Automation in Construction 64 (2016) 65–77

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Review

3D terrain reconstruction of construction sites using a stereo camera

Changhun Sung *, Pan Young Kim

Hyundai Heavy Industries, Republic of Korea

ARTICLE INFO

Article history:
Received 16 March 2015
Received in revised form 15 December 2015
Accepted 28 December 2015
Available online 23 January 2016

Keywords:
Dense 3D reconstruction
Multi-scale descriptor
Cascade matching
Probabilistic 3D model

ABSTRACT

This paper presents a fast and robust three-dimensional (3D) terrain reconstruction system that uses a stereo camera. Local feature-based dense 3D reconstruction consists of two major steps: matching correspondence points and dense 3D reconstruction. In the matching step, the descriptor is an important component, as its properties significantly affect the precision of the matching. Furthermore, matching is the most time-consuming step. In this paper, correspondence points are found using multi-scale descriptors (MSDs) because of their robustness and computational efficiency. A two-stage cascade matching method suitable for MSDs is also proposed. In the dense 3D reconstruction step, a probabilistic model is proposed for dense reconstruction that provides high precision through the use of robustly matched correspondence points and computational efficiency by narrowing the search range using coarsely inferred disparity values from precisely calculated triangle meshes. To collect experimental data, a prototype stereo camera system is also built that is mounted on the front of an excavator. This paper concludes by comparing the proposed dense 3D reconstruction with different types of dense 3D reconstruction methods in terms of processing time and similarity to the shape of the terrain. The results from an evaluative experiment show that MSD-based dense 3D reconstruction is suitable for various autonomous control applications in construction sites where computation time and precision are vital.

Contents

1. Introduction

Even though the problem of three-dimensional (3D) terrain reconstruction is essential for the automation of construction sites, it is still far from completely solved because of the challenging construction environment that consists of irregularly shaped and textureless ground, as shown in Fig. 1. Once the 3D terrain has been reconstructed, it can be used in various applications such as path planning, navigation, and various autonomous control applications. Reconstructed terrain also plays an important role in the areas of safety and productivity, where it is able to significantly improve the safety and productivity of hazardous

* Corresponding author.
E-mail address: sungch@hhi.co.kr (C. Sung).

http://dx.doi.org/10.1016/j.autcon.2015.12.022
0926-5805/

environments by helping operators recognize its properties more efficiently.

In traditional terrain reconstruction techniques, there are two major approaches. One is vision-based terrain reconstruction, and the other is Light Detection And Ranging (LiDAR)-based terrain modeling [1,5]. In many fields, such as the construction industry and especially for construction site surveying, LiDAR is used to scan the real environment to generate 3D discrete surface samples. LiDAR provides highly accurate distance measurements of the observed surface. Because of its wide field of view (FOV), LiDAR can acquire wide-range measurements that cannot be obtained with vision sensors [2]. However, LiDAR-based terrain reconstruction is limited when applied to practical 3D construction. Scanning construction sites with LiDAR and post-processing from measured 3D point clouds for 3D modeling is a time-consuming task, because of the considerable size of the point cloud (HDL-64E, over 1.3 million points) compared to a vision sensor (PointGrey Bumblebee, under 0.3 million points). Furthermore, in order to operate LiDAR, significant power is required, and it is difficult to supply such power to automated construction equipment attached to the LiDAR interface.

In order to overcome the disadvantages of LiDAR-based terrain reconstruction, vision sensors are often used. Vision sensors can capture the construction-site environment more quickly (PointGrey Bumblebee, over 30 Hz) than LiDAR (HDL-64E, about 15 Hz). Consequently, vision sensors reflect changes to environmental conditions more rapidly. Furthermore, vision-based terrain reconstruction requires less computation time than LiDAR-based terrain reconstruction, owing to the use of optimized computer vision technology. Vision-based terrain reconstruction can thus be applied to practical 3D automated construction. In terms of the reconstruction system, vision sensors consume less power than LiDAR sensors, and they are also inexpensive and lightweight. This makes it easy to build a reconstruction system for automated construction equipment. Therefore, various computer vision techniques have been used for 3D modeling systems such as Simultaneous Localization And Mapping [3] and Structure from Motion [4].

Nevertheless, reconstructing the 3D terrain of a construction site using a camera sensor is also problematic. The ground at a construction site comprises textureless surfaces and repeating patterns, such as muddy areas and dirt roads. Furthermore, the dominant plane of the ground is highly slanted when images are captured with excavator-mounted forward-looking cameras. This makes it difficult to reconstruct 3D terrain because the widely used feature-matching based techniques implicitly assume that the surface of the ground is perpendicular to the image plane [12]. Another major problem is the computation time. In order to use the reconstructed 3D terrain in construction-site applications, the total computation time is required less than 1 s (more than 1 frame/s). To ensure safety at construction sites, operators are instructed to move the equipment at no more than 15 km/h (4.1 m/s). Therefore, it is sufficient to provide reconstructed 3D terrain information to the automated construction equipment each second.

In order to solve the 3D reconstruction problem in these challenging environments, we present a novel local-feature-based dense stereo matching algorithm suitable for use on a construction site. In practical construction-site applications, algorithms have to run quickly and robustly. In order to achieve these requirements, we first determine the correspondence points quickly using a multi-scale descriptor (MSD) [13] without sacrificing matching precision. We then estimate the dense stereo using our proposed probabilistic model. This proposed method not only requires less computation than other robust descriptors such as SIFT and SURF, but also it reconstructs highly precise construction site terrain.

This paper is organized as follows. After discussing related work and proposed framework in Section 2, we introduce our matching pipeline in Section 3 and describe our dense reconstruction model in Section 4. In Section 5, we present our results and compare our algorithm with other dense reconstruction methods. Finally, we conclude the paper in Section 6.

2. Related work and proposed framework

2.1. Related work

One well-known vision-based approach to 3D reconstruction is global correspondence-based stereo matching. Global methods provide dense and accurate matching results by imposing continuity constrains. They determine correspondence points by minimizing an energy function. In order to optimize this energy function, various approximation algorithms have been proposed such as graph-cuts [6] and belief propagation [7]. Global methods perform well; however, these methods are limited in practical applications because of their high computational cost and memory requirements. In order to mitigate the disadvantages of global methods, semi-global methods [8,9] have been proposed. Semi-global methods estimate correspondence points from initial correspondence points, called seeds. Once the initial correspondence seeds are found, they are used to estimate disparity values. Semi-global methods require less computation than global methods. However, in these approaches, the range of disparity has to be decided beforehand and good parameter selection is critical. Furthermore, semi-global methods have difficulty reconstructing feature-poor slanted surfaces such as construction site ground.

Fig. 1. Type of work task under consideration.

Another popular approach is local correspondence point-based stereo matching [24,25]. These feature-based dense 3D reconstruction methods generally consist of four steps: interest point detection, descriptor generation, feature matching, and dense 3D reconstruction. In feature-based dense stereo matching in outdoor construction environments, the descriptor is an especially important component because descriptor-based matching results significantly affect the precision of the estimated 3D point cloud, and the computation of the descriptor is the most time-consuming part of the overall dense stereo matching process.

The neighborhood of an interest point can be represented in various ways, for instance, by its gradient information [14] or intensity comparison [15]. The simplest descriptor is the pixel intensity within a certain region around the extracted interest point. This descriptor is not robust because its pixel values can change according to imaging conditions such as illumination, scale, or viewpoint. The most widely used descriptor is probably the SIFT descriptor [10]. In order to describe an interest region, the SIFT descriptor uses local gradient histograms that are sampled in a square grid around the interest point. This descriptor has been shown to outperform others in terms of robustness of scale, rotation, viewpoint, and illumination variation [14]. However, increased complexity and robustness comes with an increase in computation. Therefore, to reduce the computational effort, there have been many attempts to develop descriptors that are faster to compute and match. One of these descriptors is SURF, proposed by Bay et al. [11]. This descriptor describes the region of interest as the sum of the Haar wavelet responses using an integral image in order to achieve computational efficiency. SURF reduces computational effort compared to SIFT. Even so, SURF descriptors are still too slow to apply to the dense stereo matching task. Therefore, various evaluations have shown that the high computational cost of SIFT- and SURF-based reconstruction systems restrict their use in real applications [23].

As mentioned, construction terrain is very ambiguous, making it difficult to find correspondence points. Hence, robustness is the key property of a point descriptor for construction site terrain. Furthermore, computational time is another important property for real applications. Recently, Sung et al. [13] proposed a fast and robust descriptor called the MSD for practical outdoor stereo camera applications. This descriptor has already performed well in outdoor visual motion estimation tasks. Therefore, in this paper, correspondence points are found using MSD.

Another important part of dense 3D reconstruction algorithms is the calculation of dense stereo matches. Various methods have been proposed for dense stereo matching. Many state-of-the-art methods find correspondence points between stereo input images using local features and then impose global shape constraints such as dynamic programming [16] or graph-cuts [6]. In these local feature-based dense 3D reconstructions, the results of sparsely matched points are used as anchors for inferring the disparities of all pixels. After finding the correspondence points using local features, the disparities of the matched points are propagated to their neighbors [17]. Alternatively, the matched points are treated as seeds for an iterative estimation of the depth maps [18]. These seed-and-grow algorithms show good results and reduce the computational effort required by global approaches. However, these methods also provide a limited disparity map and tend to perform poorly on images that contain repeated patterns or textureless regions.

In this paper, we propose a fast and robust 3D terrain reconstruction algorithm suitable for construction site terrain reconstruction. In the finding correspondence point step, we propose a matching method that combines bucket-based fast corner detection and multi-scale, descriptor-based robust and fast feature matching specifically to reconstruct textureless environments with complex shapes. We also propose a two-stage cascade matching method to improve MSD matching efficiency. During the dense 3D reconstruction process, we infer the important disparity value with computation efficiency by proposing a probabilistic model.

2.2. Proposed framework

The proposed reconstruction algorithm can be divided into four major steps: interest point detection, descriptor generation, feature matching, and dense 3D reconstruction. To detect the interest point, corner points are extracted at distinctive locations in the left image. When selecting corner points, the distribution of the corner significantly affects the result of the dense 3D reconstruction. In order to resolve this problem, the proposed method independently extracts the same number of corner points from each sub-region to force the distribution of corner points to be uniform. To generate the descriptor, we use MSD, which consists of three descriptors at different scales in order to robustly find correspondence points in construction site terrain.

One popular way to search for correspondence points between two rectified images is to compare all computed descriptors in the first image to all other calculated descriptors in the second image. This linear search has quadratic computational complexity. Furthermore, the dimension of the descriptor has a direct impact on the matching time. Therefore, lower dimensions are more desirable for fast matching [22]. However, in order to improve the distinctiveness of the descriptor, MSD uses multi-scale information. Accordingly, this increases the dimension of the descriptor. Therefore, MSD requires a more efficient matching method. Thus, we propose an efficient matching method suitable for MSD.

The final step of feature-matching-based dense 3D reconstruction is the generation of dense disparity maps. This process is computationally expensive, because all pixels in the image must be matched to their corresponding points. We propose a probabilistic model for dense disparity map. The proposed model has two major advantages. The first advantage is that it significantly improves the computation efficiency by narrowing the search range using coarsely inferred disparity values from a precisely calculated triangle mesh. The second advantage to the proposed model is that its disparity values are more robust. The proposed model estimates the optimal disparity values by comparing robust descriptor vectors for each pixel. It is hence possible to improve the precision of the disparity values (Fig. 2).

3. Finding correspondence points

3.1. Corner detection

Interest points are extracted from distinctive locations in the image. The most widely used detector is the Harris corner detector [19], which is based on the eigenvalues of the second moment matrix. Rosten and Drummond [20] presented a high-speed corner detector called the Features from Accelerated Segment Test (FAST). This detector compared the pixel value at the center of a discretized circle around a candidate point in order to avoid costly window or filter operations. A candidate point is identified as a corner if there exists a contiguous arc of at least nine pixels. The FAST detector was further accelerated with machine learning techniques. The method uses a trained decision tree to classify whether a candidate pixel is a corner or non-corner. This FAST corner detector has shown good performance and high computational efficiency. Therefore, we also extract corners in the left image using the FAST detector.

The proposed reconstruction algorithm utilizes a triangle mesh for dense 3D reconstruction that is built from the closest matched point (see Section 4). If the distribution of the matched point cloud is irregular, regions of the cloud that include many matched points can describe the corresponding terrain in detail. However, the regions that contain only a few matched points are reconstructed coarsely because of the lower amount of ground information (see Fig. 3). This can be one of the causes of 3D reconstruction robustness and precision problems.

In order to overcome this problem, in the proposed method, the image plane is divided into small, regular, square sub-regions, and the same number of corner points is independently extracted from each

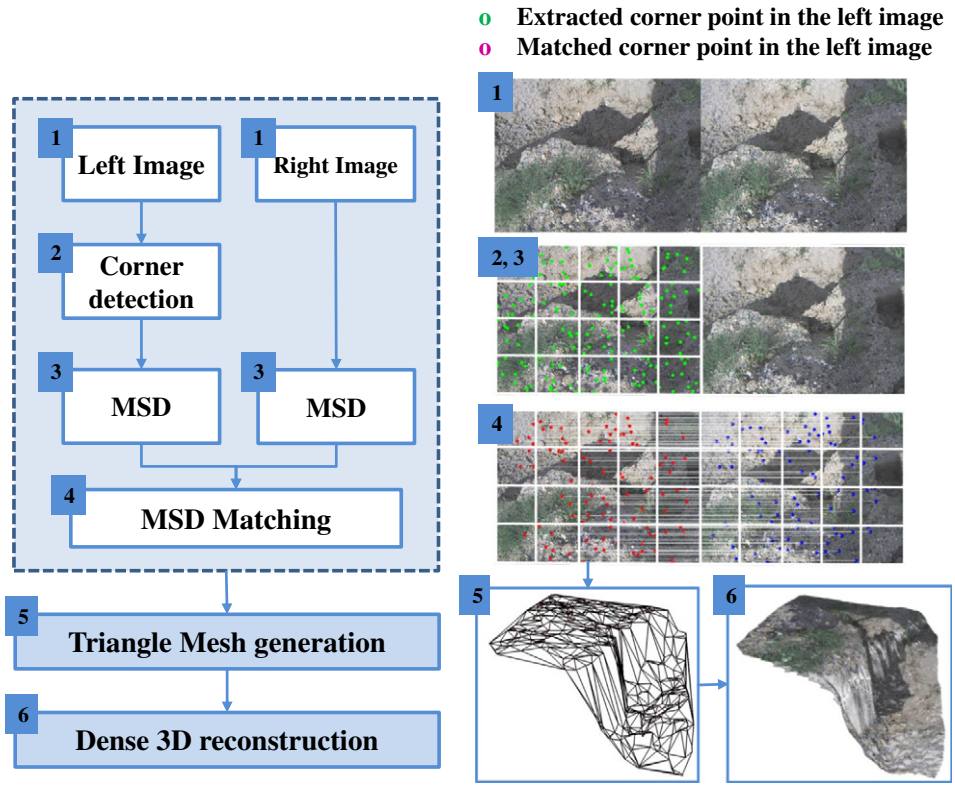

Fig. 2. Overview of the proposed dense 3D reconstruction system.

sub-region. This forces the distribution of corner points to be uniform, and the triangle mesh is created evenly over the image.

3.2. Multi-scale descriptor (MSD)

When finding correspondence points, the descriptor is a particularly important component because it significantly affects the precision of local feature matching. Furthermore, the computation of the descriptor is one of the most time-consuming matching tasks. Therefore, determining a descriptor that is suitable for challenging environments is a crucial task. Descriptor performance is generally evaluated by precision [14]. Popular local feature descriptors such as SIFT and SURF already provide high precision in many practical applications. However, for real practical tasks, computational load is another requirement to consider. To achieve robust and fast 3D construction site terrain reconstruction, a different type of descriptor is needed. Recently, MSD was developed for vision applications in challenging outdoor environments. This descriptor combines multi-scale gradient information and integral images for descriptor distinctiveness and low computational complexity. In our proposed algorithm, correspondence points are found using MSD because of its good performance.

In order to keep this paper self-contained, we only briefly explain the concept of MSD. MSD is computed based on the sum of the gradients over multiple scales. By describing different scales of the same corner point, the representation of its characteristics is improved. As shown in Fig. 4, MSD consists of three pre-defined scale descriptors (s1,s2,s3) at each corner point. Each patch is divided regularly into smaller 3×3 square sub-regions around the selected interest point. The size of each patch is defined by scale factor $\Delta s < 1, (s_{n-1} = s_n \times \Delta s)$, where scale index $n = 1, 2, 3$, and s_n is the n-th scale.

Each sub-region has a 4D descriptor vector $\mathbf{v} = (\sum d_x, d_y, |d_x|, |d_y|)$ that is similar to the SURF descriptor because of its computational efficiency and good performance. Concatenating these descriptors for all 3×3 sub-regions at each scale, a descriptor vector of length 108 (4

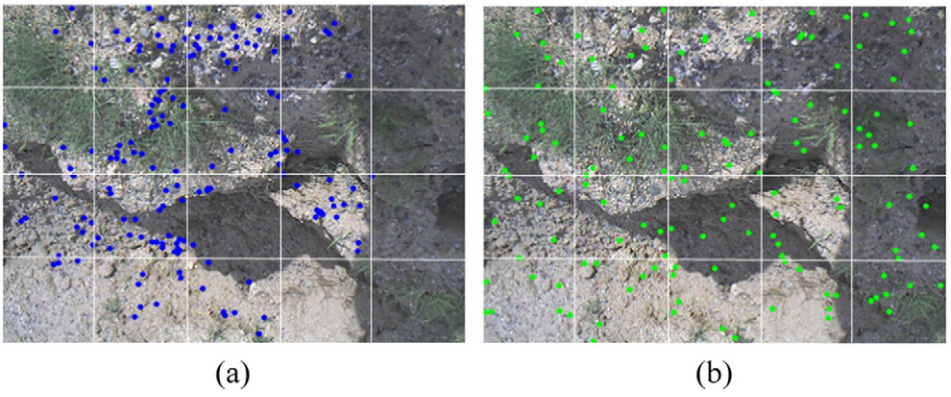

Fig. 3. Corner points extracted from equally divided sub-regions: (a) results of normal corner extraction (150 total corners), (b) results of bucket-based corner extraction (20 buckets × 7 corners per bucket = 140 total corners).

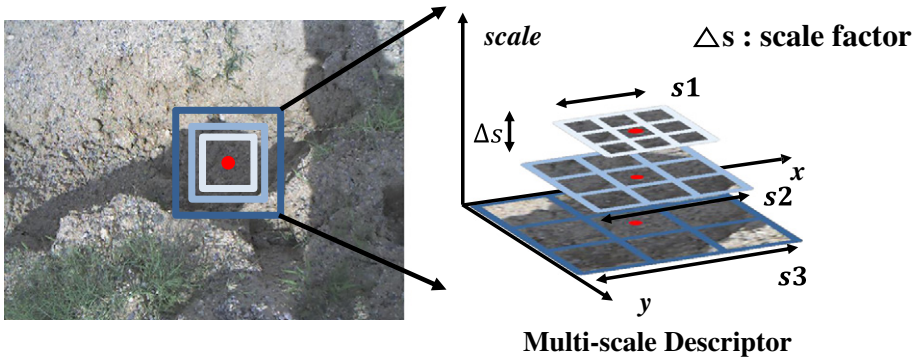

Fig. 4. MSD concept.

descriptor vectors × 9 sub-regions × 3 scales = 108) is obtained. In order to make it invariant to contrast, MSD is transformed to a unit vector.

Note that each MSD scale descriptor D_{s1}, D_{s2}, and D_{s3} has low distinctiveness. However, combining the different scale descriptors improves the representation of the corner point characteristics, as the MSDs repeatedly describe the same interest points at various scales. This makes MSD more robust than other descriptors.

Because MSD consists of multiple descriptors for each interest point, the computation of MSD is a time-consuming step. In order to significantly improve the computational speed, MSD is computed on integral images.

Each integral image is defined as follows.

$$I_1(\boldsymbol{x}) = \sum_{i=0}^{i\le width}\sum_{j=0}^{j\le height} d_x(i,j)\; I_2(\boldsymbol{x}) = \sum_{i=0}^{i\le width}\sum_{j=0}^{j\le height} |d_x(i,j)|$$
$$I_3(\boldsymbol{x}) = \sum_{i=0}^{i\le width}\sum_{j=0}^{j\le height} d_y(i,j)\; I_4(\boldsymbol{x}) = \sum_{i=0}^{i\le width}\sum_{j=0}^{j\le height} |d_y(i,j)|. \quad (1)$$

The value of integral image $I_n(\mathbf{x})$ at location $\mathbf{x}=\{x,y\}$ represents the sum of the gradient and absolute gradient within a rectangular region formed by the origin and x. The values $d_x(i,j)=I(i+1,j)-I(i-1,j)$ and $d_y(i,j)=I(i,j+1)-I(i,j-1)$ represent the intensity gradients at (i,j) in the horizontal and vertical directions, respectively. Here, $I(i,j)$ represents the image intensity at (i,j).

Note that once the integral image has been computed, it takes three additions to calculate the sum of the intensities over any rectangular area. Hence, its calculation time is independent of size. This significantly improves the computational efficiency of MSD.

After the four integral images are computed, MSD is computed from nine sub-regions at three different scales as follows.

$$D_{sn}(4m+k-4) = I_k(sn_{l,t}(m)) - I_k(sn_{l,b}(m)) - I_k(sn_{r,t}(m)) + I_k(sn_{r,b}(m)) \quad (2)$$

where scale index $n=1,2,3$, subregion index $m=1,2,\ldots,9$, and integral image index $k=1,2,3,4$. The descriptor at the sn-th scale is denoted by D_{sn}. Here, $sn_{l,t}(m)$ and $sn_{r,b}(m)$ represent the top-left and bottom-right coordinates of a sub-region $sn(m)$, respectively. Therefore, each descriptor vector $\mathbf{v}=(\sum d_x, \sum d_y, \sum|d_x|, \sum|d_y|)$ is calculated based on each integral image I_1, I_2, I_3, and I_4 respectively. This overall procedure is shown in Fig. 5. For further details, readers are referred to [13].

3.3. Acceleration of MSD matching

To achieve robust and fast MSD based matching, we propose a two-stage cascade matching method that combines single-scale coarse

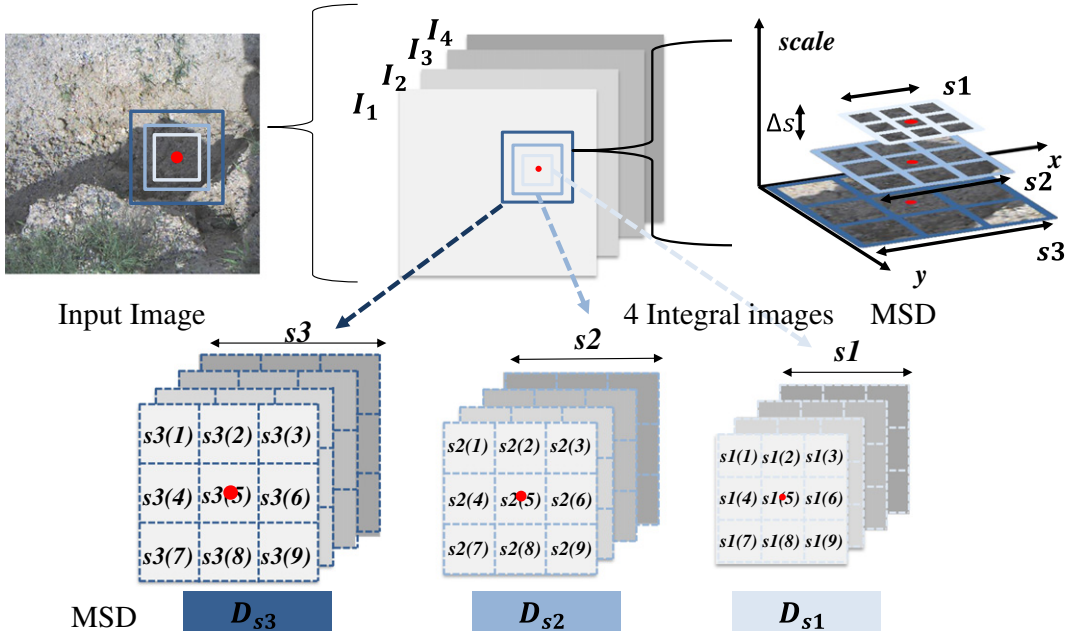

Fig. 5. MSD, based on the sum of the gradients over multiple scales of four integral images.

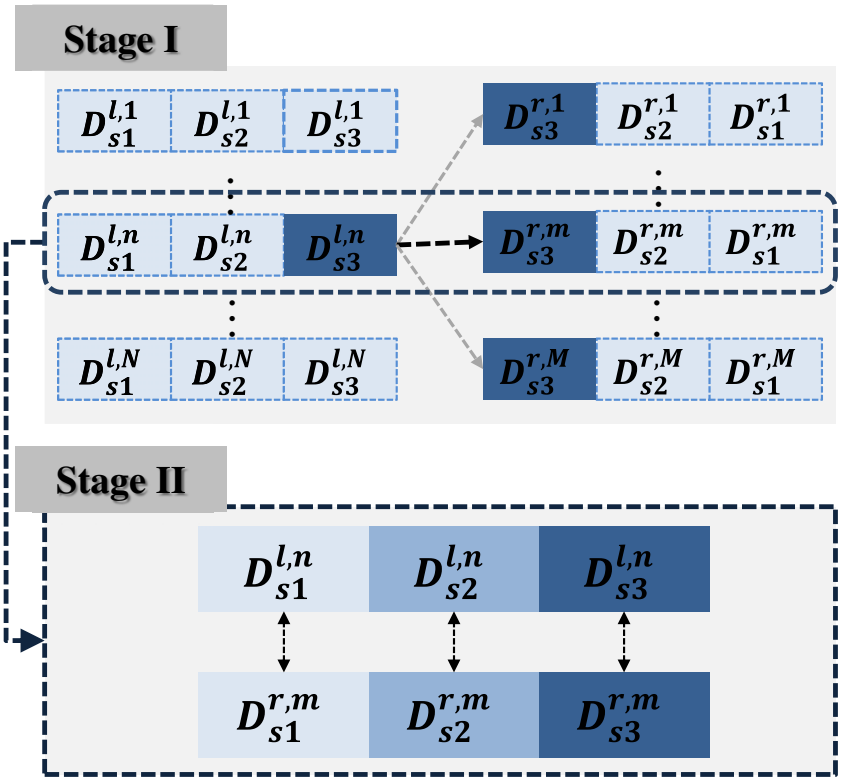

Fig. 6. Overview of the proposed two-stage cascade matching method. Stage I is a coarse single-scale descriptor matching, and Stage II is a fine three-scale descriptor matching.

matching and multiple-scale fine matching. We empirically determined that a larger patch size tends to achieve a higher recognition rate for a single-scale descriptor. We hence used the largest patch size D_{s3} for the first stage. As shown in Fig. 6, in the first stage, we coarsely find candidate correspondence points with the single scale descriptor D_{s3} (36 vectors) using a Euclidean distance for fast dissimilar descriptor

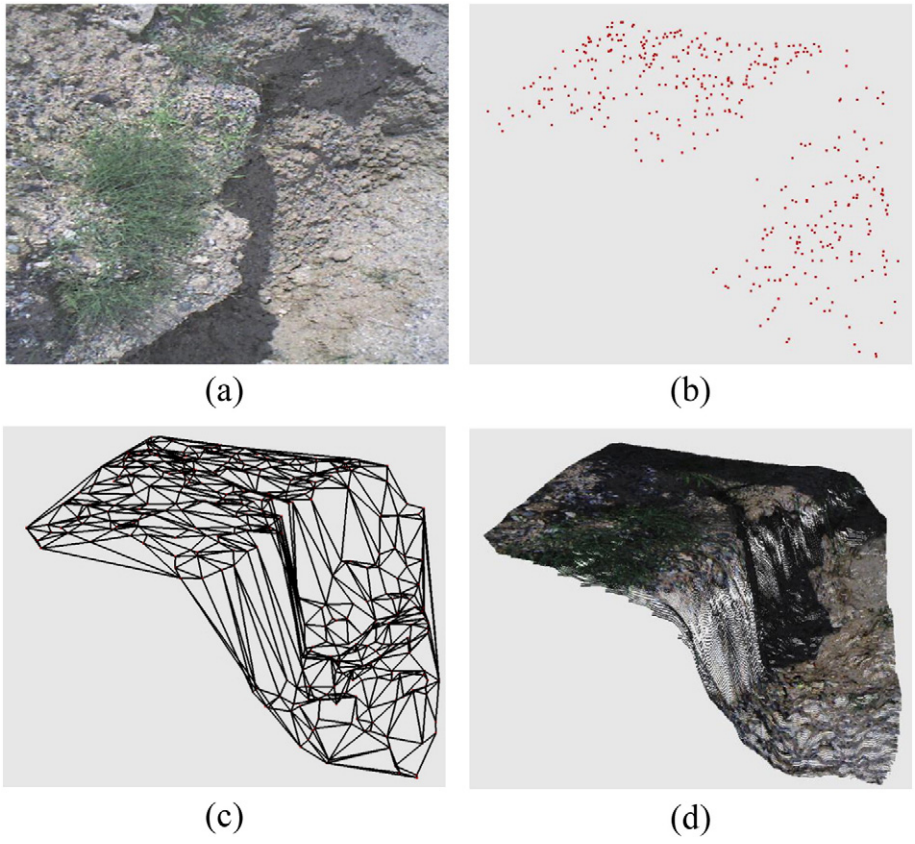

Fig. 7. Overview of dense 3D reconstruction: (a) input image, (c) sparse 3D point cloud, (c) triangle mesh, and (d) dense 3D reconstruction.

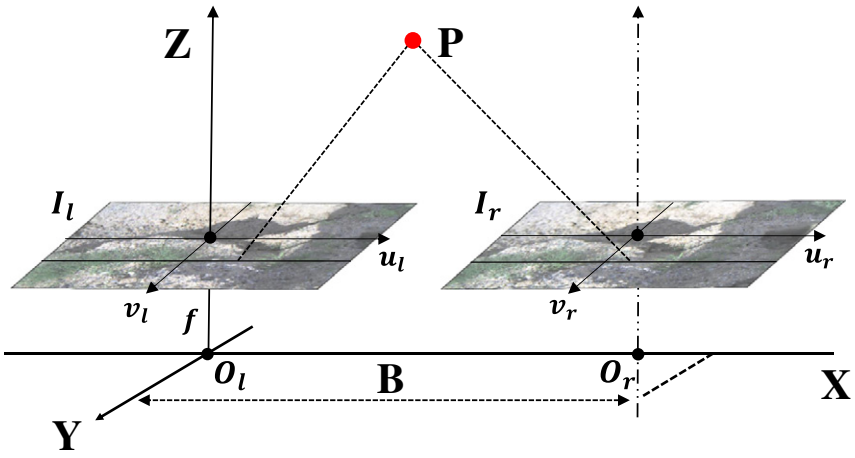

Fig. 8. Stereo camera geometry and triangulation.

filtering. Here, $D_{s3}^{l,n}$ and $D_{s3}^{r,m}$ represent the third scale (s3) descriptor of the n-th interest point in the left image and m-th interest point in the right image, respectively. The second stage then computes the Euclidean distance between MSD vectors D_{s1}, D_{s2}, and D_{s3} (108 vectors) selected from the results of the first matching stage to find correspondence points precisely. After completing the two main matching stages, if the distance of certain correspondence points is smaller than a predefined threshold *th*, the correspondence points are declared to be matched.

Note that the cascade matching method improves the matching efficiency significantly without reducing performance. This is because we filter correspondence point candidates quickly by using a lower dimension descriptor (one scale, 36 vectors) instead of a higher dimension descriptor (three scales, 108 vectors) and then find correspondence points precisely by comparing three different scale descriptors.

In order to eliminate mismatched points, we also check for consistency. Correspondence points are kept only if left-to-right matched points and right-to-left matched points are consistent.

4. Dense 3D terrain reconstruction

The proposed dense 3D reconstruction method consists of three major steps. The first step is sparse 3D reconstruction using a calibrated stereo rig, and the second step generates a triangle mesh in 3D space. Finally, dense 3D terrain is obtained using a probabilistic 3D model. Fig. 7 shows the steps of this process.

4.1. Sparse 3D point cloud and triangle mesh

In order to reconstruct 3D points from a stereo camera, it is essential to calibrate and rectify the stereo image. Calibrating the camera involves estimating the intrinsic and extrinsic parameters of each camera. Intrinsic parameters are related with the optical characteristics of the camera, such as the principle point and focal length, and extrinsic parameters represent the location of the each camera with global coordinates, such as its rotation and translation. After estimating the intrinsic and extrinsic parameters, a point in the left image searches for its corresponding point in the right image along an epipolar line. Once epipolar lines are aligned and parallel, the corresponding point lies on the same horizontal scan line. Thus, it is possible to find corresponding points more rapidly. This process is called image rectification. In this paper, OpenCV [21] was used to calibrate and rectify the stereo images. OpenCV is a popular open library, and it was selected because of its superlative performance.

We reconstruct 3D points via triangulation using the calibration parameters of the stereo camera rig with the rectified stereo image (see Fig. 8).

When O_l and O_r are the left and right camera centers respectively, we can define the camera stereo geometry and parameters as follows:

- homogeneous image coordinates $\boldsymbol{x} = (u, v, 1)^T$
- camera focal length f
- 3D point coordinates $P = (X, Y, Z)^T$
- baseline B.

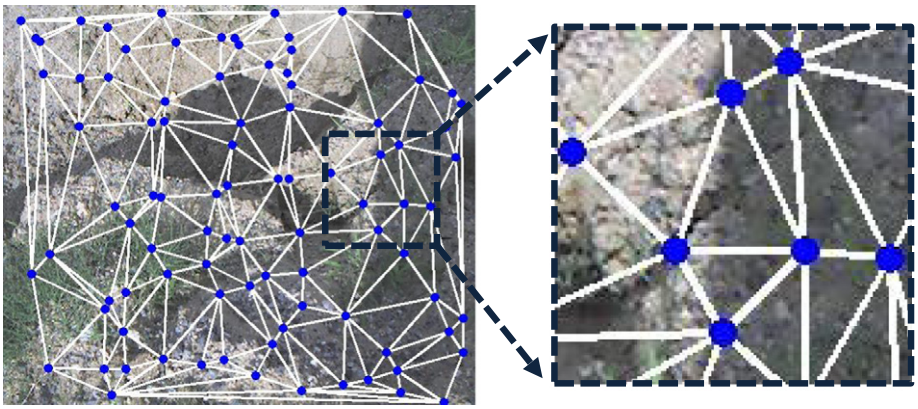

Fig. 9. 2D triangular using matched point cloud with Delaunay triangulation.

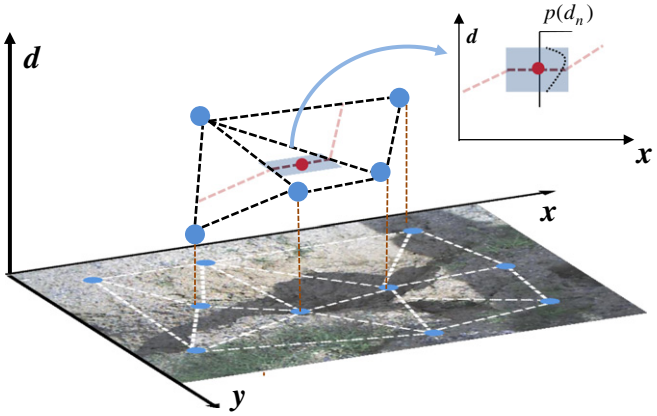

Fig. 10. Proposed probabilistic model inferred disparity values for every pixel using MAP estimation.

With these defined camera parameters, the 3D points are estimated as follows:

$$Z = f\frac{B}{u_l - u_r} = f\frac{B}{d}, \mathrm{X} = u_l\frac{B}{d}, \mathrm{Y} = v_l\frac{B}{d} \tag{3}$$

where d denotes the disparity, which refers to the difference between a pair of corresponding points, u_l-u_r. After determining the corresponding points, we can estimate the disparity value of each matched pair using Eq. (3). Once we have calculated the sparse 3D point cloud, we then compute the triangle mesh.

To construct the triangle mesh in 3D coordinates, as shown in Fig. 9, we first build 2D triangles using the matched point cloud via Delaunay triangulation. We then calculate the triangle mesh parameter $Tr(\mathrm{x}_l^n)$ by

$$Tr_i(\mathrm{x}_l^n) = a_i u_n + b_i v_n + c_i \tag{4}$$

where i is the index of triangle that contains the pixel $\mathrm{x}_l^n = (u_n, v_n)$. For each triangle with three 3D points, we can obtain the plane parameters (a_i, b_i, c_i) by solving a linear equation using the three 3D vertexes of the triangle. Note that on construction sites, we can generally assume that the ground consists of continuous surfaces. Hence, small areas of ground can approximate a plane. Therefore, constructing 2D triangles with robustly matched near corner points represents the ground more robustly, particularly for textureless ground patterns. Furthermore, the estimated triangle mesh provides precise initial disparity values for our proposed dense 3D reconstruction model. This significantly reduces the computational effort by narrowing the search range of each pixel in the input image.

After building the triangle mesh with the sparse 3D point cloud, we then estimate the dense 3D terrain using the proposed probabilistic model.

4.2. Probabilistic model for dense 3D reconstruction

We now describe our probabilistic model for dense 3D terrain reconstruction (Fig. 10).

Given the stereo images and triangle mesh formed from the three 3D points, we can estimate the optimal disparity value of each pixel using maximum a posteriori (MAP) estimation as follows:

$$\hat{d}_n = \operatorname{argmax} \mathrm{p}\left(d_n | Tr, x_l^n, x_r^1, \ldots, x_r^M\right) \tag{5}$$

where $\mathrm{x}_r^1, \ldots, \mathrm{x}_r^M$ indicate all pixels on the right image that are on the epipolar line x_l^n. The posterior is factorized with the prior and likelihood as follows:

$$p\left(d_n | Tr, x_l^n, x_r^1, \ldots, x_r^M\right) \propto p\left(d_n | Tr, x_l^n\right) p\left(x_r^1, \ldots, x_r^M | x_l^n, d_n\right). \tag{6}$$

We define the prior term to be a Gaussian model based on disparity values coarsely inferred from the triangle mesh.

$$p\left(d_n | Tr, x_l^n\right) \propto \begin{cases} \exp\left(-\dfrac{\left(d_n - Tr\left(x_l^n\right)\right)^2}{2\sigma^2}\right) & \text{if } \left|d_n - Tr\left(x_l^n\right)\right| < 3\sigma \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $Tr(\mathrm{x}_l^n)$ is the estimated triangle mesh containing pixel $\mathbf{x}_l^n = (u_n, v_n)$. We take the likelihood term to be the Euclidean distance between descriptor vectors as follows.

$$p\left(x_r^m | x_l^n, d_n\right) \propto \begin{cases} -\dfrac{1}{\left\|D_l^n - D_r^m\right\|} & \text{if } \begin{pmatrix} u_l^n \\ v_l^n \end{pmatrix} = \begin{pmatrix} u_r^m + d_n \\ v_r^m \end{pmatrix} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where D_l^n and D_r^m represent n-th left and m-th right point descriptor vectors. We also use the stereo constraint that states that corresponding points can only exist on the same epipolar line. Therefore, the if-condition in Eq. (8) ensures that corresponding points are found on

Fig. 11. Experimental environment.

Fig. 12. Stereo camera system mounted on an excavator.

the same epipolar line. We can model the distribution of the factorized likelihood term as follows:

$$p(x_r^1, \ldots, x_r^M | x_l^n, d_n) \propto \sum_{i=1}^{M} p\left(x_r^i | x_l^n, d_n\right). \quad (9)$$

Note that given the proposed prior and likelihood terms, the proposed method increases not only the computational speed but also the precision of the estimated disparity value. This can be explained by the fact that the disparity value is estimated only if $|d_n - Tr(x_l^n)| < 3\sigma$. This reduces the computational burden dramatically by narrowing the search range. The optimal disparity value is determined from a good initial value that is obtained from the precisely calculated triangle mesh.

5. Experiments and analysis

In this section, we compare the proposed algorithm, MSD-based dense 3D reconstruction (MSD DR), with other state-of-the-art dense 3D reconstruction algorithms. To evaluate the algorithms in a more practical situation, we mounted a stereo camera system on an excavator to obtain test images from a real construction environment (see in Fig. 11). The camera setup on the excavator is shown in Fig. 12. We used Flea3 USB 3.0 cameras (Point Grey), because they perform well and offer an effective interface, making them suitable for building stereo systems on construction equipment. The detailed model specifications are provided in Table 1. We used four test images, each of which captured different ground shapes and materials. Image I (gravel and slop), Image II (soil and slop), Image III (mud and hole), and Image IV (mud and bumps) are depicted in Fig. 13. This test set was captured in a complex environment with different solid materials to evaluate various algorithms in more practical conditions. Therefore, we believe that this dataset adequately reflects the variety of stereo camera-based 3D reconstruction problems at construction sites.

In order to analyze the benefits of the proposed method, we compared the proposed algorithm with three different methods: block matching (BM), semi-global block matching (SGBM), and SURF descriptor-based dense 3D reconstruction (SURF DR). We chose these methods because of their good performance and similar properties to those of MSD [26].

To fairly evaluate the performance of the SURF DR, we replaced the MSD descriptor with a SURF descriptor while keeping the other parts of the proposed method such as corner detection, feature matching, and dense 3D reconstruction the same. The algorithms were implemented in VC++ using the latest Open-source Computer Vision (OpenCV) library [21]. In addition, we used the latest BM, SGBM, and SURF implementations provided by OpenCV. All evaluations were run on a PC with a 2.6 GHz processor and 8 GB memory. We only used a single core.

5.1. Experimental results

In this subsection, we show the results of the three major steps of the proposed algorithm for the four test images. As can be seen in Fig. 14, each step of the proposed method reconstructs the actual construction site well. The shape of the reconstructed ground at each step is similar to the actual ground appearance. Note that, as can be seen in the second column of Fig. 14, dense 3D reconstruction is built from the sparse 3D points of the matched point cloud. In the proposed algorithm, the correspondence points are found using MSD, and its three different scale descriptors have robustly and precisely computed the point cloud. This positively affects the results of the dense 3D reconstruction.

MSD DR infers the dense 3D point cloud using the triangle mesh as the initial depth, assuming that construction site terrain consists of continuous surfaces. In order to highlight the effects of using this triangle mesh, we show the result of dense 3D reconstruction from varying viewpoints in Fig. 15. The results of the reconstructed 3D surfaces are very similar to the real ground appearance. This estimated ground surface has less noise than other methods such as BM and SGBM (see Figs. 16 and 17).

Table 1
Technical data for Flea3 (used in experiments).

	Model specification
Readout method	Global shutter
Interface	USB 3.0
Frame rate	60 FPS
Resolution	1280 × 1024

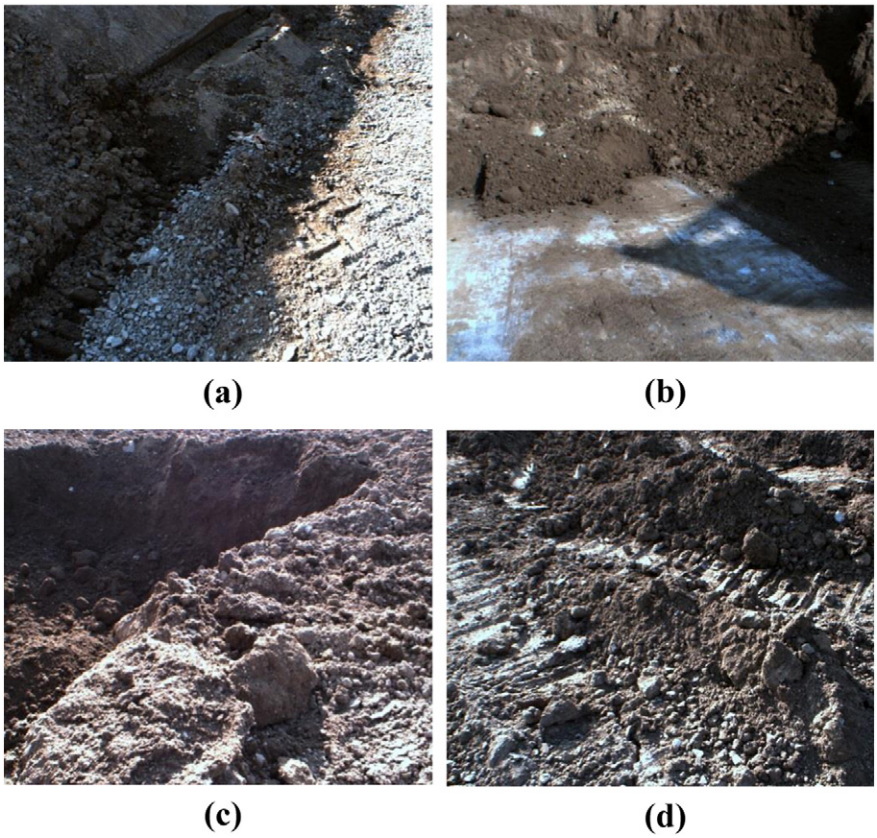

Fig. 13. Test images, (a) Image I (gravel and slop), (b) Image II (soil and slop), (c) Image III (mud and hole) and (d) Image IV (mud and bumps).

To objectively evaluate MSD DR, we compared it with three different methods: BM, SGBM, and SURF DR. As can be seen in Fig. 16, the proposed reconstruction method outperformed BM and SGBM for all test images. MSD DR and SURF DR both calculated dense 3D point clouds similar to the real ground shape. However, the results of BM and SGBM showed very noisy 3D point clouds. This is because BM and

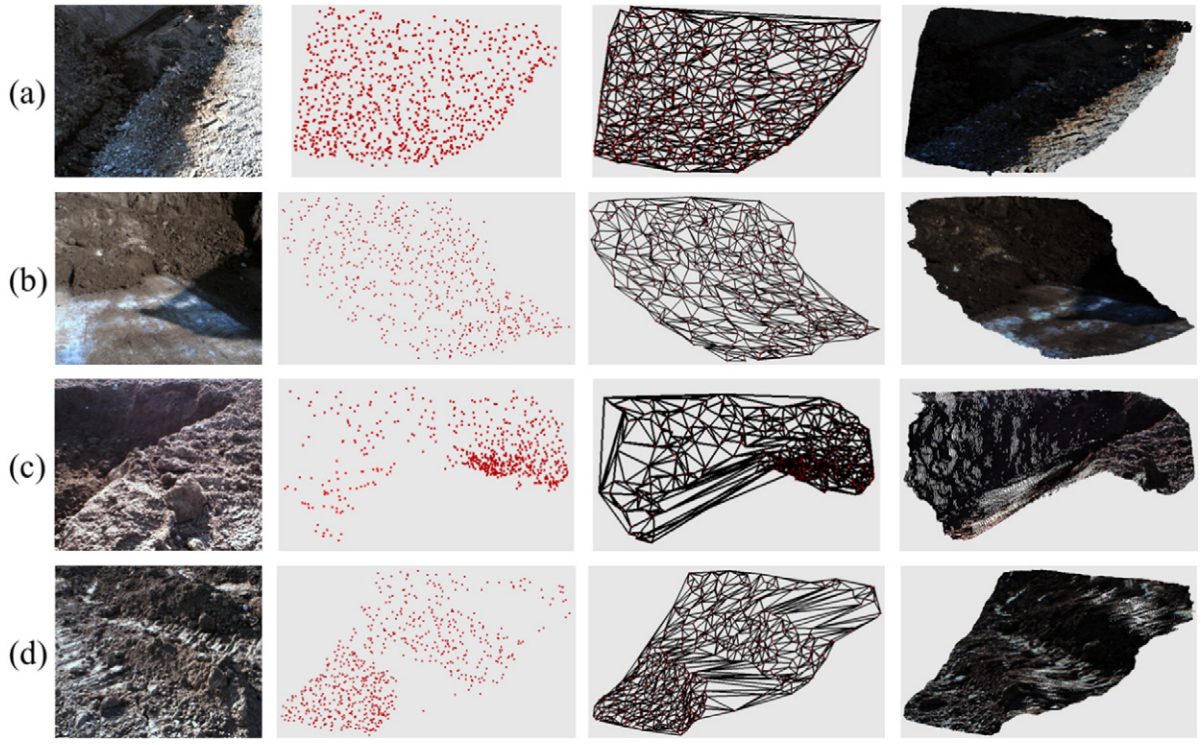

Fig. 14. Results of the proposed dense 3D reconstruction method: original images (column 1), sparse 3D reconstruction (column 2), triangle mesh (column 3), and dense reconstruction (column 4).

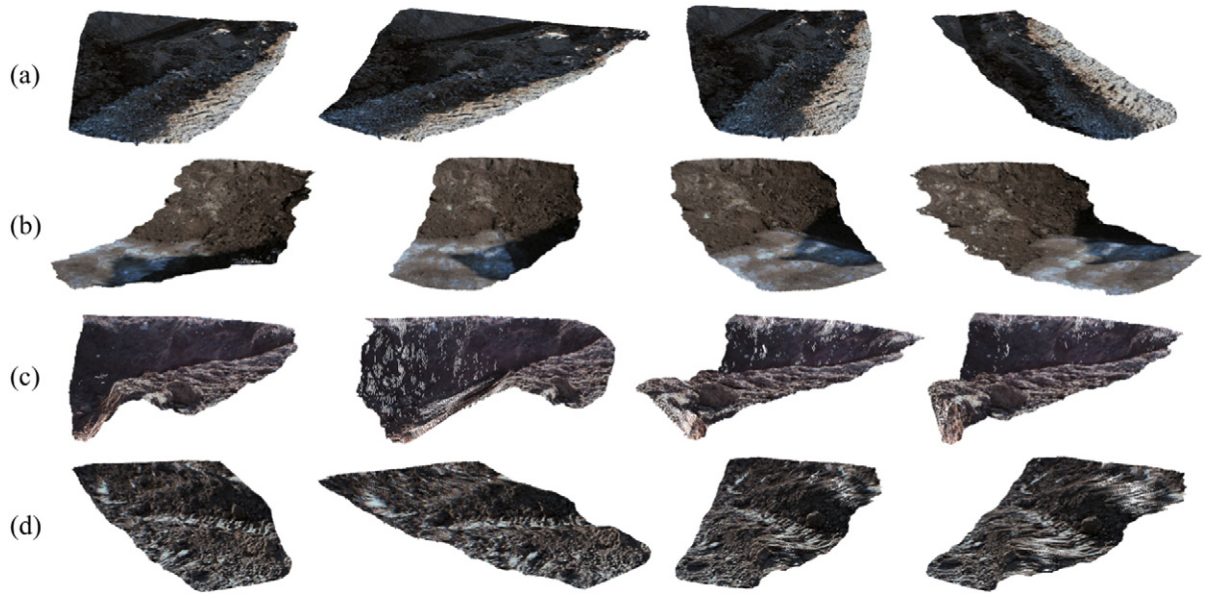

Fig. 15. Dense 3D reconstruction from various viewpoints for each test image using the proposed model.

SGBM basically determine the cost value between correspondence points by comparing pixel intensity. This kind of patch-based descriptor is less robust than other descriptors such as SIFT, SURF, and MSD, increasing correspondence point mismatches and affecting the result of dense 3D reconstruction. However, in the proposed algorithm, the triangle mesh that is computed from the result of robustly matched points provides good initial depth for the dense 3D reconstruction model. This improves the robustness of the 3D point cloud compared with patch-based dense reconstruction methods such as BM and SGBM. Note that the results of SURF DR and MSD DR are similar, but the computation time of MSD DR is 31 times faster than SURF DR, as detailed in Section 5.2.

5.2. Computation time

In practical automated construction applications, computation time is another important property. Therefore, we also evaluated the calculation time of each method.

As expected, although their results are too noisy to use, the patch-based dense reconstruction algorithms BM and SGBM showed faster

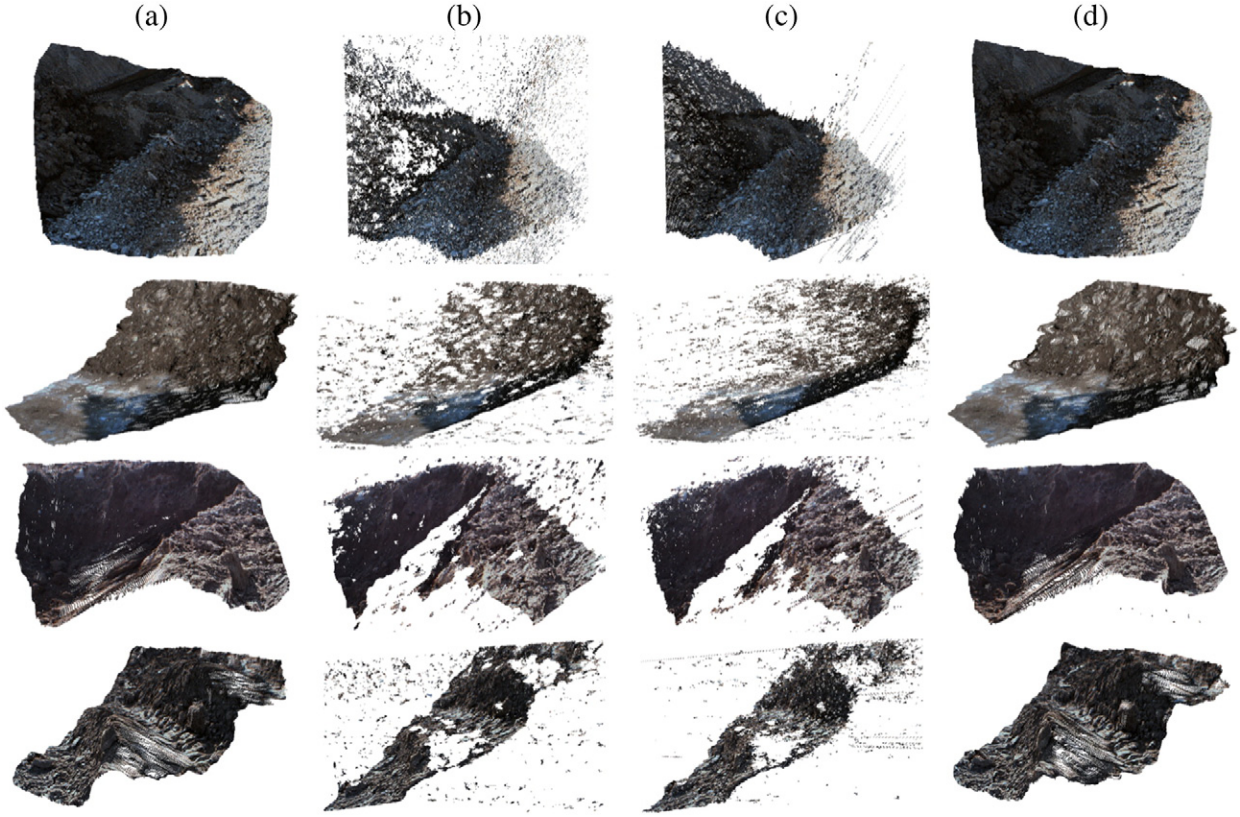

Fig. 16. Results of various dense 3D reconstruction algorithms: (a) MSD DR, (b) BM, (c) SGBM, and (d) SURF DR.

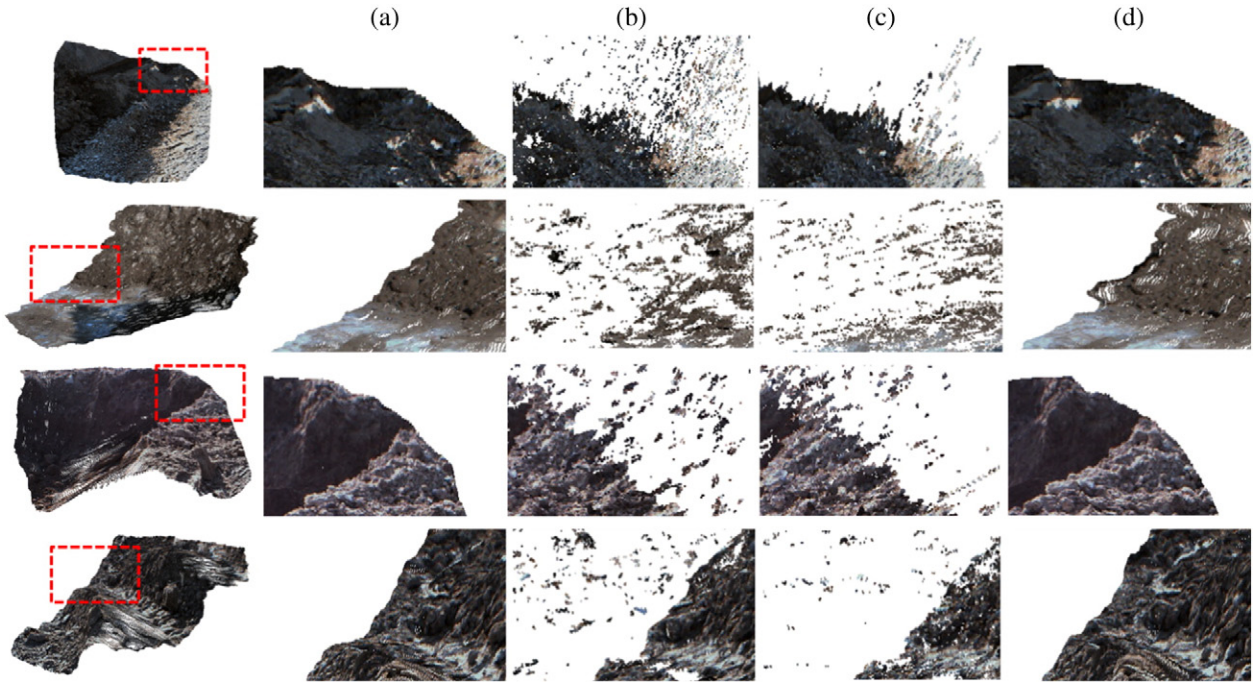

Fig. 17. Enlarged results (red rectangle) of the dense 3D reconstruction for various methods: (a) MSD DR, (b) BM, (c) SGBM, and (d) SURF DR.

computation time than MSD DR and SURF DR. Of these two methods, the computation time of MSD DR is 31 times faster than that of SURF DR. The average processing time of MSD DR is about 750 ms. Note that, for the sake of safety at construction sites, operators of equipment such as excavators, wheel loaders, and forklifts are generally instructed to move equipment at under 15 km/h (4.1 m/s). Furthermore, automated construction equipment works slowly, due to the complexity of the control system. Therefore, it suffices to provide the reconstructed 3D terrain information to automated construction equipment each second (more than 1 frame/s). Hence, we believe that 750 ms is sufficiently fast for autonomous control applications.

In the dense 3D reconstruction process, the descriptor creation and matching step and the probabilistic model-based dense 3D reconstruction step are the most time-consuming tasks. Even though the proposed method estimates a good initial depth value from the triangle mesh, it requires high computational effort to infer a highly precise depth value for all of the pixels in an image. In general, most parts of a construction site consist of a combination of plane-like structures that could be approximated using a triangle mesh-based 3D reconstruction. By using triangular mesh-based DR (TM DR) for practical construction equipment applications, the computation effort of the original MSD DR is reduced by half. In that case, the computation time of TM DR is even faster than that of SGBM without sacrificing much of the original MSD DR reconstruction performance. We compare the computation time in more detail in Tables 2 and 3.

6. Concluding remarks

This paper presented a robust and fast dense 3D reconstruction algorithm that combines a precise matching process with a proposed dense 3D reconstruction model. Results on test images collected from a construction site show that the proposed algorithm provides good performance with low computational time. Even though construction site terrain is difficult to reconstruct because of its complicated environmental conditions, the obtained results demonstrate that MSD-based dense 3D reconstruction is suitable for various autonomous control applications where computation time and precision are essential.

In future work, we shall apply the proposed algorithm to a semi-automatic construction equipment system. Our proposed algorithm will provide detailed environmental conditions to the construction equipment, in order to identify changes to its operating radius. It is possible to calculate the distance between the construction equipment and the target ground location as the initial position information for automatic ground digging or solid loading tasks. Furthermore, the proposed algorithm can measure the local ground around current semi-automatic construction equipment, making it unnecessary to use expensive surveying instruments, such as total stations, to estimate the ground conditions. The proposed 3D reconstruction algorithm can be utilized as a simple and cheap attachment for local surveys of the terrain around

Table 2
Computation time per image (ms).

Dataset		MSD DR	SURF DR
Image I	ACET	9.2	9.8
	AD & MT	349.6	17,189.3
	ATPT	10.6	10.9
	ADRT	398.0	7108.2
	APT	767.5	24,318.3
Image II	ACET	9.8	9.1
	AD & MT	338.3	17,122.0
	ATPT	9.2	9.4
	ADRT	408.5	7402.2
	APT	765.8	24,542.7
Image III	ACET	8.9	9.5
	AD & MT	318.3	17,070.7
	ATPT	8.5	9.5
	ADRT	392.9	7586.6
	APT	728.6	24,676.2
Image IV	ACET	10.6	9.1
	AD & MT	346.3	17,362.2
	ATPT	9.9	9.5
	ADRT	386.5	7130.2
	APT	753.3	24,511.0

ACET: Average Corner Extraction Time.
AD & MT: Average Descriptor Computation and Matching Time.
ATPT: Average Triangular Plane Generation Time.
ADRT: Average Dense Reconstruction Time.
APT: Average Processing Time.

Table 3
Computation time per image (ms).

Dataset	BM	SGBM	SURF DR	MSD DR	TM DR
Image I	198.6	429.8	24,318.3	767.5	369.4
Image II	201.7	415.2	24,542.7	765.8	357.3
Image III	201.0	423.2	24,676.2	728.6	335.7
Image IV	199.5	425.9	24,510.9	753.3	366.8

semi-automatic equipment with various semi-automatic construction applications.

Appendix A. Supplementary data

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.autcon.2015.12.022.

References

[1] Y. Arayici, An approach for real world data modelling with the 3D terrestrial laser scanner for built environment, Autom. Constr. 16 (6) (2007) 816–829, http://dx.doi.org/10.1016/j.autcon.2007.02.008.
[2] J.-C. Du, H.-C. Teng, 3D laser scanning and GPS technology for landslide earthwork volume estimation, Autom. Constr. 16 (5) (2007) 657–663, http://dx.doi.org/10.1016/j.autcon.2006.11.002.
[3] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: real-time single camera SLAM, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 1052–1067, http://dx.doi.org/10.1109/tpami.2007.1049.
[4] G.J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla, Segmentation and recognition using structure from motion point clouds, Computer Vision — ECCV 2008 2008, pp. 44–57, http://dx.doi.org/10.1007/978-3-540-88682-2_5.
[5] M. Golparvar-Fard, J. Bohn, J. Teizer, S. Savarese, F. Peña-Mora, Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques, Autom. Constr. 20 (8) (2011) 1143–1155, http://dx.doi.org/10.1016/j.autcon.2011.04.016.
[6] V. Kolmogorov, R. Zabih, Computing visual correspondence with occlusions using graph cuts, Proceedings Eighth IEEE International Conference on Computer Vision, 2001http://dx.doi.org/10.1109/iccv.2001.937668.
[7] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient belief propagation for early vision, Int. J. Comput. Vis. 70 (1) (2006) 41–54, http://dx.doi.org/10.1007/s11263-006-7899-4.
[8] H. Hirschmuller, Stereo processing by semiglobal matching and mutual information, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 328–341, http://dx.doi.org/10.1109/tpami.2007.1166.
[9] J. Cech, R. Sara, Efficient sampling of disparity space for fast and accurate matching, 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007http://dx.doi.org/10.1109/cvpr.2007.383355.
[10] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110, http://dx.doi.org/10.1023/b:visi.0000029664.99615.94.
[11] H. Bay, T. Tuytelaars, L. Van Gool, SURF: speeded up robust features, Lecture Notes in Computer Science (2006) 404–417, http://dx.doi.org/10.1007/11744023_32.
[12] G. Le Besnerais, M. Sanfourche, F. Champagnat, Dense height map estimation from oblique aerial image sequences, Comput. Vis. Image Underst. 109 (2) (2008) 204–225, http://dx.doi.org/10.1016/j.cviu.2007.07.003.
[13] C. Sung, M.J. Chung, Multi-scale descriptor for robust and fast camera motion estimation, IEEE Signal Process Lett. 20 (7) (2013) 725–728, http://dx.doi.org/10.1109/lsp.2013.2264672.
[14] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings, 2003http://dx.doi.org/10.1109/cvpr.2003.1211478.
[15] J. Heinly, E. Dunn, J.-M. Frahm, Comparative evaluation of binary features, Lect. Notes Comput. Sci (2012) 759–773, http://dx.doi.org/10.1007/978-3-642-33709-3_54.
[16] O. Veksler, Stereo correspondence by dynamic programming on a tree, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005http://dx.doi.org/10.1109/cvpr.2005.334.
[17] J. Yao, W.-K. Cham, 3D modeling and rendering from multiple wide-baseline images by match propagation, Signal Process. Image Commun. 21 (6) (2006) 506–518, http://dx.doi.org/10.1016/j.image.2006.03.005.
[18] T. Strecha, V. Gool, Dense matching of multiple wide-baseline views, Proceedings Ninth IEEE International Conference on Computer Vision, 2003http://dx.doi.org/10.1109/iccv.2003.1238627.
[19] C. Harris, M. Stephens, A combined corner and edge detector, Procedings of the Alvey Vision Conference 1988, 1988http://dx.doi.org/10.5244/c.2.23.
[20] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, Lecture Notes in Computer Science (2006) 430–443, http://dx.doi.org/10.1007/11744023_34.
[21] G. Bradski, A. Kaehler, Learning OpenCV, O'Reilly Media Inc., 2008
[22] J.S. Beis, D.G. Lowe, Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997http://dx.doi.org/10.1109/cvpr.1997.609451.
[23] S. Gauglitz, T. Höllerer, M. Turk, Evaluation of interest point detectors and feature descriptors for visual tracking, Int. J. Comput. Vis. 94 (3) (2011) 335–360, http://dx.doi.org/10.1007/s11263-011-0431-5.
[24] M.-D. Yang, C.-F. Chao, K.-S. Huang, L.-Y. Lu, Y.-P. Chen, Image-based 3D scene reconstruction and exploration in augmented reality, Autom. Constr. 33 (2013) 48–60, http://dx.doi.org/10.1016/j.autcon.2012.09.017.
[25] I. Brilakis, H. Fathi, A. Rashidi, Progressive 3D reconstruction of infrastructure with videogrammetry, Autom. Constr. 20 (7) (2011) 884–895, http://dx.doi.org/10.1016/j.autcon.2011.03.005.
[26] H. Hirschmuller, Accurate and efficient stereo processing by semi-global matching and mutual information, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005http://dx.doi.org/10.1109/cvpr.2005.56.