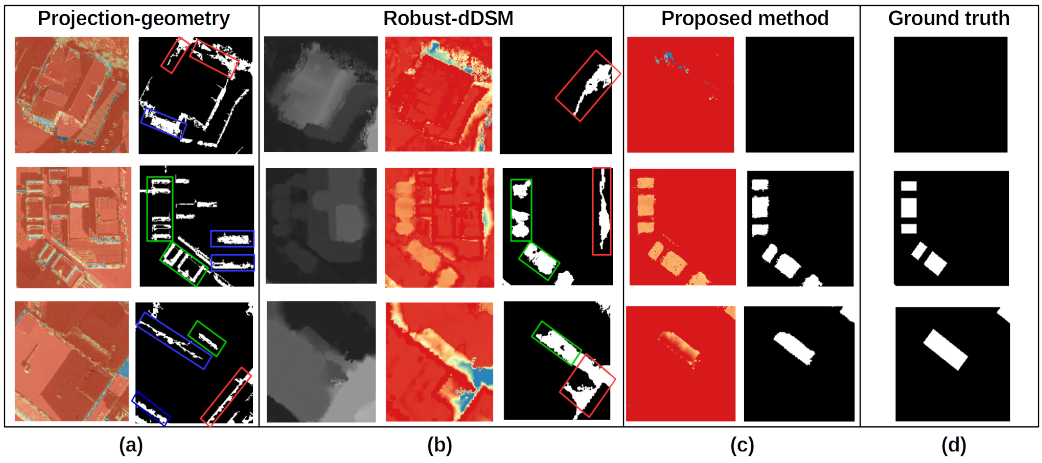


Figure 10: Comparison of results from projection–geometry (a), robust–dDSM (b) and proposed method (c) for three cases introduced in Figures 3 and 5. The ground truth is shown in (d). Top: no change. Middle: multiple small new buildings. Bottom: a new building in the shadow.

images using the software Pix4D. For each pixel in the DSM of the photogrammetric point cloud, a $7 \times 7$ window is applied to select neighborhood pixels in the LiDAR DSM to calculate the minimum difference. If the height difference is more than 2 m, the pixel is labeled as change. Changes are filtered following the filtering steps in Section 3.4. In Figure 10b, the poor quality of the edges from DSMs converted from LiDAR and photogrammetric point clouds is mitigated by robust–dDSM. However, the quality problem of photogrammetric point clouds in low texture and shadow regions results in large areas of false changes as shown in the red boxes in Figure 10b. These false changes are difficult to separate from real changes in the green boxes. The results of the proposed method in Figure 10c shows the proposed LiDAR guidance in dense matching improves 3D information extracted in low quality regions.

#### *4.2.3. Assen results*

The change detection and updating results are shown in Figure 11d. As shown in Table 2, all 952 unchanged buildings are verified. There are 163 buildings correctly detected. while 12 buildings are missed, and 9 objects are falsely-detected. The *Comp* and *Corr* of new building detection is 0.93 and 0.95 respectively. As trees adjacent to buildings in Assen data are less, no buildings were removed due to tree as described for the Amersfoort experiment. In the 12 missed buildings, two were missed in the first step of dense matching due to homogeneous areas and shadows, while the other 10 missed buildings were caused by the failure to extract point clouds in change areas due to low textures and shadows in the second step of dense matching. As shown in Figure 12a, the quality of the 3D points

reconstructed for the new building in the image is poor, due to low texture and repetitive rooftop pattern. From the 9 falsely-detected objects, three objects are high trees and vegetation, which are not removed using NDVI. Another two objects are balcony parasol and pavilion in the park. The pavilion is shown in the upper image of Figure 12b. Two big trucks were also false detected. One of them is shown in the lower image of Figure 12b. Two objects are piles of sands and stones on a construction site as shown in the Figure 13b.

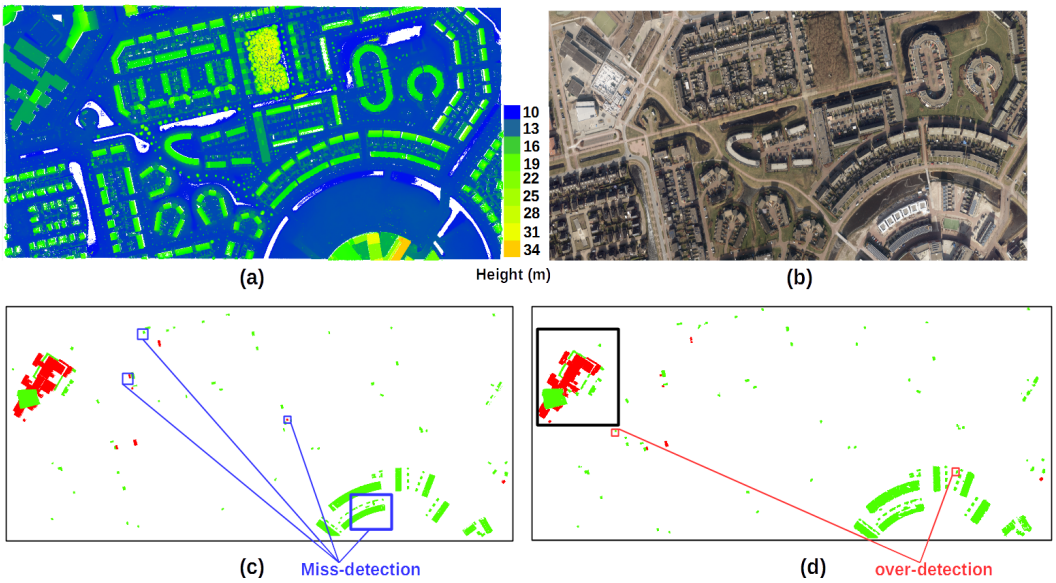


Figure 11: Building change detection results in Assen. (a) AHN2 LiDAR data for Assen. (b) VHR image of the same area of the AHN2 LiDAR data. (c) Ground truth of building changes. Removed buildings are shown in red and new buildings are shown in green. Missed new or removed building are indicated in blue boxes. (d) Changes detected from the proposed method. False-detected buildings are indicated in the red boxes. Black boxes indicate examples of correctly detected buildings. The details in the boxes are shown in Figure 13.

Table 2: Quantitative change detection result for Assen.

| Pixel-based | *Comp* | *Corr* | F1 | Object-based | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | GT | TP | FP | *Comp* | *Corr* | F1 |
| Building verification | | | | 952 | 952 | | | | |
| New building | 0.88 | 0.94 | 0.91 | 175 | 163 | 9 | 0.93 | 0.95 | 0.94 |
| Removed building | 0.84 | 0.98 | 0.91 | 11 | 9 | 0 | 0.82 | 1 | 0.90 |

There are 9 (partially) removed buildings correctly detected, while 2 removed buildings are missed. As shown in the upper image of Figure 12c, one small building was removed, however, a car was

parked at the location in the VHR image. As the black car has similar height as the small building, the displacement is too small to be detected in the first step of dense matching. Another removed building is correctly detected after two dense matching steps, However, there is a wall near the removed building in the VHR image as shown in the red boxes of Figure 12c bottom images. The wall occludes parts of grounds resulting in incomplete change detected as shown in the red boxes in the bottom right image of Figure 12c. The updated change is split into two. One, shown in the red box in the bottom right image of Figure 12c, is removed when filtering small changes with areas less than $2 \times 2$ $m^2$. Finally, the remaining change detected covers less than 50% of ground truth. Therefore, the removed building is miss-detected.

The proposed method manages to detect many small buildings and performs effectively in complicated change scenarios. In Figure 13, a large building was subject to complicated change. Some parts were rebuilt while other parts were demolished, regarded as removed buildings. Even making the ground truth takes lots of effort for a human operator. In general, the detections matches well with the ground truth. In addition, three correctly detected small changes are shown in the red boxes in Figure 13c. The detection results shown in Figure 13 proves the good performance of the proposed method. There are still missed detections in several small areas as shown in the blue boxes of Figure 13b. The reason is that no points are reconstructed from images in occlusion areas and low texture sand areas.

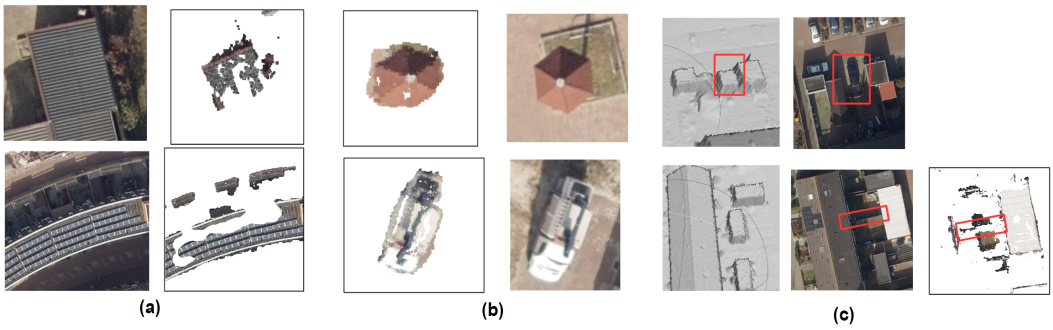


Figure 12: Examples of missed and falsely detected building changes. (a) Missed new buildings due to a failure of the second dense matching step for a building with low and repetitive textures. (b) The false-detected objects of a pavilion in a park and a truck are not considered as buildings. (c) Missed removed buildings due to a car and a wall.

Table 2 summarizes the pixel-based evaluation. The correctness of new and removed building detection are 0.94 and 0.98 respectively. The completeness of new and removed building detection are a bit lower at 0.88 and 0.84 respectively. The completeness of new building detection is affected by the failure of the second step of dense matching. Both new and removed building detection have a high F1 score with values of 0.91. The average height difference between updated point clouds and

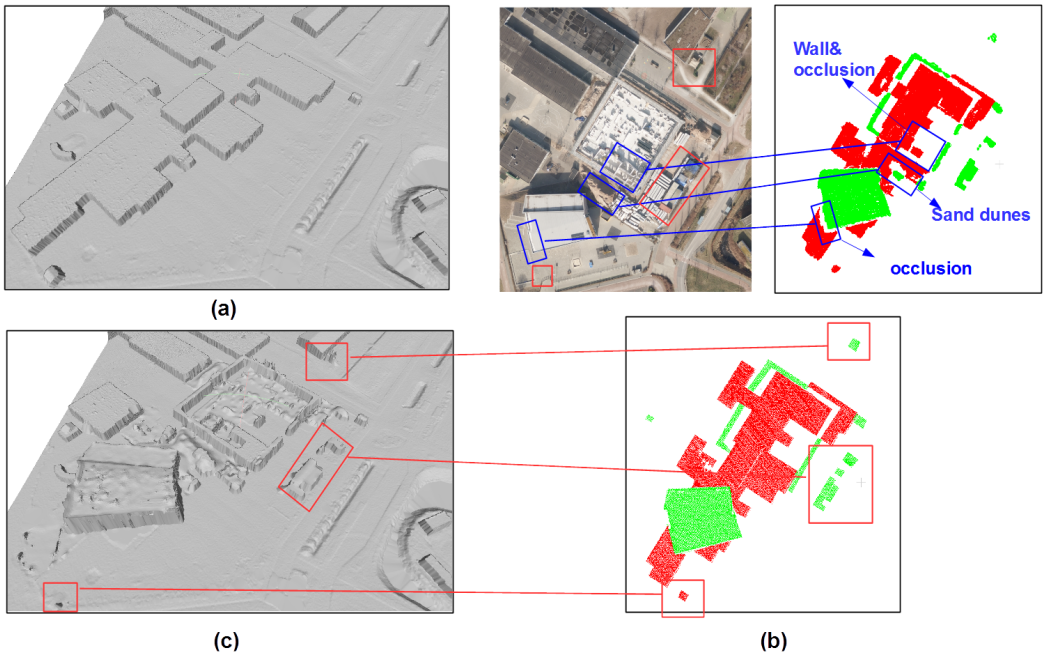


Figure 13: Building change detection and updating for a complicated scenario. (a) 3D mesh of 2012 LiDAR data. (b) Top-left: 2018 VHR image used to update LiDAR data. Top-right: changes detected by proposed method. Removed areas are shown in red, while rebuilt areas are shown in green. Blue boxes show missed detection by the proposed method. Bottom: Ground truth for change detection. (c) 3D mesh of updated LiDAR data. The correctly detected new and removed small buildings are shown in red boxes.

ground truth is 0.32 m.

### *4.3. Discussion*

In this discussion, the main error sources of change detection and updating are first discussed and summarized. Then, parameter settings are discussed. Finally, the contributions are revisited to provide an overall evaluation.

#### *4.3.1. Error sources*

The reasons for false detection of new buildings are various. The main reason is building-like small objects, such as cars, trees, sand piles and pavilions. One false detection in Amersfoort is due to wrong height measurements of a ground window caused by LiDAR penetration. There are three reasons of miss-detection: (1) buildings are covered by trees; (2) missed detection in the first step of dense matching due to homogeneous areas and shadows; (3) missed detection in the second step of dense matching due to the failure of extracting point clouds in the areas affected by shadow, low textures and occlusion. Table 3 shows the number of false detection and miss detection due to the reasons listed above.

In [33], the percentage of overlap, above 50% or 80%, between detected objects and ground truth defines if objects are partially or strongly detected respectively. Table 1 and 2 show that the proposed method provides effective results on detecting change partially. However, when we test for 80% overlap in the Assen experiment, the number of correctly detected (TP) new buildings and removed buildings drops from 163 to 127 and from 9 to 1 respectively. We expect that the main reason is that only one stereo pair is used for detecting changes. The effects of occlusion, shadow and low texture in one stereo pair can be reduced by adding more images.

Table 3: The table of the false detections and miss detections of new buildings in Amersfoort and Assen. (F-DM: first dense matching step. S-DM: second dense matching step)

| False detection | building-like objects | LiDAR penetration | Missed detection | | |
|---|---|---|---|---|---|
| | | | Trees or occluded | F-DM | S-DM |
| Amersfoort | 2 | 1 | 2 | 0 | 2 |
| Assen | 9 | 0 | 0 | 2 | 10 |

*4.3.2. Parameters setting*

Most parameters are involved in the LEAD-Matching step to detect partial changes,which consists of four parts. The parameters in the first two parts, are self-explanatory. In the first part, in order to address the mixed return problem in LiDAR points, a window size is set based on the ground sampling distance (GSD) of the LiDAR points to get better candidates planes for each DSM pixel from its neighborhood. In the second part, in order to create clean segments in the image for dense matching, the filter size for (i) removing small and isolated edge segments, and (ii) filling hole in edge segments are set to 3 and 7 pixels respectively. As the effects considered caused by the transformation between height and disparity, the parameter values can be set the same for different datasets. After candidate disparities are obtained, the disparity range is set to $[-2, 2]$ pixels for reducing the effect of misalignment between two data sources. In [9], the length of the disparity range is suggested to be 5–10 pixels. As long as the two data sources are geo-referenced independently within an accuracy of a few centimeters as described in our data specification, a range length between 5 and 7 pixels is sufficient.

Several parameters in part 3 and 4 are critical. In the association potential of the Markov random field in Equation 3, the parameters of the logistic function $\omega(\bar{c}_{p_i})$ and the constant $\epsilon$ to tune LiDAR guidance in shadow areas are important. The parameters of $\omega(\bar{c}_{p_i})$, defining the probability that a pixel is in the shadow, are set according to the shadow intensity in the VHR images. The parameters $a, b$ and $\theta$ are set to 0.1, 40 and 8 respectively based on the observation that the intensity of shadows is often less than 40 while non-shadows often have values of more than 70. With these parameter

values, the function defines that if $\overline{c}_{p_i}$ is below 40, $\omega(\overline{c}_{p_i}) > 0.9$, while if it is above 70, $\omega(\overline{c}_{p_i}) < 0.2$. In general, the defined parameter values would work effectively on normal images. If shadow intensity is largely affected by illumination and environment reflections, a shadow detection algorithm [34], which is able to adapt to different environments, can be applied. We do not require to set the constant $\epsilon$ to solve problem in shadow completely. As long as the LiDAR guidance term reduces the size of wrong disparity estimation in shadows, false alarms will be reduced. The filter designed to remove speckle and irrelevant changes in the fourth part will help to further reduce false alarms. Empirically, $\epsilon$ is set to 0.1–0.2. In addition, in the interaction potential in Equation 6, the parameters of the logistic function $\omega(\Delta g_{ij})$, defining the probability that an edge occurs between two pixels, is set based on the gradient difference of the two adjacent pixels. The logistic function is defined with the same $a$ and $\theta$, but with a different shifting parameter $b$, now set to 10. If $\Delta g_{ij}$ is smaller than 10, the probability is above 0.9, while if it is above 40, the probability drops below 0.2. These parameter values are expected to be applicable for other image scenarios as the gradient is less affected by illumination.

In the partial change detection part, the color difference of a same object from a stereo pair, acquired consecutively in a forward path with similar illumination conditions, is small. The threshold for detecting changes allows a small color variation, e.g. 20–30. This value works well for different datasets. Even though such small thresholding value results in more partial changes, these changes will be verified in the second step of dense matching. Finally, the value for the filter size to remove speckle and irrelevant changes from actual building changes is set adaptively to different datasets according to Equation 7.

#### *4.3.3. Overall evaluation*

Three contributions are revisited and confirmed by experiments. (1) The comparison of our proposed approach with two other change detection methods shows that LiDAR-guided dense matching successfully addresses the quality problems in both data sources. The related false alarms as shown in Figure 10a and b are largely reduced as shown in Figure 10c. (2) The comparison with the projection–geometry approach also shows that the second step of dense matching is necessary and effective. Otherwise, only partial changes can be derived as shown in Figure 10a. (3) Both experiments show that the proposed method successfully verifies unchanged buildings, while detecting minimum building changes of $2 \times 2 \times 2$ m$^3$ with a high rate of success as indicated in Tables 1 and 2. Examples of accurate change detection were also given in Figure 13, while change detection of small objects was shown in Figure 9e. These results demonstrate that our method could meet the requirements for updating a large scale 3D map.

## 5. Conclusions

This chapter proposes a two-step dense matching framework to detect and update building changes in LiDAR data using new VHR images. The first step proposes LEAD-Matching to derive accurate partial changes by addressing the quality problems from two data sources. Accurate plane information from LiDAR data is exploited to limit the disparity search space for dense matching, especially in the shadow and low texture areas, while dense matching exploited detailed building boundary information in a stereo pair to select optimal disparity. The second dense matching step employs hierarchical dense matching to derive complete changes and update 3D information simultaneously. The proposed method successfully verifies unchanged buildings, while detecting minimum building changes of $2 \times 2 \times 2$ $m^3$ with a F1 score of more than 0.9 in the Assen experiment. This is the first time that it could be shown that the airborne stereo images for 3D change detection and updating can be used to meet the requirements of a large scale 3D map.

## References


[1] PDOK, Inwinjaren AHN2 & AHN3, `http://ahn.maps.arcgis.com/apps/Cascade/index.html?appid=75245be5e0384d47856d2b912fc1b7ed` (2019).

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

[3] H. Hirschmuller, Stereo processing by semiglobal matching and mutual information, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2) (2008) 328–341.

[4] F. Remondino, M. G. Spera, E. Nocerino, F. Menna, F. Nex, State of the art in high density image matching, The photogrammetric record 29 (146) (2014) 144–166.

[5] X. Huang, R. Qin, C. Xiao, X. Lu, Super resolution of laser range data based on image-guided fusion and dense matching, ISPRS Journal of Photogrammetry and Remote Sensing 144 (2018) 105–118.

[6] M. Dalla Mura, J. A. Benediktsson, F. Bovolo, L. Bruzzone, An unsupervised technique based on morphological filters for change detection in very high resolution images, IEEE Geoscience and Remote Sensing Letters 5 (3) (2008) 433–437.

[7] F. Bovolo, S. Marchesi, L. Bruzzone, A framework for automatic and unsupervised detection of multiple changes in multitemporal images, IEEE Transactions on Geoscience and Remote Sensing 50 (6) (2012) 2196–2212.

[8] M. Volpi, D. Tuia, F. Bovolo, M. Kanevski, L. Bruzzone, Supervised change detection in vhr images using contextual information and support vector machines, International Journal of Applied Earth Observation and Geoinformation 20 (2013) 77–85.

[9] R. Qin, Change detection on LOD 2 building models with very high resolution spaceborne stereo imagery, ISPRS Journal of Photogrammetry and Remote Sensing 96 (2014) 179–192.

[10] J. Tian, S. Cui, P. Reinartz, Building change detection based on satellite stereo imagery and digital surface models, IEEE Transactions on Geoscience and Remote Sensing 52 (1) (2014) 406–417.

[11] J. Wu, S. Jie, W. Yao, U. Stilla, Building boundary improvement for true orthophoto generation by fusing airborne lidar data, in: Proceedings of Joint Urban Remote Sensing Event, IEEE, 2011, pp. 125–128.

[12] S. Du, Y. Zhang, R. Qin, Z. Yang, Z. Zou, Y. Tang, C. Fan, Building change detection using old aerial images and new lidar data, Remote Sensing 8 (12) (2016) 1030.

[13] R. Qin, J. Tian, P. Reinartz, 3D change detection–approaches and applications, ISPRS Journal of Photogrammetry and Remote Sensing 122 (2016) 41–56.

[14] J. Trinder, M. Salah, Aerial images and lidar data fusion for disaster change detection, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 1 (2012) 227–232.

[15] A. O. Ok, Automated detection of buildings from single vhr multispectral images using shadow information and graph cuts, ISPRS Journal of Photogrammetry and Remote Sensing 86 (2013) 21–40.

[16] J. Tian, H. Chaabouni-Chouayakh, P. Reinartz, T. Krauß, P. dAngelo, Automatic 3D change detection based on optical satellite stereo imagery, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (2010) 586–591.

[17] T. A. Teo, T. Y. Shih, Lidar-based change detection and change-type determination in urban areas, International journal of remote sensing 34 (3) (2013) 968–981.

[18] P. J. Besl, N. D. McKay, Method for registration of 3-D shapes, in: Sensor Fusion IV: Control Paradigms and Data Structures, Vol. 1611, International Society for Optics and Photonics, 1992, pp. 586–607.

[19] D. Lague, N. Brodu, J. Leroux, Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the rangitikei canyon (nz), ISPRS Journal of Photogrammetry and Remote Sensing 82 (2013) 10–26.

[20] A. Gruen, D. Akca, Least squares 3D surface and curve matching, ISPRS Journal of Photogrammetry and Remote Sensing 59 (3) (2005) 151–174.

[21] A. F. Habib, E. M. Kim, C. J. Kim, New methodologies for true orthophoto generation, Photogrammetric Engineering & Remote Sensing 73 (1) (2007) 25–36.

[22] S. N. Sinha, D. Scharstein, R. Szeliski, Efficient high-resolution stereo matching using local plane sweeps, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1582–1589.

[23] G. Vosselman, B. G. Gorte, G. Sithole, T. Rabbani, Recognising structure in laser scanner point clouds, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 46 (8) (2004) 33–38.

[24] A. Fusiello, E. Trucco, A. Verri, A compact algorithm for rectification of stereo pairs, Machine Vision and Applications 12 (1) (2000) 16–22.

[25] S. Kumar, M. Hebert, Discriminative random fields, International Journal of Computer Vision 68 (2) (2006) 179–201.

[26] J. Niemeyer, F. Rottensteiner, U. Soergel, Contextual classification of lidar data and building object detection in urban areas, ISPRS journal of photogrammetry and remote sensing 87 (2014) 152–165.

[27] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11) (2001) 1222–1239.

[28] M. Rothermel, K. Wenzel, D. Fritsch, N. Haala, Sure: Photogrammetric surface reconstruction from imagery, in: Proceedings LC3D Workshop, Berlin, Vol. 8, 2012, p. 2.

[29] F. Rottensteiner, G. Sohn, M. Gerke, J. D. Wegner, U. Breitkopf, J. Jung, Results of the isprs benchmark on urban object detection and 3D building reconstruction, ISPRS Journal of Photogrammetry and Remote Sensing 93 (2014) 256–271.

[30] C. V. D. Sande, S. Soudarissanane, K. Khoshelham, Assessment of relative accuracy of ahn-2 laser scanning data using planar features, Sensors 10 (9) (2010) 8198–8214.

[31] ISPRS, Evaluation of 2D semantic labeling contest, http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (2019).

[32] M. Schmidt, Ugm: A matlab toolbox for probabilistic undirected graphical models. URL http://www.radig.informatik.tumuenchen.de/ISPRS/WG-III4-IV2-Report.html

[33] F. Rottensteiner, J. Trinder, S. Clode, K. Kubik, Using the dempster–shafer method for the fusion of lidar data and multi-spectral images for building detection, Information fusion 6 (4) (2005) 283–300.

[34] K. Zhou, R. Lindenbergh, B. Gorte, Automatic shadow detection in urban very-high-resolution images using existing 3D models for free training, Remote Sensing 11 (1) (2019) 72.