

Disparity refinement in depth discontinuity using robustly matched straight lines for digital surface model generation

Rongjun Qin, *Member, IEEE*, Min Chen, Xu Huang, Kun Hu

Abstract—Dense Image matching (DIM) using remote sensing images is of particular importance for providing ground geometric data for object modeling and analysis. These methods normally operate in a rectified (epipolar) space producing disparity images that are correlated with the final 3D geometry. However, the resulting digital surface models (DSM) can be problematic on discontinuities of the terrain object, e.g. building edges, which largely limit their practical use for highly accurate object modeling. Existing works put forth efforts for dealing with this issue by defining better edge constraints and more global energy optimization, while intended to improve the disparity maps, it might be challenging to leverage the result of all the pixels through a single energy minimization, leading to either over-smoothed object boundaries or noisy surfaces. In this paper, we propose an intuitive method that integrates straight line primitives to enhance the disparity map. For each matched line, we perform a local discontinuity analysis and propose an intensity-based weighting method for a local plane fitting using iteratively solved weighted least squares adjustment, such that straightness of the object's edges (e.g., buildings) can be preserved, as the straight lines are detected and matched. Experiments on both aerial and satellite dataset show that proposed method yields visually clear object edges and has achieved higher accuracy (2-3 pixels improvement) around edges than DSM generated from typical stereo matching result (i.e. from Semi-global Matching).

Index Terms—Line matching, dense image matching, semi-global matching, disparity refinement, plane adjustment

I. INTRODUCTION

Digital surface models (DSM) are an important source of information for many geoscience and urban applications [1]. The recent development of dense image matching (DIM) in the photogrammetry and computer vision community have largely driven the use of image-derived 3D information, thus in part leading to the prevalence of, e.g. UAV (unmanned aerial vehicle) based applications and 3D remote sensing from stereo spaceborne images [2, 3]. As compared to LiDAR (light detection and ranging), image-derived DSM are normally low

in acquisition / equipment cost, and couples with the color/spectral information that can be easily used for large-scale remote sensing analysis. However, unlike LiDAR which generate independent 3D measurements of the ground objects per light beam, the image-based DSM are largely scene-dependent, thus with varying uncertainties over different objects. A widely known issue for image-based DSM is the lack of certainty at object edges, for example, building boundaries are often extended subject to the enforced smooth constraints of most matching algorithms [4, 5], or being entirely missing due to small object sizes, which leads to large errors in measurements and subsequently associated applications such as building modeling and volume estimation [6, 7]. Although it is generally known that the object edges appear to be sharper and clearer in images, the resulting geometric edges have lower quality than LiDAR data (though sparser) [8]. Therefore, ongoing efforts in dense image matching tend to incorporate the image edge information to yield better matching results.

Existing stereo dense matching algorithms tend to operate in the rectified (epipolar) space, where correspondences follow a perspective co-planar constraint such that they lie on the same row of a pair of rectified images (epipolar images). Thus, the matching is equivalent to finding disparity values (column index difference) between corresponding pixels of two images. There are mainly two common issues that potentially lead to matching errors with the edges: 1) the widely used window-based methods (Fig. 1(a)) similarity measurement may create extra buffers of edges leading to extended edges in the final disparity. With some computational expenses, this issue can be moderately handled by edge-aware windows (i.e. bilateral or image-guided filters) [9, 10] or multi-proposal windows (varying the center position of the window to avoid the inclusion of depth discontinuities) [11]. Besides, computing the per-pixel score is only an initial procedure that normally requires a relatively small window (e.g. 5×5 or 7×7) in the state-of-the-art algorithms, thus the impacts are relatively minor. 2) A more common issue is depth discontinuities around edges, where the occlusions due to relief difference and shadows create problematic regions for correspondence search (Fig. 1(c)). In addition, to deal with the noises and ambiguities in pixel similarity metrics, existing algorithms normally enforce a smooth prior with edge constraints that assume piecewise planar surfaces of the geometry, while suboptimally tuned smooth priors bring extended regions on the edge as well.

R. Qin is with the Dept. of Civil, Environmental and Geodetic Engineering & Electrical and Computer Engineering, The Ohio State University (e-mail: qin.324@osu.edu).

M. Chen is with Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 611756, China (e-mail: minchen@home.swjtu.edu.cn) (*Corresponding author: Min Chen*)

X. Huang and K. Hu are with the Department of Civil, Environmental and Geodetic Engineering, The Ohio State University (e-mail: huang.3651@osu.edu; hu.1775@osu.edu)

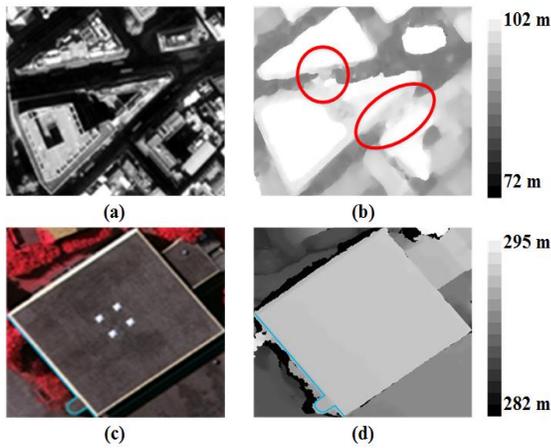


Fig. 1. Problems of existing dense matching methods on edges. (a) and (c) show the original images. (b) and (d) are the corresponding DSM. The red circles in (b) are extra buffers of the edges. The blue lines in (c) and (d) are actual edges of the roof.

Therefore, many approaches were developed to handle the aforementioned issues, including: 1) more complicated matching score computation [12] (e.g. deep-learning based similarity metrics); 2) adaptive edge constraints (e.g. learned from samples) [13]; 3) post-processing methods for disparity refinement. The first two categories of solutions are more frequently investigated and have resulted in noticeable improvement on the overall disparity maps, while these require either data-rich samples for training and careful parameter tuning that leverages the smoothness and edge constraints [14]. The post-processing approach are often heuristic, where refinements are normally done through segmenting the original images into multiple segments [15], while the final refinement results might subject to the segmentation results (e.g. over-segmentation or under-segmentation), as shown in Fig. 2. The red circled regions in Fig. 2(b) are under-segmentation regions where intensities/colors are consistent while disparities are not. Such segmentations might provide wrong cues for post-processing leading to over-smoothed disparity values, as shown in Fig. 2(d). Other works also employ external datasets [16], for example, building footprints to refine the DSM to obtain sharper building boundaries, while such methods might be limited by the availability of the data and the registration accuracy.

Straight line poses important information about the man-made object, which are very often in building edges, and straight-line texture patterns [20]. As compared to segments, straight lines host more semantic information (due to the straight shapes) and can be particularly useful for post disparity refinement. Therefore, in this work, we propose a novel and heuristic solution that takes the matched straight lines in stereo pairs as the units for post disparity refinement. By hypothesizing that the straight line representing a depth discontinuity edge, we use a deterministic approach through analyzing the surrounding buffers to identify actual building edges and perform a weighted local plane adjustment that incorporates disparity of the straight line. The proposed method is able to correct erroneous depth discontinuities and yield clearer boundaries of the manmade objects. The rest of this

paper is organized as follows: section 2 presents the proposed method including robust line matching and the proposed straight-line based plane adjustment method; in section 3, we present the experimental results on both the satellite and aerial dataset and section 4 concludes our paper by analyzing the pros and cons of the proposed method.

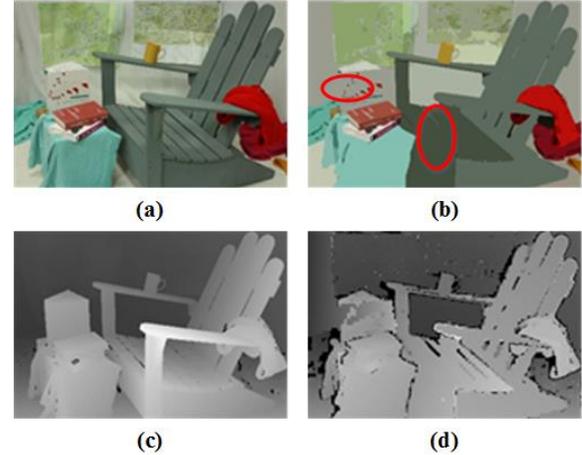


Fig. 2. An example of under-segmentation case. (a) is an original image from Middlebury benchmark [17]; (b) is a mean-shift segmentation result [18]; (c) is the corresponding ground truth; (d) is a post-processing result of semi-global matching [19].

II. METHODOLOGY

An initial disparity map is generated using a hierarchical semi-global matching algorithm [19, 21]. Our proposed method operates on the disparity image, where straight line primitives are extracted and matched using a geometrically constrained line matching method. Then, a consistency check between the disparity map and the matched straight-line primitives is performed to select lines representing depth edges. Finally, the selected lines are used for a plane-based disparity adjustment. Fig. 3 shows the proposed workflow for disparity adjustment, and the complete workflow is introduced through two of the major parts 1) robust line matching and 2) line-based plane adjustments, in subsections II.A and II.B respectively.

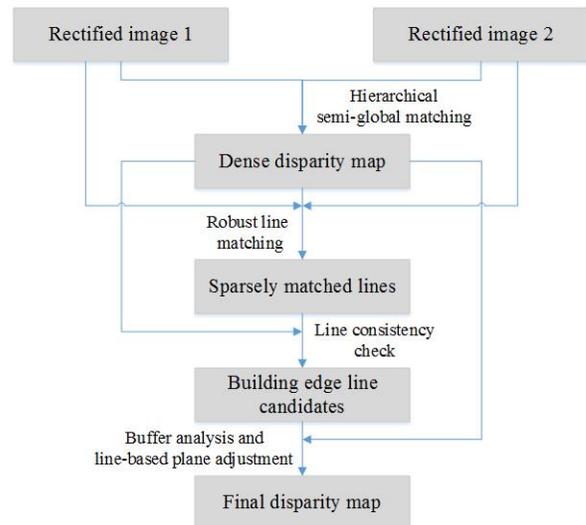


Fig. 3. The workflow of the proposed method.

A. Robust line matching

Matching straight lines is normally more robust than single point matching due to its stronger geometric constraint [22, 23]. The end points of the detected straight lines might not always be correspondences; thus, we consider the epipolar constraint to regularize that the matched lines to have the same starting and end points [24]. Our robust line matching method is based on our previously published work [24] that aimed to design novel descriptors for straight lines, which has shown superior results especially for top-view images. The main difference when applied in our context is that we take the initial disparity map into account to build geometrical constraints, which are expected to reduce false matches and further improve the matching accuracy. The proposed workflow consists of three typical steps: 1) straight line detection and initial matching candidate selection; 2) line feature descriptor extraction and 3) Feature matching.

1) Straight line detection and initial matching

We adopt the state-of-the-art line detector LSD (Line segment detector) [26], which has shown great advances in terms of its speed, flexibility and accuracy. The LSD method can detect sub-pixel level line segments in linear time by analyzing local line-support region based on the Helmholtz principle which makes LSD control false positive without parameter tuning [27]. Normally in this step a large amount of line segments can be extracted. Given that potential matches of lines might not have the same starting and ending points and a permutation strategy for matching is time-consuming, we run a preprocessing step by implementing the epipolar constraints and initial disparity maps (generated from hierarchical SGM [20] through RSP software [21]): for each line on the left epipolar image of a stereo pair, we produce its estimated line on the right epipolar image based on the SGM-derived disparity of the two endpoints of the line, and this provides an initial position for a more refined match. Fig. 4 shows this process, where, for example, p_1 and p_2 are two endpoints of the straight-line L_i in the basic image (e.g. left image in this paper). p_1' and p_2' lying on the epipolar lines e_{p_1} and e_{p_2} are estimated based on the computed disparity values. We define two local regions R_1 and R_2 centered at p_1' and p_2' as the search space for line matching, where t_d indicates the search space size (in the disparity direction), and yellow-dashed lines indicate the bound of the potential matching lines.

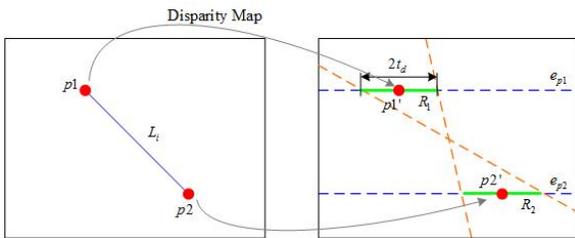


Fig. 4. Candidate matches computation.

2) Line feature descriptor extraction

Based on the search area constraint, for each line L_i on the left image, we search for detected lines in the search region of L_i and then only take the portion of the lines between two

epipolar lines of the two endpoints of L_i (see the right image of Fig. 5). Suppose L_i' is a potential match, we then build a rectangle support window by extending the line in its perpendicular direction, and the feature descriptor is extracted using multi-region histogram of oriented gradient (HOG).

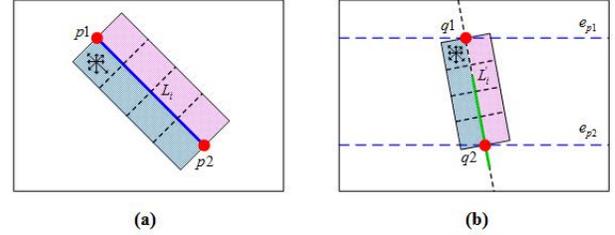


Fig. 5. Line descriptor construction. The blue line is the straight line in the left image. The green line is one of potential corresponding lines in right image.

In a first step, we divide the support window into four sub-regions on each side of straight line (shown in Fig. 5). In each sub-region, a HOG is computed, in which the direction of the straight line is taken as the major orientation direction, to ensure rotation invariance. For each side of the straight line, by combing the HOG of each cell we obtain a 32-dimension feature vector and thus both sides yield a 64-feature vector:

$$Desc = \begin{bmatrix} d_{left} \\ d_{right} \end{bmatrix} \quad (1)$$

where $Desc$ is a 64-dimensional descriptor, which contains two 32-dimensional feature vectors d_{left} and d_{right} corresponding to the left and right support regions of the line, respectively.

3) Feature Matching

Among the candidate matches of a line from the left image, we compute the matchability measurement by nearest neighbor distance ratio (NNDR) strategy with a ratio threshold t_r (the ratio between the smallest distance value over the second smallest distance value). If the ratio is smaller than t_r , this candidate will be identified as a match. Given the possible occlusions on one side of the line at depth discontinuity edge leading to the texture contents being inconsistent, we compute the distance in the NNDR strategy by taking the minimal Euclidean distance of the descriptors of the two texture sections (Eq. (2)):

$$Distance(Desc^{ref}, Desc^{ci}) = \min \left(\begin{array}{l} \|d_{left}^{ref} - d_{left}^{ci}\|, \\ \|d_{right}^{ref} - d_{right}^{ci}\| \end{array} \right) \quad (2)$$

where $Desc^{ref}$ and $Desc^{ci}$ denote the descriptors of the reference line and the i^{th} candidate line, d_{left}^{ref} and d_{right}^{ref} are the feature vectors corresponding to the left and right support regions of the reference line, d_{left}^{ci} and d_{right}^{ci} are the feature vectors corresponding to the left and right support regions of the i^{th} candidate line.

Our method has two tunable parameters t_d and t_r , where t_d refers to the search space and t_r is the ratio threshold for determining a match. A larger t_d will generate more candidate matches. The matching recall will be higher, but the algorithm will require more computation time and vice versa. For the second threshold t_r , increasing the threshold will produce more matches while more outliers at the same time. Empirically,

$t_a \in [5, 10]$ and $t_r \in [0.6, 0.8]$ will be acceptable to yield good matching results in our experiments.

B. Plane-based adjustment for disparity enhancement

We take the matched lines in section II.B as cues to improve the disparity map particularly on objects with straight lines. For straight lines appearing at a disparity discontinuity edge, we assume a planar face with different depths and rotations at both sides, where we proposed a guided plane method for fitting the planar face with the existing disparity values. The basic assumption of our guided plane method is that pixels with similar intensities may share similar disparities, thus along with adjusting the plane equations, the intensity values are able to guide for eliminating disparity outliers. Rectangular regions centered at each straight line are defined as buffers (shown in Fig. 6), from which enough samples are collected for plane adjustments. To ensure consistencies with the matched lines, the disparity values of the matched lines are also incorporated into the plane adjustment.

1) Buffer definition

The rectangle buffer is simply defined with a pre-defined width, generated by taking the straight line as the center line, separating the buffer region into two as S_1, S_2 (Fig. 6).

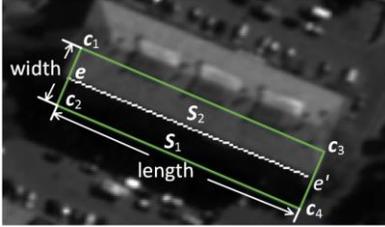


Fig. 6. Buffer definition. White straight line is an edge of a roof. Green rectangle is a buffer centered at the straight line. S_1, S_2 are sets of pixels in both sides of the buffer. (e, e') are endpoints of the straight line. c_1, c_2, c_3 and c_4 are four corners of the buffer.

For the buffer at each side of the straight line (S_1 or S_2), we determine the inside region pixels by scanning pixels over a minimal enclosing region of the buffer, and then computing two cross product results between the length/width of the side and the pixels values, as shown in Eq. (3), where pixels with both cross product results larger than zero are identified as inside-region pixels.

$$p \in \begin{cases} S_1 & \begin{aligned} & \text{if } (\overline{ec_2} \times \overline{ep})(\overline{c_4e'} \times \overline{c_4p}) \geq 0 \\ & \cap (\overline{c_2c_4} \times \overline{c_2p})(\overline{e'e} \times \overline{e'p}) \geq 0 \end{aligned} \\ S_2 & \begin{aligned} & \text{if } (\overline{c_1e} \times \overline{c_1p})(\overline{e'c_3} \times \overline{e'p}) \geq 0 \\ & \cap (\overline{ee'} \times \overline{ep})(\overline{c_3c_1} \times \overline{c_3p}) \geq 0 \end{aligned} \\ \text{Outside regions,} & \text{otherwise} \end{cases} \quad (3)$$

The buffer width ($2r$) is an empirical value. A too small width (< 10 pixels) may not be able to provide sufficient samples for plane adjustment, while a too large value (> 40 pixels) may not satisfy the plane assumption. As the matching accuracies around the edges are usually about 3-5 pixels, we take a width = 20 pixels in all experiments.

2) Straight lines in disparity discontinuity

For disparity enhancement we are interested in objects with straight boundaries (e.g. roofs and other man-made objects). Considering such boundaries often appear in the disparity jump, we only consider straight line matches with large height

differences (i.e. > 3 pixels) in the buffer region on both sides of the reference line (the line on the basic image). Simply taking an average or median value of the two buffer regions might not provide an accurate evaluation of their disparity differences, thus we propose to use a weighted median using the intensity values (Eq. (4)):

$$d_m(S) = \text{med}_{p \in S} \{d(p) | \text{Rep}(d(p), \text{round}(M \cdot w(p)))\} \quad (4)$$

where S is a set of pixels in the same side of buffer, $d_m(S)$ is the weighted median of disparities in S , med refers median filter operation, $d(p)$ refers the disparity value of a pixel p in S , $w(p)$ is the weight of p , which measures the distance between the intensity of p and the predominant intensity (the median value of pixel intensities in the side where p lies) through a Gaussian function. Pixels close to the straight line are not included in the median filter since disparities of these pixels might introduce errors. The basic assumption is that the pixels with intensity values close to predominant intensity refer to those in the major plane of each side, thus this method is able to account for a region containing multiple smaller planes. The median filter is operated on an extended sequence of disparity values, where a disparity value $d(p)$ is replicated by $\text{round}(M \cdot w(p))$ times (referred as the $\text{rep}()$ operation), being said that if the intensity values of p is similar to the median intensity values, we expect the disparity value make higher contribution in the final median filter, thus this value is replicated in the data sequence for filtering. M is an empirical constant, which we set as 3 pixels in our all experiments. The weighted median value is more robust in reflecting the disparity value of a region, as it is able to handle the case where there exist multiple planes (with a dominant one). Essentially, we only consider edges whose disparity differences (computed through weighted median filter) are larger than three pixels.

3) Plane-based adjustment

For each identified edge, the disparities in both buffers are fitted using the plane function in Eq. 5 such that the disparity jump follows the detected straight line.

$$d = ax + by + c \quad (5)$$

where d is the adjusted disparity, (x, y) is the image coordinate of a pixel, a, b , and c are parameters of plane function.

Our task is to accurately compute the plane parameters, while inconsistent disparities (e.g. mismatches) may exist in buffers. To remove such inconsistent disparities, iteratively weighted least squares method is used to compute the optimal plane, which adaptively reduces the weights of inconsistent disparities during the adjustment. In this process we also incorporate the disparity of the matched line as a constraint. The reason that we apply weighted least squares are mainly two-fold: 1) it leverages between the classical statistical least-squares (LS) method and stochastic RANSAC (Random sample consensus) algorithms. Weight LS takes into account the entire observation set and at the same time is able to reduce the impact of outliers. 2) Stochastic approaches such as RANSAC requires large number of observations, while in our case the number of observations might vary with the length of the lines and the size of buffer region where occasionally smaller observation set might lead to unstable results. Weighted LS are able to work with relatively small sized data.

As a first step, we test which one of the two buffer regions the matched straight line belongs to by comparing its average disparity with the predominant disparity of each side (the weighted median values of each side of buffer regions computed through Eq. (4)), using a predefined threshold (3 pixels) in our experiments. It should be noted that if the disparity of the line does not belong to any of these two buffer regions, we regard this matched straight line as an outlier and discard it. As the line parallel to image row direction cannot be used to compute the endpoint correspondences, we do not consider such straight line as the additional constraint in the adjustment. The straight lines are formulated as linear equations with row direction as independent variable and column direction as dependent variable:

$$x = ky + h \quad (6)$$

where x is the image column coordinate, y is the image row coordinate, k and h are parameters of the line equation. On the side which contains the straight line, we take the disparity of the straight line as a constraint in a weighted iterative least squares adjustment, which can iteratively reduce the weights of outliers in the adjustment to get robust fitting results. Given the set of n pixels in one side of buffer $S = \{(x_1, y_1, d_1), (x_2, y_2, d_2), \dots, (x_n, y_n, d_n)\}$ with x being column coordinate, y being row coordinate, d being disparity, we set them as the observations of the plane function:

$$\begin{aligned} d_1 &= ax_1 + by_1 + c \\ &\vdots \\ d_n &= ax_n + by_n + c \end{aligned} \quad (7)$$

Its residual equations being:

$$V = AX - L \quad (8)$$

where V is a vector of least square residuals and X refers to the plane parameters $\{a, b, c\}$, and

$$A = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix}, \quad L = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} \quad (9)$$

For the matched straight line, we substitute Eq. (5) to Eq. (6), thus getting the 3D line function parameterized by the disparity plane.

$$\begin{aligned} d &= a(ky + h) + by + c \\ &= (ak + b)y + (ah + c) = my + t \end{aligned} \quad (10)$$

where m and t are parameters of the disparity function of the straight line, and are computable based on the known disparity values of the matched straight line (two endpoints are enough). The computed m and t directly serve as linear constraints for plane equation adjustment.

$$m = ak + b, t = ah + c \quad (11)$$

Integrating this equation (Eq. (11)) into the observational functions of equation (8) gives an updated Matrix A and L

$$A = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \\ k & 1 & 0 \\ h & 0 & 1 \end{pmatrix}, \quad L = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ m \\ t \end{pmatrix} \quad (12)$$

For the side that the straight line does not fall in, we simply formulate the equation without constraint (9). Since each pixel might contribute differently in the observation function, we use a weighted iterative least squares adjustment that incorporates the differences between the initial disparity and the estimated

disparity in the previous iteration (residuals), formulated in below (Eq. (13)):

$$w^i(x, y) = \exp(-|d^0(x, y) - d^{i-1}(x, y)|/\sigma_d) \quad (13)$$

where w^i weights the observation based on the residuals of the previous iteration, $d^{i-1}(x, y)$ is the estimated value in the previous iteration and $d^0(x, y)$ is the initial disparity. In the first iteration, d^{i-1} is set as the predominant disparity $d_m(S)$ in Eq. (4). The bandwidth of the Gaussian kernel σ_d is set as a large value (5 pixels) to allow disparities forming slanted regions in the first iteration and a small value (1.5 pixels) to allow effective elimination of outliers in next iterations.

In each iteration, the weights are updated based on Eq. (13) serving as a priori matrix for adjustment. For the side of buffer region containing straight line, we give a unit weight to the available line constraint parameters m and t , thus the weight matrix is as shown in Eq. (14). Again, for the side of buffer region without line constraint, the weight matrix follows a dimension of $n \times n$, n being the number of observations, thus generating a priori matrix: $W^i = \text{diag}(w^i(x, y))$ with diag being a diagonal matrix.

The plane parameters are then estimated with a classic weight least squares estimator (Eq. (15)).

$$W^i = \begin{pmatrix} w^i(x_1, y_1) & 0 & \dots & 0 \\ 0 & w^i(x_2, y_2) & \dots & 0 \\ & \vdots & & \\ 0 & 0 & \dots & 1 \ 0 \\ 0 & 0 & \dots & 0 \ 1 \end{pmatrix}_{(n+2) \times (n+2)} \quad (14)$$

$$X = (A^T W A)^{-1} (A^T W L) \quad (15)$$

The computation converges until the residuals $\bar{d} = \sum_{(x,y) \in S} |d^0(x, y) - d^i(x, y)| / n$ is smaller than a predefined threshold τ_d , which is set as 1.5 pixels in our experiments. We use the computed plane parameters to correct disparities of pixels in the side according to Eq. (5). To ensure smooth transition from the buffered region and other regions, we only adjust pixels whose intensity is similar to the predominant intensity of the side (intensity difference less than 15). In the case that the adjustment computation does not converge, the original disparity is kept.

The main difference between traditional iterative weighted least squares with variable weights and our method is the weighting strategy in the first iteration, where traditional method uses identical weights in the first iteration, while our method adaptively defines weights in the first iteration (Eq. (13)). The comparison results of both iterative methods are shown in Fig. 7. Both methods are able to improve the resulting disparity map, while our adaptive weighting (Fig. 7(b)) shows a sharper boundary (corresponding to the red line in Fig. 7(a)) than the identical weighting (Fig. 7(c)) due to the low weights of inconsistent disparities in our method. Fig. 8 shows that it is necessary to perform the iteratively weighted adjustment for plane fitting: we compared the results with iterative solution (until converges to a threshold τ_d) and a non-iterative approach (i.e., taking the first iteration) under the same weighting strategy based on (Eq. (13)). The iterative solution (Fig. 8(b)) shows more complete edges (corresponding to red lines in Fig. 8(a)) than the non-iterative scenario (Fig. 8(c)). It is because the

weights of some inaccurate observations may still be high after only one iteration, instead, they can be decreased iteratively, thus generating more accurate edges.

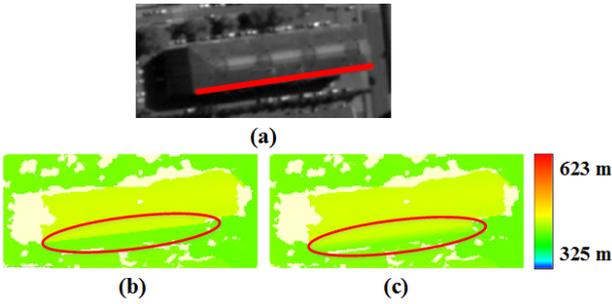


Fig. 7. Comparison between the proposed method and the equal weight case. (a) original image. (b) intensity-adaptive weighting using Eq. (4). (c) identical weighting in traditional method. Where (b) in the circled area appears to be sharper than that in (c).

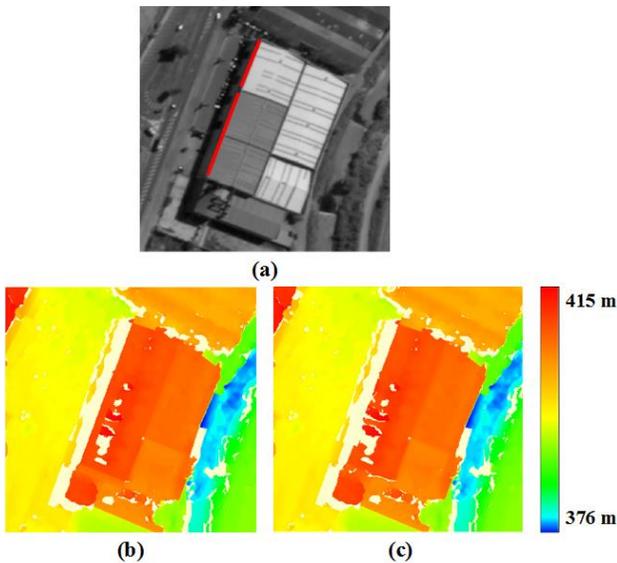


Fig. 8. Comparison between the proposed method and the non-iterative case: (a) original image; (b) our method with iterative solution; (c) our method with non-iterative solution.

III. EXPERIMENT

Experiments are performed on the aerial and satellite stereo images, particularly on urban scenes with dense urban objects [28, 29]. The ground truths data are available as LiDAR (Light detection and Ranging) based DSM. The stereo images are rectified as epipolar images for matching, and then the computed disparity values are then triangulated as DSM for comparison. Both of these two datasets are from ISPRS Benchmark, including a stereo pair of Vaihingen aerial images and WorldView-1 stereo images of Terressa. For each dataset, we compute three DSMs: 1) the original DSM from SGM: ODSM; 2) The LiDAR derived DSM, LDSM, and 3) DSM generated by the proposed post-processing method on SGM-derived DSM, PDSM. The datasets used for experiments are shown in Fig. 9, where the aerial dataset has an average of 9 cm GSD (ground sampling distance) mainly containing moderate sized residential houses and the satellite dataset has a GSD of 0.5 meters with many industrial buildings.

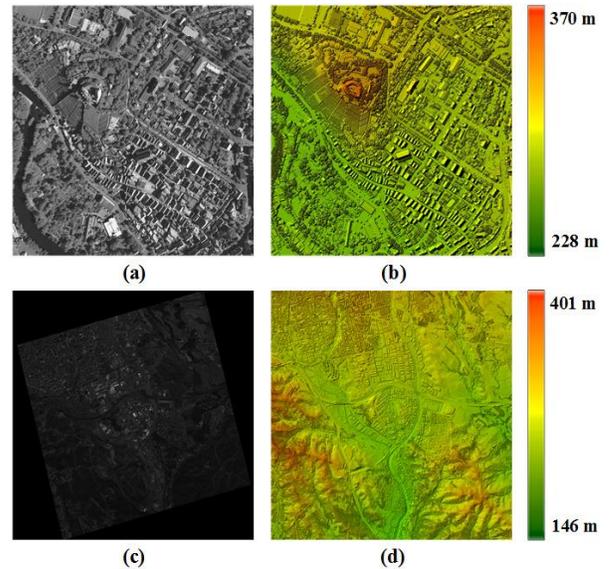


Fig. 9. The experimental datasets (a) the left image of the aerial dataset (8014×14109 pixels); (b) the corresponding LiDAR DSM of the aerial dataset; (c) the left image of the satellite dataset (13420×12590 pixels); (d) the corresponding LiDAR DSM of the Satellite dataset.

The experiments include two parts: straight line matching method evaluation and disparity enhancement evaluation.

A. Straight line matching method evaluation

Two pairs of sub-regions (Fig. 10) are selected from the aerial and satellite stereo images to evaluate the performance of the proposed line matching method. We compared the proposed approach with three other state-of-the-art straight-line matching methods: MSLD [30], LPI [23] and LJJ [31]. For fair comparison, the same line segment detector (LSD) [26] is adopted to detect line segments and the straight lines shorter than 30 pixels are eliminated before matching for all comparative methods. The parameters of all comparative methods are set following their literature's suggestion. In the proposed method, the threshold t_d was set as 5 pixels and t_r was set as 0.8.

Three metrics are used to evaluate the matching performance: number of correct matches (NCM), matching precision (MP), and execution time (ET). The NCM value of each method on each image pair is computed manually. After the NCM value is obtained, the MP value is computed as:

$$MP = (NCM/NTM) \cdot 100\% \quad (16)$$

where NTM (number of total matches) is the number of obtained matches.

Table I and Table II show the quantitative comparisons in terms of NCM and MP between the proposed method and MSLD, LPI and LJJ on both image pairs. It can be seen from the results in Table I and Table II that the proposed matching method performs best in terms of NCM and MP on both image pairs. Especially on the satellite image pair, there are many parallel and similar lines that are difficult to be distinguished by MSLD, LPI and LJJ. The proposed method doubles the number of correct matches, with the improvement being more than 10 percent in MP , demonstrating the effectiveness of the proposed method. The performance of the proposed method

mainly benefits from the geometrical constraint that developed from the initial disparity map, which reduces the search area of correspondence. Fig. 11 and Fig. 12 show the matches obtained by the proposed method and the comparative methods.

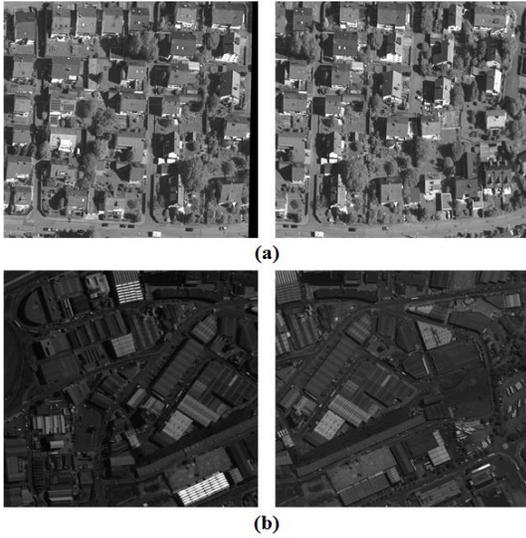


Fig. 10. The experimental datasets for line matching method evaluation. (a) image pair from the aerial stereo images. Image size is 2015×1480 pixels; (b) image pair from the satellite stereo images. Image size is 1000×1000 pixels.

TABLE I
NCM VALUES

Method	MSLD	LPI	LJL	Proposed
Aerial image pair	139	196	156	219
Satellite image pair	29	45	59	120

TABLE II
MP VALUES (%)

Method	MSLD	LPI	LJL	Proposed
Aerial image pair	81.29	86.73	92.31	96.90
Satellite image pair	70.73	64.29	86.76	96.77

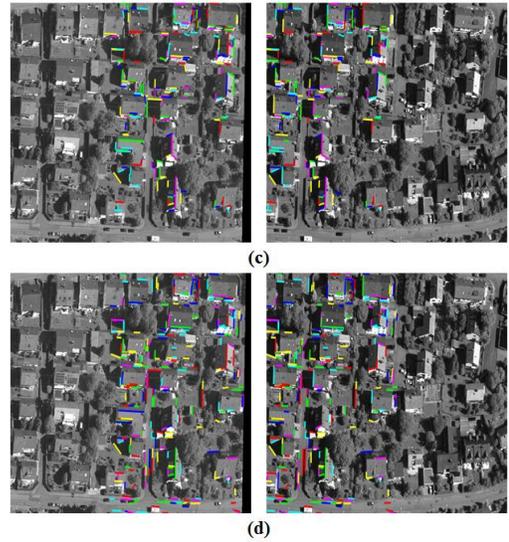
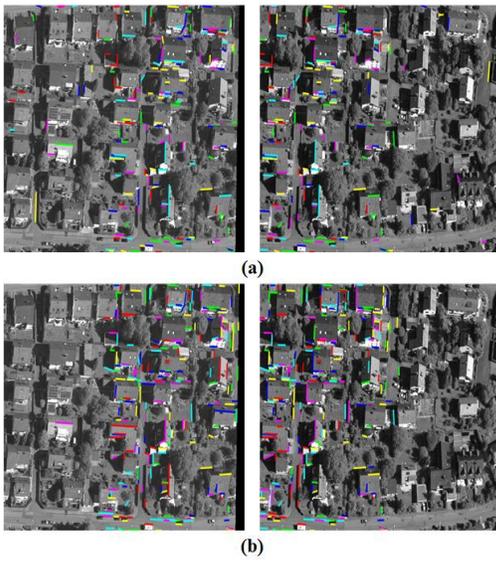


Fig. 11. Straight line matches on the aerial image pair obtained by different methods. (a) MSLD; (b) LPI; (c) LJL; (d) Proposed. Different colors are used to distinguish different match pairs.

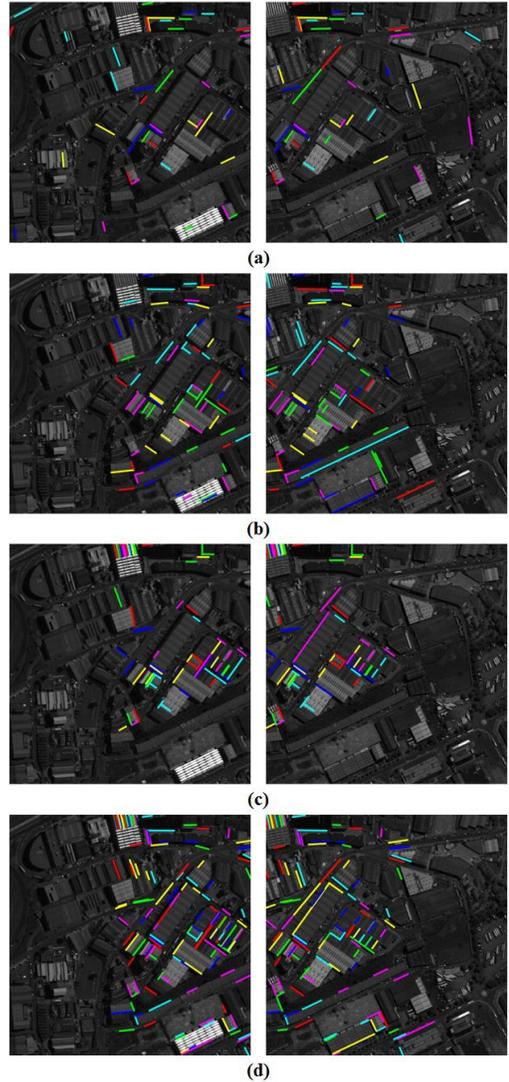


Fig. 12. Straight line matches on the satellite image pair obtained by different methods. (a) MSLD; (b) LPI; (c) LJL; (d) Proposed. Different colors are used to distinguish different match pairs.

In term of time efficiency, we compared the proposed method with LJJ by considering that LJJ achieves the second-best performance and it is implemented by the authors. Table III shows the execution time of LJJ and the proposed matching method on both image pairs. Both methods are performed under the same environment (Intel Core i7 2.50 GHz Windows 7 PC). It can be seen that the time efficiency of the proposed method is superior to LJJ.

TABLE III
EXECUTION TIME (S)

Method	LJJ	Proposed
Aerial image pair	26.87	7.67
Satellite image pair	25.61	4.11

We also evaluate the feature descriptor used in the proposed method. The two-side descriptor adopted in this study is labeled as TSD. The whole-region descriptor computed from the whole support window centered on the straight line is labeled as WRD. The matching results of the methods by combining the proposed matching framework and TSD, the proposed matching framework and WRD are shown in Table IV.

TABLE IV
NCM AND MP VALUES BASED ON TSD AND WRD

Method	TSD		WRD	
	NCM	MP (%)	NCM	MP (%)
Aerial image pair	219	96.90	196	97.03
Satellite image pair	120	96.77	97	96.04

It can be seen from Table IV that the methods based on both the two descriptors achieve similar matching precision. But the method based on TSD increases the number of correct matches significantly. This is because TSD is more robust than WRD to image geometric distortion in discontinuous area caused by viewpoint change.

B. Disparity enhancement evaluation

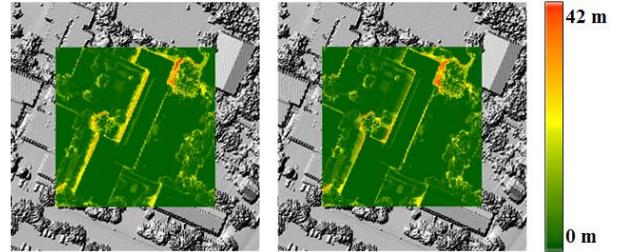
In total 12372 straight lines are matched in the aerial dataset (Fig. 9(a)). The identified straight lines in the disparity discontinuities are 5950 in number. Given that the images roughly contain 596 buildings, this is said that on average 10 edge lines are detected per building. Without tuning any parameters, we obtained statistics for detected edge lines on the satellite image dataset (Fig. 9(c)), being: 21454 straight lines are matched and 8346 of them are identified as edge lines, based on the rough number of the buildings in the scene (3519 buildings) this refers to 2 edge lines per building. The average number of edge lines for each building on the satellite image is smaller than that on the aerial image because some building edges have not been extracted from the satellite image due to the relatively lower spatial resolution.

The enhanced DSM using the proposed method (PDSM) and the ODSM are compared with the ground truth LDSM after an intuitive 3D-shift based registration for compensating the systematic errors. The resulting RMSE (root-mean squared error) in different scenarios are shown in Table V, where the overall RMSE of the DSM and “Buffered regions” only consider the improved regions, essentially the buffered regions.

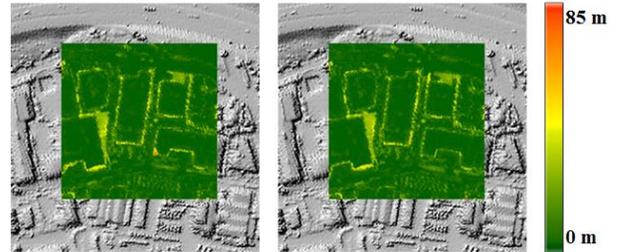
TABLE V
COMPARATIVE RESULTS

RMSE (meter)	ODSM	PDSM
Aerial - Overall	1.180	1.111
Aerial - Buffered Regions	3.916	2.999
Satellite - Overall	1.648	1.646
Satellite - Buffered Regions	5.250	4.817

It should be noted in Table V that although the improvement of the entire DSM is not statistically significant, there is a notable enhancement in areas with disparity jumps, which can potentially facilitate critical tasks in object extraction and modeling. An example of error distribution of each dataset before and after our proposed post-processing method is shown in Fig. 13, which shows that the boundary errors are clearly reduced after post-processing using our line-based disparity enhancement method.



(a) An example in the aerial dataset



(b) An example in the satellite dataset

Fig. 13. An example showing the error distribution around the edges. Left: before and right: after post-processing.

In Figs. 14-17 we choose two representative examples from each dataset showing that the area around the building boundaries are improved to straight lines, and some showing even better visual effect than the LiDAR data (e.g. circled regions). Figures are properly rendered for visualization purpose.

The cross-section analysis on some of the edges in Figs. 14-17 shows a significant improvement (e.g. from a RMSE of 6.07 meters to 1.72 meters for the first satellite example, from 2.06 meters to 0.67 meters for the second satellite example, from 2.45 meters to 1.57 meters for the first aerial example, and from 0.51 meters to 0.26 meters for the second aerial example). As compared to the original SGM algorithm in the profile of these four examples, it all shows that profile of the proposed method have reached to a quality closer to and somehow even less wavy than the LIDAR profile.

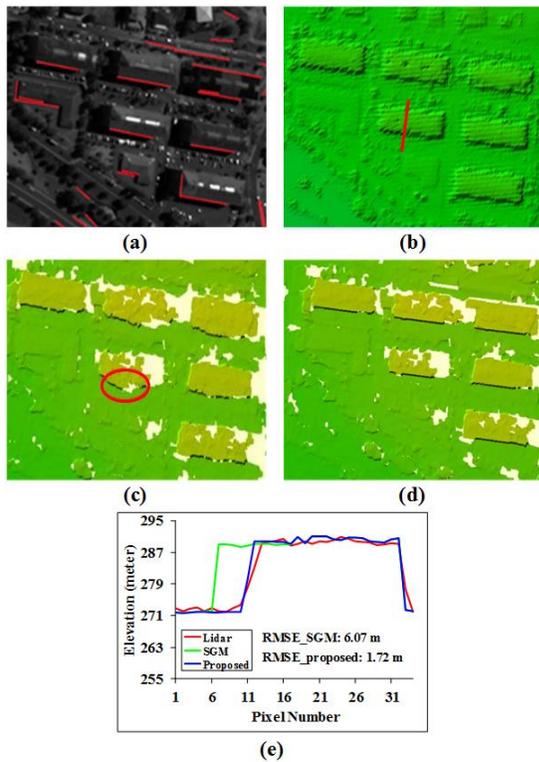


Fig. 14. The first example of satellite dataset showing that the proposed post-processing method is able to recover sharp building edges and fill missing regions around the disparity jumps (e.g. red circled region).

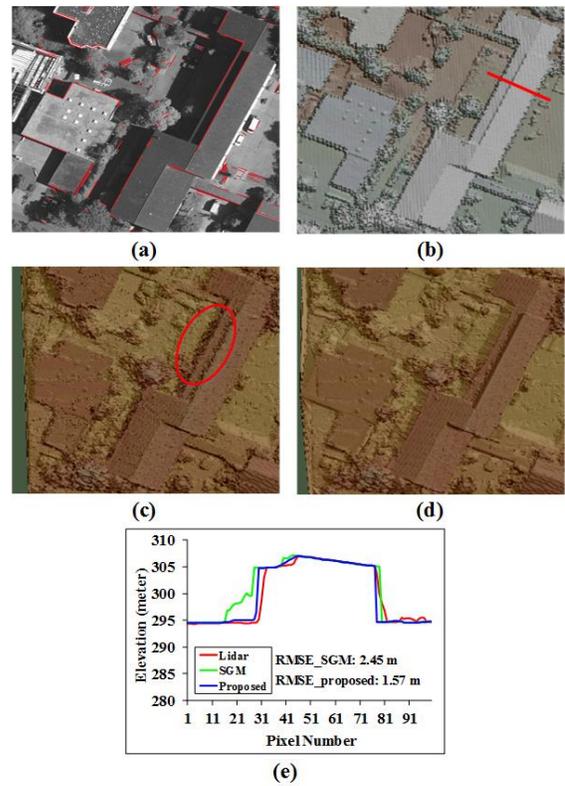


Fig. 16. The first example of aerial dataset showing that the proposed post-processing method is able to recover sharp building edges and fill missing regions around the disparity jumps (e.g. red circled region).

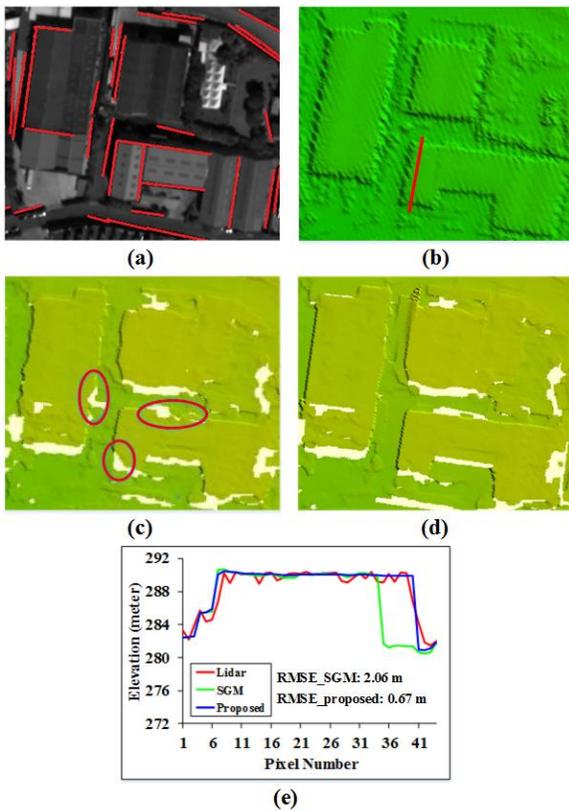


Fig. 15. The second example of satellite dataset showing that the proposed post-processing method is able to recover sharp building edges and fill missing regions around the disparity jumps (e.g. red circled region).

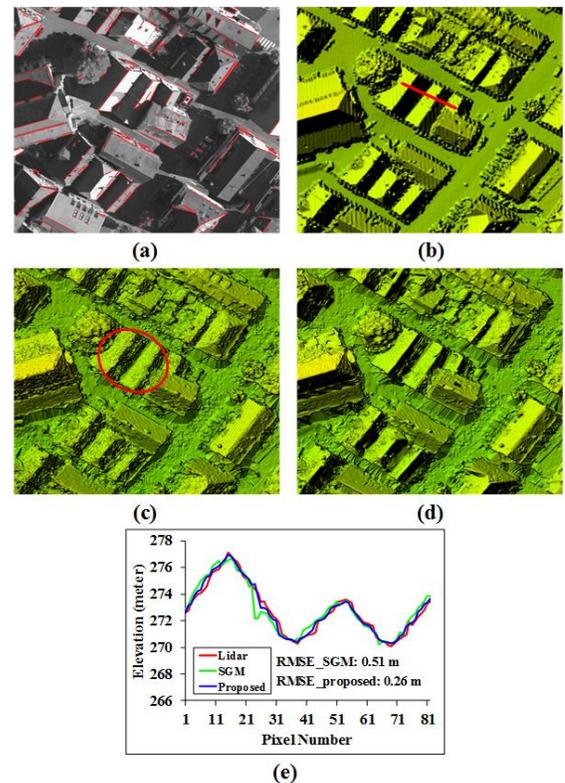


Fig. 17. The second example of aerial dataset showing that the proposed post-processing method is able to recover sharp building edges and fill missing regions around the disparity jumps (e.g. red circled region).

We compared the proposed disparity enhancement method with the weighted median filtering method [32], which is often used as a post-processing for dense matching results. The refinement results in Fig. 18 shows that the weighted median filtering method may not be able to enhance results of regions where a large amount of mismatches present (e.g. red-circled regions). This method is essentially a fixed-window approach, which is not able to process regions with variable sizes. Our proposed method refines disparities in the support window of the whole roof boundaries, which is more robust to such cases.

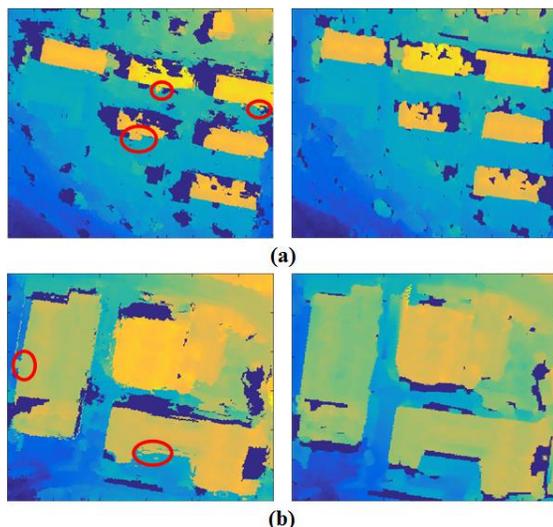


Fig. 18. Comparison of the weighted median filtering and the proposed method. (a) and (b) are two pairs of examples on the satellite dataset. The left one and the right one of each pair are the results of the weighted median filtering method and the proposed method, respectively.

Based on the comparative experiments, it is expected that a DSM processed by our method might be able to better accommodate many algorithms such as building detection, change detection and object modeling. It should be noted that such improvement may only occur in the buffered region of which a straight line is detected and matched, while this local correction provides a valid way of locally incorporating highly reliable information into the matching.

IV. CONCLUSION

In this paper, we present a stereo-matched line-based disparity enhancement method for improving the image-based DSM. A novel procedure that integrating initial disparities into a robust invariant line matching algorithm for efficient edge line detection, and an intensity weighted plane fitting method are developed to optimally correct the disparity values on each side of the straight lines. The analytical results in section 2.2.3 shows that our intensity-based weighing strategy is more effective than traditional plane fitting methods, and experiments on both aerial and satellite stereo images show that the proposed method is able to improve the quality of DSM around the man-made object boundaries to notable level, where we show examples in profile analysis that a case of satellite data by improving the RMSE from 2.67 meters to 0.6 meters and a case for area image from 0.51 meters to 0.26 meters, approximately 2-3 pixel accuracy improvement. Moreover, our

proposed method is not restricted to only improve the DSM generated from SGM, it can be used for improve any DSM where stereo images are available. However, we are also aware that the proposed method is a local method and the results are largely dependent on the matched lines. Therefore, in the future work we will focus on more robust line detection and matching algorithms, and potentially include parameterized curve line for DSM quality improvement.

REFERENCES

- [1] G. Zhou, C. Song, J. Simmers, and C. Peng, "Urban 3D GIS from LiDAR and digital aerial images," *Comput. Geosci.*, vol. 30, no. 4, pp. 345-353, 2004.
- [2] F. Nex and F. Remondino, "UAV for 3D mapping applications: a review," *Appl. Geomatics*, vol. 6, no. 1, pp. 1-15, 2014.
- [3] L. Zhang and A. Gruen, "Multi-image matching for DSM generation from IKONOS imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 60, no. 3, pp. 195-211, 2006.
- [4] C. Stal, F. Tack, P. De Maeyer, A. De Wulf, and R. Goossens, "Airborne photogrammetry and lidar for DSM extraction and 3D change detection over an urban area—a comparative study," *Int. J. Remote Sens.*, vol. 34, no. 4, pp. 1087-1110, 2013.
- [5] S. Gehrke, K. Morin, M. Downey, N. Boehrer, and T. Fuchs, "Semi-global matching: An alternative to LIDAR for DSM generation," in *Proc. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Calgary, AB, Canada, 2010, vol. 38, (B1), p. 6.
- [6] M. A. Aguilar, M. del Mar Saldaña, and F. J. Aguilar, "Generation and quality assessment of stereo-extracted DSM from GeoEye-1 and WorldView-2 imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 1259-1271, 2014.
- [7] C. Brenner, "Building reconstruction from images and laser scanning," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 6, no. 3-4, pp. 187-198, 2005.
- [8] X. HUANG, "Building reconstruction from airborne laser scanning data," *Geo-spat. Inf. Sci.*, vol. 16, no. 1, pp. 35-44, 2013.
- [9] Q. Yang, "Hardware-efficient bilateral filtering for stereo matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 1026-1032, 2014.
- [10] Q. Yang, D. Li, L. Wang, and M. Zhang, "Full-image guided filtering for fast stereo matching," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 237-240, 2013.
- [11] J. Joglekar and S. S. Gedam, "Area based image matching methods—A survey," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 1, pp. 130-136, 2012.
- [12] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, pp. 1-32, 2016.
- [13] A. Seki and M. Pollefeys, "Sgm-nets: Semi-global matching with neural networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA, 21-26 July 2017.
- [14] Y. Zhan, Y. Gu, K. Huang, C. Zhang, and K. Hu, "Accurate image-guided stereo matching with efficient matching cost and disparity refinement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1632-1645, 2016.
- [15] X. Huang, Y. Zhang, and Z. Yue, "Image-guided non-local dense matching with three-steps optimization," in *Proc. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. III-3, pp. 67-74, 2016.
- [16] M. Brđić, O. Tournaire, B. Vallet, and N. Champion, "Extracting polygonal building footprints from digital surface models: a fully-automatic global optimization framework," *ISPRS J. Photogramm. Remote Sens.*, vol. 77, pp. 57-65, 2013.
- [17] Middlebury stereo vision page [Online]. Available: <http://vision.middlebury.edu/stereo/>
- [18] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603-619, 2002.
- [19] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328-341, 2008.

- [20] H. Li, X.G. Zhang, and Z. Sun, "A line-based adaptive-weight matching algorithm using loopy belief propagation," *Mathematical Problems in Engineering*, pp. 1-13, 2015.
- [21] R. Qin, "Rpc Stereo Processor (Rsp)—a Software Package for Digital Surface Model and Orthophoto Generation From Satellite Stereo Imagery," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, III-1, pp. 77-82, 2016.
- [22] M. Al-Shahri and A. Yilmaz, "Line matching in wide-baseline stereo: a top-down approach," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4199-4210, 2014.
- [23] B. Fan, F. Wu, and Z. Hu, "Robust line matching through line-point invariants," *Pattern Recognit.*, vol. 45, no. 2, pp. 794-805, Feb. 2012.
- [24] M. Chen, S. Yan, S. Huang, and Q. Zhu, "Accurate and reliable line segment matching improvements in oblique image processing," *In Proceedings of IGTF – Imaging Geospatial Technol. Forum, ASPRS Annu. Conf.*, Baltimore, MD, USA, 11-17 March 2017.
- [25] C. Schmid and A. Zisserman, "Automatic line matching across views," *in Proc. of IEEE Internatioanl Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997.
- [26] R. G. Von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722-732, 2010.
- [27] N. G. Cho, A. Yuille, and S. W. Lee, "A Novel Linelet-Based Representation for Line Segment Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1195-1208, 2018.
- [28] ISPRS Benchmark [Online]. Available: <http://www2.isprs.org/commissions/comm3/wg4/tests.html>.
- [29] ISPRS Benchmark [Online]. Available: <http://www2.isprs.org/commissions/comm1/wg4/benchmarktest.html>.
- [30] Z. Wang, F. Wu, and Z. Hu, "MSLD: A robust descriptor for line matching," *Pattern Recognit.*, vol. 42, no. 5, pp. 941-953, 2009.
- [31] K. Li and J. Yao, "Line segment matching and reconstruction via exploiting coplanar cues," *ISPRS J. Photogramm. Remote Sens.*, vol. 125, pp. 33-49, 2017.
- [32] M. Mozerov and J. Van de Weijer, "Accurate stereo matching by two-step energy minimization," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp.1153-1163, 2015.