IPSJ Transactions on Computer Vision and Applications Vol.5 60–64 (July 2013)

[DOI: 10.2197/ipsjtcva.5.60]

Express Paper

Direct Ground Surface Reconstruction from Stereo Images

Shigeki Sugimoto[1,a)] Takaaki Kato[1,b)] Kouma Motooka[1,c)] Masatoshi Okutomi[1,d)]

Received: March 11, 2013, Accepted: April 24, 2013, Released: July 29, 2013

Abstract: We propose a method for directly estimating a square grid ground surface from stereo images. We estimate the heights of all vertices in a square mesh, in which each square is divided into two triangular patches, drawn on a level plane of the ground, from a pair of images captured by nearly front-looking stereo cameras. We formulate a data term, representing the sum of the squared differences of photometrically transformed pixel values in homography-related projective triangular patches between the two stereo images, by the inverse compositional trick for both surface and photometric parameters for realizing an efficient estimation algorithm. The main difficulty of this problem formulation lies in the estimation instability for the heights of the distant vertices from the cameras, since the image projections of the distant triangular patches are crushed in the images. We effectively improve the stability by the combinational use of an additional smoothness term, update constraint term, and a hierarchical meshing approach. We demonstrate the validity of the proposed method through experiments using real images, and the usability for mobile robots by showing traversable area detection results on the ground surfaces estimated by the proposed method.

Keywords: ground surface estimation, square grid, stereo vision, direct image alignment

1. Introduction

3D reconstruction of a ground surface is one of the fundamental problems of mobile robotics, especially for the vehicles and robots traversing off-road environments. It is desirable that the ground 3D information be represented by surface model parameters, not by dense point clouds, not only to reduce the amount of 3D data, but also because of the importance of surface normals in robot action (e.g., a land rover requires surface normals for searching an almost level site to move safely).

A 3D surface can be recovered from point clouds by fitting a polygonal mesh or B-spline surface (e.g., Ref. [5]). Considered with the requirement for computational efficiency in robotics, a possible choice is to fit a surface model to the output of a recent fast dense stereo reconstruction method based on local and/or semi-global techniques (e.g., Refs. [4], [6], [10]). For ground surface reconstruction, however, it is preferable to use a global method since roads and off-roads often have weakly textured surfaces and repeated patterns (e.g., wheel tracks). More importantly, these methods implicitly assume that the target surfaces are nearly front-parallel for realizing fast cost aggregation. This assumption is completely corrupted for the ground surfaces observed from in-vehicle front-looking cameras, as clearly depicted in Ref. [11].

A more promising choice is to adopt a global method which directly estimates surface model parameters from stereo images [7], [8]. In this approach, depths of the mesh vertices drawn on a reference image are estimated by direct image alignment. When we apply this approach to a ground surface using front-looking cameras, however, the resultant surface mesh is quite irregular on the ground, with a detailed mesh on the surface near to the cameras and a rough mesh far from the cameras. Another important requirement from mobile robotics is to represent the ground by a regular square grid with level heights at all vertices, so-called a digital elevation map (DEM) [9], [12]. Since a DEM can directly provide a per-unit-length gradient at every edge, we can easily detect traversable paths on the ground in real-time. The edge-wise traversability information is also necessary for the existing path-planning algorithms including the A^*-search [3].

In this paper, we propose a method for directly reconstructing a ground surface with a regular square grid from stereo images. We estimate the heights of all vertices in a square mesh, composed of piecewise triangular patches, drawn on a level plane of the ground, from a pair of images captured by nearly front-looking stereo cameras. Our basic idea is to minimize a cost function, representing the sum of the squared differences of photometrically transformed pixel values in homography-related projective triangular patches between the two stereo images. For realizing high computational efficiency, the cost for computing update parameters is re-formulated by using the inverse compositional trick [1], [2] for both surface and photometric parameters.

The main difficulty of this problem formulation lies in the instability of the height estimation of the distant vertices from the camera. This is because the pixel numbers in the image projection triangles of the distant patches are too small to contribute to the vertex height measurements. Although an additional smooth-

[1] Tokyo Institute of Technology, Meguro, Tokyo 152–8550, Japan
[a)] shige@ok.ctrl.titech.ac.jp
[b)] taka@ok.ctrl.titech.ac.jp
[c)] kmotooka@ok.ctrl.titech.ac.jp
[d)] mxo@ctrl.titech.ac.jp

ness term somewhat improves the estimation stability, the use of only two terms cannot control *flaps* of the surface in the distant part. We show that the stability can be effectively improved by the combinational use of an additional update constraint term and a hierarchical meshing approach. We also demonstrate the usability of the proposed method for mobile robots by showing traversable area detection results on the estimated ground surfaces.

2. Preliminaries

The coordinate relationships of a *local*[*1] ground $\boldsymbol{x} = (x, y, z)^T$, a reference camera $\boldsymbol{x}_0 = (x_0, y_0, z_0)^T$, and the other camera $\boldsymbol{x}_1 = (x_1, y_1, z_1)^T$ are expressed by $\boldsymbol{x}_0 = \boldsymbol{R}\boldsymbol{x}+\boldsymbol{t}$ and $\boldsymbol{x}_1 = \boldsymbol{R}_s\boldsymbol{x}_0+\boldsymbol{t}_s$, as shown in **Fig. 1**, where $\boldsymbol{R}_s, \boldsymbol{t}_s, \boldsymbol{R}$, and $\boldsymbol{t}$ are assumed to be known. The z-axis of the local ground system is determined to be parallel to the gravity direction, which can be provided by in-vehicle accelerometers. Then the x-y plane of the local ground system can be set at an arbitrary z-position, which is enough to represent a ground surface relative to the camera position (but we prefer setting the plane as near as possible to the actual zero-level plane). We define $\boldsymbol{R}$ and $\boldsymbol{t}$ so that the origin of $\boldsymbol{x}$ is arranged at the intersection point between the x-y plane and the perpendicular line dropped from $\boldsymbol{x}_0 = \boldsymbol{0}$ to the x-y plane, and that the y-axis corresponds to the perpendicular projection of the z_0-axis on the x-y plane.

We set a 2D square grid, in which each square is divided into two triangles for generating a triangular mesh, on the x-y plane of the local ground system, as also shown in Fig. 1. The triangular mesh includes V vertices, whose 3D positions are specified by $\boldsymbol{x}_v = (x_v, y_v, z_v)^T, (v = 1, \cdots, V)$ where x_v, y_v are known, and N triangular patches $S_n, (n = 1, \cdots, N)$. Let $\boldsymbol{z} \equiv (z_1, z_2, \cdots, z_V)^T$ represent the surface parameter vector to be estimated.

Let $I_0[\boldsymbol{u}_0]$ and $I_1[\boldsymbol{u}_1]$ be the pixel values (gray levels) of the reference image I_0 and the other image I_1, respectively, where $\boldsymbol{u}_0 = (u_0, v_0)^T$ and $\boldsymbol{u}_1 = (u_1, v_1)^T$ respectively denote the corresponding points in I_0 and I_1. For avoiding complexity, let $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$ be in the *canonical* image configuration.

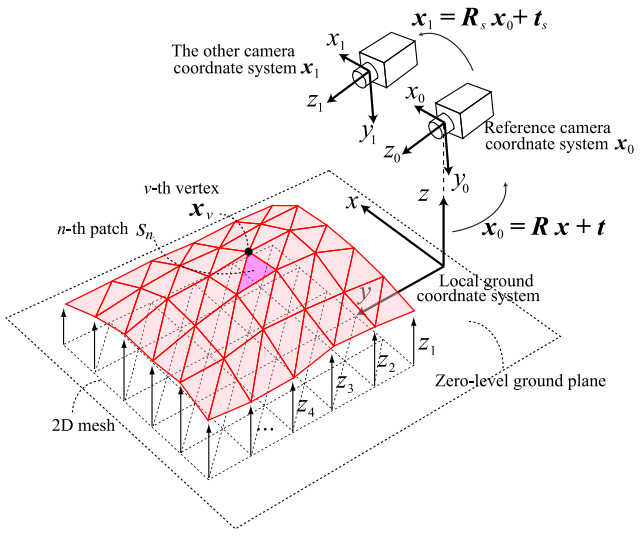


Fig. 1 Geometry relationship.

[*1] While the vehicle moves around, the local ground system also moves with the camera systems, but the relationship is elastic.

Considered with the possible estimation instability due to a large number of surface parameters to be estimated, the pixel value differences between $I_0[\boldsymbol{u}_0]$ and $I_1[\boldsymbol{u}_1]$ caused by the differences of device characteristics and viewpoints are undesirable, even with several gray-levels, since the ground often have weakly textured surfaces. Therefore we take a photometric transformation into consideration. A widely-used transformation is represented by $I_0[\boldsymbol{u}_0] = \alpha_1 I_1[\boldsymbol{u}_1] + \alpha_2$, where α_1 and α_2 denote the gain and the bias, respectively. Let $\boldsymbol{\alpha} \equiv (\alpha_1, \alpha_2)^T$ represent the photometric parameter vector also to be estimated.

3. Direct Ground Surface Reconstruction

We minimize a cost composed of a data term and a smoothness constraint term by an iterative manner. In this section, however, to make full use of limited page space, we directly introduce an approximated cost function with an additional update constraint term for iteratively computing parameter updates, instead of showing the exact cost function to be minimized.

We define an additive update rule for the surface parameter vector as $\bar{z} \leftarrow \bar{z} + \Delta z$, where $\bar{z}$ and Δz respectively denote a current estimate and update of z. On the other hand, we adopt an inverse compositional update rule for the photometric parameter vector [2] as $\bar{\boldsymbol{\alpha}} \leftarrow (\frac{\bar{\alpha}_1}{1+\Delta\alpha_1}, \frac{\bar{\alpha}_2-\Delta\alpha_2}{1+\Delta\alpha_1})^T$, where $\bar{\boldsymbol{\alpha}} = (\bar{\alpha}_1, \bar{\alpha}_2)^T$ and $\Delta\boldsymbol{\alpha} = (\Delta\alpha_1, \Delta\alpha_2)^T$ respectively denote a current estimate and update of $\boldsymbol{\alpha}$.

The cost function, C, for computing updates, Δz and $\Delta\boldsymbol{\alpha}$, is written as

$$C(\Delta z, \Delta\boldsymbol{\alpha}) = C_D(\Delta z, \Delta\boldsymbol{\alpha}) + C_S(\Delta z) + C_U(\Delta z), \tag{1}$$

where C_D, C_S, and C_U denote a data term, smoothness constraint term, and update constraint term, respectively.

3.1 Data Term

We formulate the above data term by using the inverse compositional trick [1], [2] for both surface and photometric parameters for accelerating the estimation.

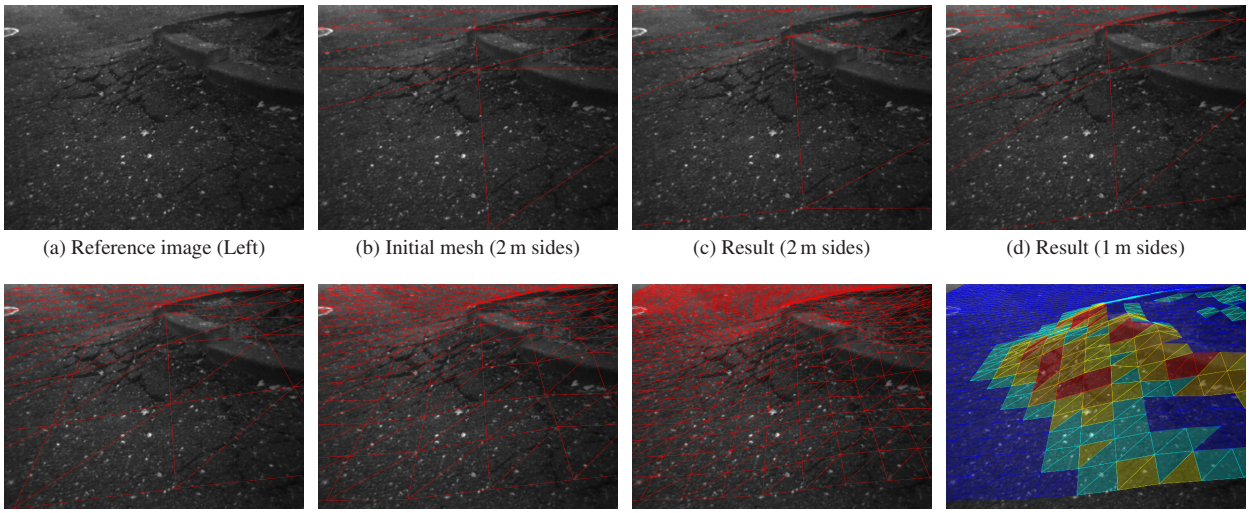
We denote by $\boldsymbol{w}_n(\boldsymbol{u}_0; \bar{z})$ a 2D transformation of a reference image point $\boldsymbol{u}_0$, dropped in the projection of the n-th triangular patch on the reference image, with a surface parameter vector $\bar{z}$. $\boldsymbol{w}_n(\boldsymbol{u}_0; \bar{z})$ can be written by a homography warp with homogeneous coordinate expressions $\tilde{\boldsymbol{u}}_0, \tilde{\boldsymbol{u}}_1$ as

$$\tilde{\boldsymbol{u}}_1 \propto \boldsymbol{H}_n\tilde{\boldsymbol{u}}_0 \tag{2}$$

$$\text{where} \quad \boldsymbol{H}_n = \boldsymbol{R}_s + \boldsymbol{t}_s\boldsymbol{m}_n^T(\boldsymbol{n}_n(\bar{z})), \tag{3}$$

where $\boldsymbol{n}_n(\cdot), \boldsymbol{m}_n(\cdot)$ and $\boldsymbol{H}_n$ respectively denote a 3×1 vector function representing plane parameters defined in the ground coordinate system, a similar one but defined in the reference camera coordinate system, and a 3×3 homography matrix.

We also denote by $\Delta\boldsymbol{w}_n(\boldsymbol{u}_0, \bar{z}; \Delta z)$ a *local* 2D transformation with an update vector Δz, which is only valid for a small parameter space around a given surface vector $\bar{z}$. We assume $\boldsymbol{w}(\boldsymbol{u}; \bar{z}) \circ \Delta\boldsymbol{w}(\boldsymbol{u}, \bar{z}; \Delta z)^{-1} = \boldsymbol{w}(\boldsymbol{u}; \hat{z})$, where $\hat{z}$ is the minimizer of the original data term. $\Delta\boldsymbol{w}_n(\boldsymbol{u}_0, \bar{z}; \Delta z)$ can also be expressed by a homography matrix $\Delta\boldsymbol{H}_n$, extended from the expression in Ref. [7] for fast plane parameter estimation.



(a) Reference image (Left) (b) Initial mesh (2 m sides) (c) Result (2 m sides) (d) Result (1 m sides)

(e) Result of (50 cm sides) (f) Result (25 cm sides) (g) Result (12.5 cm sides) (h) Patch clustering (12.5 cm)

Fig. 2 Result of the proposed method using hierarchical meshing. The image (h) shows the angle between the plane normal and the z-axis of each patch in the final hierarchical meshing results (g) (12.5 cm). Blue: smaller than 10. Cyan: 10~15, Yellow: 15~20, Orange: 20~25, Red: larger than 25 (in degrees).

$$\Delta \boldsymbol{H}_n = \boldsymbol{I} - \frac{1}{1 + \bar{\boldsymbol{m}}_n^T \boldsymbol{R}_s^T \boldsymbol{t}_s + \Delta \boldsymbol{m}_n^T \boldsymbol{R}_s^T \boldsymbol{t}_s} \boldsymbol{R}_s^T \boldsymbol{t}_s \Delta \boldsymbol{m}_n^T, \tag{4}$$

where $\bar{\boldsymbol{m}}_n = \boldsymbol{m}_n(\boldsymbol{n}_n(\bar{\mathsf{z}}))$. Since an additive update rule was adopted for $\boldsymbol{m}$ in Ref. [7], we can derive $\Delta \boldsymbol{m}_n(\Delta \boldsymbol{n}_n(\Delta z))$ by linear approximation from the expression of $\boldsymbol{m}_n$ (these exact expressions are omitted here).

We also denote a photometric transformation of a pixel value ξ with a photometric parameter vector $\bar{\boldsymbol{\alpha}}$ by $\mathcal{P}(\xi; \bar{\boldsymbol{\alpha}})$, and a local transformation with an update vector $\Delta\boldsymbol{\alpha}$ by $\Delta\mathcal{P}(\xi; \Delta\boldsymbol{\alpha})$. These transformations are presented in Ref. [2] as

$$\mathcal{P}(\xi; \bar{\boldsymbol{\alpha}}) = \bar{\boldsymbol{\alpha}}_1 \xi + \bar{\boldsymbol{\alpha}}_2, \tag{5}$$

$$\Delta\mathcal{P}(\xi; \Delta\boldsymbol{\alpha}) = (1 + \Delta\boldsymbol{\alpha}_1)\xi + \Delta\boldsymbol{\alpha}_2. \tag{6}$$

In this case we assume that $\Delta\mathcal{P}(\xi; \Delta\boldsymbol{\alpha})^{-1} \circ \mathcal{P}(\xi; \bar{\boldsymbol{\alpha}}) = \mathcal{P}(\xi; \hat{\boldsymbol{\alpha}})$, where $\hat{\boldsymbol{\alpha}}$ is the minimizer.

Then we write the data term as

$$C_D(\Delta z, \Delta\boldsymbol{\alpha}) = \sum_n \sum_{\boldsymbol{u} \in \tau_n} \kappa(\boldsymbol{u}) \Big(\Delta\mathcal{P}\left(T[\Delta \boldsymbol{w}(\boldsymbol{u}, \bar{\mathsf{z}}; \Delta z)]; \Delta\boldsymbol{\alpha}\right) - \mathcal{P}\left(I[\boldsymbol{w}(\boldsymbol{u}; \bar{\mathsf{z}})]; \bar{\boldsymbol{\alpha}}\right) \Big)^2, \tag{7}$$

where $\boldsymbol{u} \in \tau_n$ denotes the pixels $\boldsymbol{u}$ in the projection of the n-th triangular patch τ_n on the reference image, and $\kappa(\boldsymbol{u})$ is an iteratively-changed binary mask indicating whether the pixel $\boldsymbol{u}$ is used or not. We set κ by checking whether the surface point on the ray of $\boldsymbol{u}$ is visible from both images and the absolute image gradient on $\boldsymbol{u}$ is larger than a pre-defined threshold.

3.2 Smoothness Constraint Term

Since the mesh has a regular grid, we can adopt a simple smoothness term representing the sum of the squared Laplacian convolution outputs over the mesh.

$$C_S(\Delta z) = \lambda_S |\boldsymbol{F}(\bar{\mathsf{z}} + \Delta z)|^2, \tag{8}$$

where λ_S denotes a user-defined weight, and $\boldsymbol{F}$ denotes a $V \times V$ matrix whose v-th row $\boldsymbol{f}_v^T$ contains a 8-neighbor discrete Laplacian kernel for the v-th vertex. More specifically, the row vector $\boldsymbol{f}_v^T$ has an element 1 at the v-th vertex position, elements $-1/8$ at the 8-neighbor positions, and zeros at the other positions.

Let us note the inherent difference of the strengths of the smoothness constraints on the vertices near the cameras and far from the cameras. In our approach, the data term is more dominant on vertices nearer to the cameras, since more pixels in larger projected patches on the images contribute to the vertex height measurements, leading to a higher precision. On the other hand, for the surface part far from the camera, the smoothness term is more dominant due to the recession of the data term, leading to the improvement of the estimation robustness.

3.3 Update Constraint Term

The above two terms have their origin in the cost function to be minimized. However, the iterative estimation only with the two terms still engenders *flaps* of the distant part of the surface. This is because there is an ambiguity due to the recession of the data term, such that the heights of a group of vertices go up and down at the same time while keeping a flat shape (i.e., the smoothness cost is also small) in the distant part. For improving the stability, we add the update constraint term representing the norm of Δz.

$$C_U(\Delta z) = \lambda_U |\Delta z|^2, \tag{9}$$

where λ_U is a user-defined weight.

A possible drawback of the term would be a slow convergence. However, the update constraint term desirably works, when we combine the term with a hierarchical meshing approach, where we first roughly estimate the surface using a mesh with large squares and the level-of-detail of the mesh is increased in stages (see **Fig. 2**) while keeping the same weights of the smoothness and update constraint terms. In this case, a current level-of-detail succeeds to the previous rougher level-of-detail, in which the data term is more dominant than the current one because of larger triangular patches (the update term is also generally smaller itself

because of a smaller number of the vertices). Therefore, the efficacy of the update constraint term is not only to prevent the surface from flapping in each iteration process, but also to keep the current surface as similar as possible to the surface estimated by the previous level-of-detail. The efficacy is getting stronger as the meshing level is increased.

3.4 Computation of Update Vectors

Let us briefly describe a few important aspects of the computation of the update vectors $\Delta z, \Delta \alpha$ without details.

We estimate Δz of all vertices in the mesh drawn in a certain area of the x-y plane, even the vertices projected out of one of the two images (i.e., some heights will be estimated without any data term contributions).

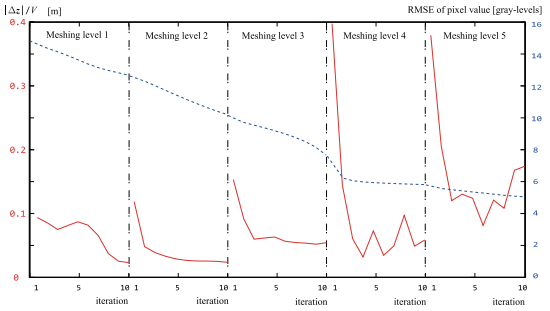
We apply Gauss-Newton optimization to the cost Eq. (1). The (approximated) Hessian of the proposed method is the summation of the three Hessians derived from the three terms. Although the data term is formulated by the inverse compositional trick, unfortunately, the Hessian of the data term should be recomputed in each iteration process, since the local 2D transformation $\Delta \boldsymbol{w}_n(\boldsymbol{u}_0, \bar{z}; \Delta z)$ depends on a current estimate $\bar{z}$ (the Hessians of the other two terms are constant). However, thanks to the inverse compositional trick, we do not need per-pixel Jacobian computations but need per-patch Jacobian computations, as indicated by Ref. [7]. In our experiments, the number of the triangular patches is at most 6,000 which is much smaller than the number of pixels (around 300,000). Therefore the Hessian computation is much faster than the case without the inverse compositional trick.

4. Experimental Results

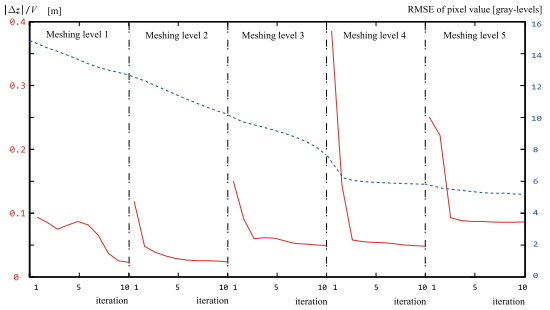
The algorithm was implemented in C++-language with a single thread and runs on a Windows7 PC (Xeon E3-1225 3.1 GHz, 16 GB). All images with the size of 640×480 pixels were captured by Point Gray Research Bumblebee2 with the baseline length of about 12 cm, mounted on a wheelchair at the height of about 1.0 m. We empirically set $\lambda_S = 1.5\times10^5$ and $\lambda_U = 5.0\times10^3$ for all experiments. We stop the iteration in each meshing level when the iteration number reaches 10 or when the maximum distance of the image projection points of the mesh vertices between $\bar{z}$ and $\bar{z} + \Delta z$ comes to smaller than 1.0 pixels.

Figure 2 (a) shows the input reference image which observes an asphalt road side where its ground level is partly raised by a long time growth of the tree in the right side of the scene. Figure 2 (b)~(g) shows results of our hierarchical meshing approach for the scene (a). We set a mesh in the range of $\{-2 \le x \le 2\} \times \{1 \le y \le 9\}$ (in meters) in front of the reference camera, and started the estimation algorithm with a mesh grid with sides 2 meter long. At each level the target surface was well approximated. The final mesh (g) (with 12.5 cm mesh sides) recovered the raised ground level in the center area while keeping other flat areas very well. Figure 2 (h) shows a patch clustering result on (g). Each color represents the angle between the plane normal of each patch and z-axis. The blue-colored patches indicate safely traversable areas for a mobile robot.

The convergence behaviors in the case with and without the update constraint term are shown in **Fig. 3** (a)(b), where the red solid lines and the blue broken lines respectively show the mean update values $|\Delta z|/V$ and the root mean squared differences of the data term over 10 iterations in the hierarchical estimation of Fig. 2. Since the update constraint term is small itself in the rough meshing levels 1~3 (2 m~50 cm), no remarkable differences between the two cases can be seen in these meshing levels. On the other hand, in the case without the update constraint term, although the data term is getting smaller in the progress of the meshing levels and iterations, the mean update value $|\Delta z|/V$ fluctuates due to the aforementioned recessions of both the data term and the smoothness term for the vertices distant from the camera. The update constraint term effectively reduces the fluctuation of Δz on the distant vertices while keeping the surface as close as possible to the one estimated in the previous meshing level.



(a) Case without the update constraint term

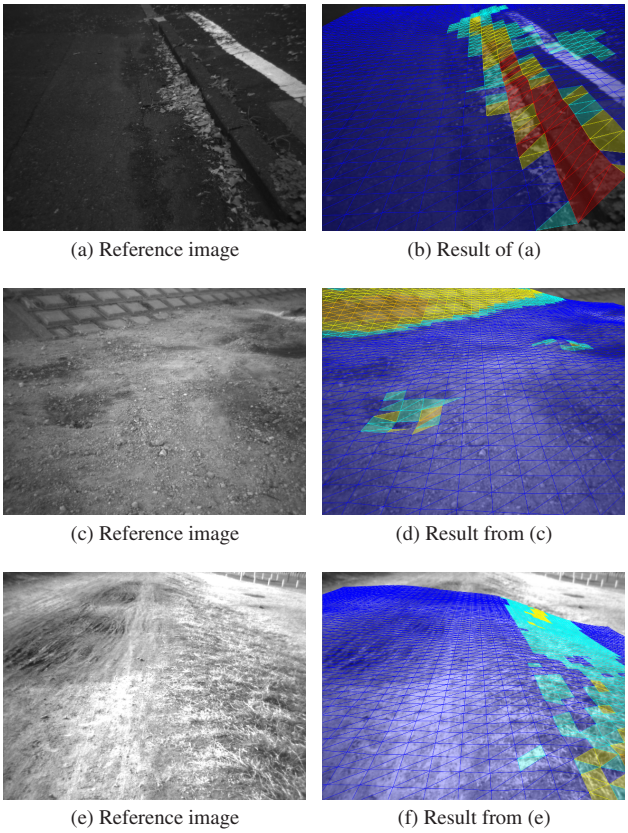


(b) Case with the update constraint term

Fig. 3 Comparision of the cases with/without the update constraint term. We plotted $|\Delta z|/V$ (red solid lines) and RMSE of pixel values (blue broken lines) at each iteration process in the five hierarchical meshing levels for the scene Fig. 2 while keeping the iteration number 10 for every meshing level.

In the case of Fig. 2, the total computational time was about 1.2 second over the five hierarchical meshing levels (the final mesh had 3,185 vertices and 6,144 patches). The iteration numbers were 10, 7, 3, 2, and 3 in the meshing levels 1 (2 m), 2 (1 m), 3 (50 cm), 4 (25 cm), and 5 (12.5 cm), respectively.

Figure 4 shows three surface reconstruction results for other scenes. Figure 4 (a) shows an asphalt road with a sidewalk bump in the right of the scene. The reconstruction result (b) shows a preferably recovered bump position. Figure 4 (c) shows an off-road scene with a very small hole and a hillock on the ground surface, which were well recovered by the proposed method as shown in (d). Figure 4 (e) and (f) also show an off-road and a recovered large slope in the right part of the scene. We believe that these colored regular-grid representations directly obtained from the stereo images are very helpful for traversable area detection and path-planning for robot systems.



(a) Reference image (b) Result of (a)

(c) Reference image (d) Result from (c)

(e) Reference image (f) Result from (e)

Fig. 4 Surface reconstruction results for other scenes.

5. Conclusions

We have proposed a method for directly reconstructing a ground surface with a regular square grid from stereo images. We have iteratively minimized a cost function composed of a data term formulated by the inverse compositional trick, a smoothness term, and an update constraint term, by Gauss-Newton optimization and a hierarchical meshing approach. The experimental results have shown that the ground surfaces could be preferably recovered even in the parts far from the camera.

The current computational time is promising for real-time applications since it is possible to highly parallelize the per-patch computation in our proposed method. Such an acceleration and ego-motion estimation will be studied during future research work.

Acknowledgments This work was partly supported by Grant-in-Aid for Scientific Research (21240015) from the Japan Society for the Promotion of Science.

References

[1] Baker, S. and Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework, *International Journal of Comoputer Vision*, Vol.56, No.3, pp.221–255 (2004).

[2] Bartoli, A.: Groupwise Geometric and Photometric Direct Image Registration, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.30, No.12, pp.2098–2108 (2008).

[3] Delling, D., Sanders, P., Schultes, D. and Wagner, D.: Engineering Route Planning Algorithms, *Algorithmics of Large and Complex Networks*, pp.117–139, Springer-Verlag, Berlin (2009).

[4] Geiger, A., Roser, M. and Urtasun, R.: Efficient Large-Scale Stereo Matching, *Asian Conference on Computer Vision*, Vol.Part I, pp.25–38 (2010).

[5] Kazhdan, M., Bolitho, M. and Hoppe, H.: Poisson surface reconstruction, *Eurographics Symposium on Geometry Processing*, pp.61–70 (2006).

[6] Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H. and Zhang, X.: On Building an Accurate Stereo Matching System on Graphics Hardware, *IEEE International Conference on Computer Vision Workshops* (*ICCV Workshops*), pp.467–474 (2011).

[7] Sugimoto, S. and Okutomi, M.: A Direct and Efficient Method for Piecewise-Planar Surface Reconstruction from Stereo Images, *CVPR*, pp.1–8 (2007).

[8] Szeliski, R. and Coughlan, J.: Spline-based Image Registration, *International Journal of Computer Vision*, Vol.22, No.3, pp.199–218 (1997).

[9] Vergauwen, M., Pollefeys, M. and Gool, L.V.: A stereo-vision system for support of planetary surface exploration, *Machine Vision and Applications*, Vol.14, No.1, pp.5–14 (2003).

[10] Wang, L., Yang, R., Gong, M. and Liao, M.: Real-time stereo using approximated joint bilateral filtering and dynamic programming, *Journal of Real-Time Image Proccessing*, pp.1–15 (2012).

[11] Williamson, T. and Thorpe, C.: A Specialized Multibaseline Stereo Technique for Obstacle Detection, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.238–244 (1998).

[12] Zhang, Z.: A stereovision system for a planetary rover: Calibration, correlation, registration, and fusion, *Machine Vision and Applications*, Vol.10, No.1, pp.27–34 (1997).

(Communicated by *Takayuki Okatani*)