A Direct and Efficient Method for Piecewise-Planar Surface Reconstruction from Stereo Images

Shigeki Sugimoto and Masatoshi Okutomi
Department of Mechanical and Control Engineering, Graduate School of Science and Engineering,
Tokyo Institute of Technology
2-12-1-S5-22 O-okayama, Meguro-ku, Tokyo, 152-8550 Japan
{shige,mxo}@ok.ctrl.titech.ac.jp

Abstract

In this paper, we propose a direct method for 3D surface reconstruction from stereo images. We reconstruct a 3D surface by estimating all depths of the vertices of a mesh composed of piecewise triangular patches on the reference (template) image.

The analyses described in this paper subsume that the deformation of the mesh between the stereo images is specified by homographies, each of which represents the deformation of a single patch. The homography deforms each patch which has 3 d.o.f. under epipolar constraints. We first formulate a fast "direct" method for estimating the three parameters of a 3D plane by incorporating inverse compositional expression into the sum of squared differences (SSD) function of two stereo images. This method is about eight times faster than the conventional method. Then we extend the direct method to the estimation of the vertex depths in the mesh for reconstructing piecewise-planar surfaces. The validity of the proposed method is demonstrated through results of experiments using synthetic and real images.

1. Introduction

In robot navigation, it is important to obtain 3D information of the ground surface. It is desirable that the 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. A helicopter or a spacecraft also requires surface normals for searching a level site to land safely. The surface normals are also required for a joint control system of a bipedal robot for walking stably on the ground, and for driver-assistance systems of a vehicle for seeking slopes and bumps in a road region.

A simple idea for surface reconstruction is to use dense depth data obtained using traditional stereo techniques [2, 6]. However, it is difficult to obtain surface model parameters from noisy stereo data. Numerous other techniques are useful to reconstruct surface models from stereo images [5, 4, 9, 11]. Unfortunately, most techniques are specific to object-oriented high-quality 3D reconstruction. Consequently, they are too time-consuming for use in robot navigation.

Our basic idea is simple; it is an extension of techniques on image registration for deformable object. When we generate a mesh composed of triangular patches on a reference image, the 3D information of a target surface is derived from the mesh deformation between two stereo images. We can assume that the deformation between the images is specified by homographies (eight-parameter projective deformations), each of which represents the deformation of a single patch. The epipolar constraints of stereo images enforce the homography of each patch on 3 degrees of freedom (d.o.f.) The degrees-of-freedom number is derived from the number of vertices of a triangle. We can estimate all vertex depths, each of which has 1 d.o.f., by minimizing the sum of squared differences (SSD) function between the reference image and the image warped from the other.

However, when we formulate an algorithm straightforwardly, the algorithm is too time-consuming. Therefore, we first formulate a fast direct method for estimating 3D plane parameters, which are homography parameters under epipolar constraints, by incorporating inverse compositional expression into the SSD function between two images. This method is about eight times faster than a conventional method, and about two times faster than full eight parameter estimation using the inverse compositional image alignment (ICIA) algorithm [1] of homography parameter estimation, while keeping higher precision. After that formulation is described, we extend the direct method to estimation of the vertex depths in the mesh for reconstructing curved surfaces.

1-4244-1180-7/07/$25.00 ©2007 IEEE

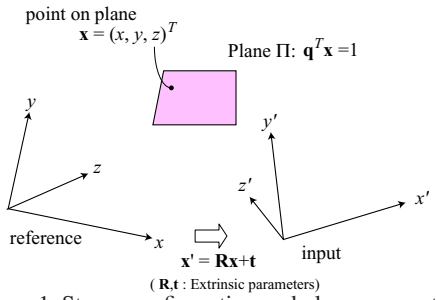


Figure 1. Stereo configuration and plane parameters

2. Plane Parameter Estimation

In this section, we describe a direct method for estimating 3D parameters of the target plane (i.e. its distance and plane normal) observed using stereo cameras. We formulate a fast method using inverse compositional expression. This direct plane parameter estimation can be regarted as the homography estimation under epipolar constraints.

2.1. Homography – Basic Notations

Let $\mathbf{x}$ and $\mathbf{x}'$ denote a scene point with respect to the two different camera views. We write: $\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t}$, where $\mathbf{R}$ and $\mathbf{t}$ respectively denote the rotation matrix and the translation vector between the two camera coordinate frames.

Let $I[\mathbf{u}]$ and $I'[\mathbf{u}']$ be the pixel values of the reference image I and the input image I', respectively, where $\mathbf{u} = (u, v)^T$ and $\mathbf{u}' = (u', v')^T$ respectively denote the corresponding points in I and I'. For avoiding complexity, let $\mathbf{u}$ and $\mathbf{u}'$ be in the *canonical* image configuration.

Let Π be a plane with a unit plane normal $\mathbf{n}$ and a distance d in the 3D coordinate frame of the reference camera. Thereby, the relationship between $\mathbf{u}$ and $\mathbf{u}'$ can be written by using a 3×3 homography matrix $\mathbf{P}$ as follows [3]:

$$\tilde{\mathbf{u}}' \sim \mathbf{P}\tilde{\mathbf{u}}, \quad (1)$$

$$\text{where } \mathbf{P} = \mathbf{R} + \mathbf{t}\mathbf{q}^T, \ \mathbf{q} \equiv \mathbf{n}/d. \quad (2)$$

Note that $\mathbf{q}$ determines a plane equation in the reference camera coordinate frame (see Fig. 1). In this section, we refer to $\mathbf{q}$ as the plane parameter vector to be estimated.

2.2. Conventional Direct Method

Let $\mathbf{w}(\mathbf{u}; \mathbf{p})$ denotes the homography warps derived from (1), where $\mathbf{p} = (p_1, p_2, \cdots, p_9)^T$ is a homography parameter vector which is a function of $\mathbf{q}$, as indicated by (2). A conventional direct method for estimating $\mathbf{q}$ minimizes the SSD function as follows [7]:

$$\sum_{\mathbf{u}\in\mathrm{ROI}} \left\{ I[\mathbf{u}] - I'\left[\mathbf{w}\left(\mathbf{u}; \mathbf{p}(\bar{\mathbf{q}} + \Delta\mathbf{q})\right)\right] \right\}^2, \quad (3)$$

where $\bar{\mathbf{q}}$ and $\Delta\mathbf{q}$ respectively denote a known current estimate and unknown increments of $\mathbf{q}$. Herein, ROI denotes a region of interest in I.

Applying Gauss-Newton optimization to (3) yields

$$\Delta\mathbf{q} = -\mathbf{H}^{-1}\mathbf{b}, \quad (4)$$

$$\text{where } \mathbf{H} \equiv \sum_{\mathbf{u}\in\mathrm{ROI}} \left\{ \left[\frac{\partial I'}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{q}}\right]^T \left[\frac{\partial I'}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{q}}\right] \right\}, \quad (5)$$

$$\mathbf{b} \equiv \sum_{\mathbf{u}\in\mathrm{ROI}} \left\{ e \left[\frac{\partial I'}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{q}}\right]^T \right\}, \quad (6)$$

$$e \equiv I[\mathbf{u}] - I'[\mathbf{w}(\mathbf{u}; \mathbf{p}(\bar{\mathbf{q}}))]. \quad (7)$$

The final estimate of $\mathbf{q}$ is obtained after iterations in each of which $\bar{\mathbf{q}}$ is updated by $\bar{\mathbf{q}} \leftarrow \bar{\mathbf{q}} + \Delta\mathbf{q}$ after computing $\Delta\mathbf{q}$ by (4). In the conventional case, $\partial I'/\partial \mathbf{w}$ and $\partial \mathbf{w}/\partial \mathbf{p}$ at each pixel should be re-computed in each iteration because these differentials are evaluated at the current estimate of $\bar{\mathbf{q}}$. These per-pixel and per-iteration computations for obtaining $\mathbf{H}$ impart large computational costs.

According to [1], in the case of homography parameter estimation, an SSD function formulated by inverse compositional expression offers remarkably fast estimation without loss of precision. In the next subsection, we incorporate an inverse compositional expression and additive expression for formulating an alternative to (3) for fast estimation of the plane parameters $\mathbf{q}$.

2.3. Efficient Estimation

A homography matrix can be written as

$$\mathbf{P} = \bar{\mathbf{P}}[\mathbf{I} + \Delta\mathbf{P}]^{-1}, \quad (8)$$

where $\bar{\mathbf{P}}$ and $\Delta\mathbf{P}$ respectively represent a current estimate of $\mathbf{P}$ and a matrix with small elements. Let $\bar{\mathbf{p}}$ and $\Delta\mathbf{p}$ be parameter vectors composed respectively by the elements of $\bar{\mathbf{P}}$ and $\Delta\mathbf{P}$. Equation (8) is a fundamental in the ICIA (Inverse Compositional Image Alignment) algorithm [1] for fast homography estimation.

Assume that $\bar{\mathbf{p}}$ and $\Delta\mathbf{p}$ are functions of $\bar{\mathbf{q}}$ and $\Delta\mathbf{q}$, respectively, and $\Delta\mathbf{p} \to \mathbf{0}$ when $\Delta\mathbf{q} \to \mathbf{0}$, where $\mathbf{q} = \bar{\mathbf{q}} + \Delta\mathbf{q}$ (concrete expressions are presented later). Then we re-write (3) as

$$\sum_{\mathbf{u}\in\mathrm{ROI}} \left\{ I[\Delta\mathbf{w}\left(\mathbf{u}; \Delta\mathbf{p}(\Delta\mathbf{q})\right)] - I'\left[\mathbf{w}\left(\mathbf{u}; \bar{\mathbf{p}}(\bar{\mathbf{q}})\right)\right] \right\}^2, \quad (9)$$

where $\mathbf{w}(\mathbf{u}; \bar{\mathbf{p}})$ and $\Delta\mathbf{w}(\mathbf{u}; \Delta\mathbf{p})$ denote the homography warps derived respectively from $\tilde{\mathbf{u}}' \sim \bar{\mathbf{P}}\tilde{\mathbf{u}}$ and $\tilde{\mathbf{u}}' \sim [\mathbf{I} + \Delta\mathbf{P}]\tilde{\mathbf{u}}$.

Applying Gauss-Newton optimization to (9) yields

$$\Delta\mathbf{q} = -\mathbf{H}^{-1}\mathbf{b}, \tag{10}$$

where

$$\mathbf{H} \equiv \sum_{\mathbf{u}\in \mathrm{ROI}} \left\{ \left[\frac{\partial I}{\partial \mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}} \frac{\partial \Delta\mathbf{p}}{\partial \Delta\mathbf{q}} \right]^T \left[\frac{\partial I}{\partial \mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}} \frac{\partial \Delta\mathbf{p}}{\partial \Delta\mathbf{q}} \right] \right\}, \tag{11}$$

$$\mathbf{b} \equiv \sum_{\mathbf{u}\in \mathrm{ROI}} \left\{ e \left[\frac{\partial I}{\partial \mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}} \frac{\partial \Delta\mathbf{p}}{\partial \Delta\mathbf{q}} \right]^T \right\}, \tag{12}$$

$$e \equiv I[\mathbf{u}] - I'[\mathbf{w}(\mathbf{u}; \mathbf{p}(\bar{\mathbf{q}}))]. \tag{13}$$

As in the conventional method described in 2.2, $\bar{\mathbf{q}}$ is updated by $\bar{\mathbf{q}} \leftarrow \bar{\mathbf{q}} + \Delta\mathbf{q}$ in each iteration. However, in this case, $\partial I/\partial \Delta\mathbf{w}$ and $\partial \Delta\mathbf{w}/\partial \Delta\mathbf{p}$ are constant in each iteration because these differentials are evaluated at $\Delta\mathbf{q} = \mathbf{0}$ (*i.e.* $\Delta\mathbf{p} = \mathbf{0}$) [1]. $\partial I/\partial \Delta\mathbf{w}$ denotes the gradients of the reference image. In addition, $\partial \Delta\mathbf{w}/\partial \Delta\mathbf{p}$ can be written as

$$\frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}} = \begin{bmatrix} u & v & 1 & 0 & 0 & 0 & -u^2 & -uv & -u \\ 0 & 0 & 0 & u & v & 1 & -uv & -v^2 & -v \end{bmatrix}. \tag{14}$$

The remaining problem is how to compute $\partial \Delta\mathbf{p}/\partial \Delta\mathbf{q}$. We derive it next.

2.3.1 Derivation of $\partial \Delta\mathbf{p}/\partial \Delta\mathbf{q}$

From (2), we can write

$$\mathbf{P} = \mathbf{R} + \mathbf{t}[\bar{\mathbf{q}} + \Delta\mathbf{q}]^T. \tag{15}$$

The relationship between $\Delta\mathbf{p}$ and $\Delta\mathbf{q}$ is defined by re-writing (15) into the form of (8).

Sherman-Morrison's formula [8] gives the inverse of (15):

$$\mathbf{P}^{-1} = \mathbf{R}' + \mathbf{t}'[\bar{\mathbf{q}}' + \Delta\mathbf{q}']^T, \tag{16}$$

$$\text{where } \mathbf{R}' \equiv \mathbf{R}^{-1},\ \mathbf{t}' \equiv -\mathbf{R}^{-1}\mathbf{t}, \tag{17}$$

$$\bar{\mathbf{q}}' + \Delta\mathbf{q}' \equiv \frac{\mathbf{R}[\bar{\mathbf{q}} + \Delta\mathbf{q}]}{1 + [\bar{\mathbf{q}} + \Delta\mathbf{q}]^T\mathbf{R}^{-1}\mathbf{t}} \tag{18}$$

Re-writing (16) yields

$$\mathbf{P}^{-1} = [\mathbf{I} + \mathbf{t}'\Delta\mathbf{q}'^T\bar{\mathbf{P}}]\bar{\mathbf{P}}^{-1}, \tag{19}$$

$$\text{where } \bar{\mathbf{P}} \equiv \mathbf{R} + \mathbf{t}\bar{\mathbf{q}}^T. \tag{20}$$

The inverse of (19) is written as

$$\mathbf{P} = \bar{\mathbf{P}}[\mathbf{I} + \mathbf{t}'\Delta\mathbf{q}'^T\bar{\mathbf{P}}]^{-1}. \tag{21}$$

Therefore, comparing (21) with (8) gives

$$\Delta\mathbf{P} = \mathbf{t}'\Delta\mathbf{q}'^T\bar{\mathbf{P}}. \tag{22}$$

From (18), $\Delta\mathbf{q}'$ can be written as

$$\Delta\mathbf{q}' = \frac{(1 + \bar{\mathbf{q}}^T\mathbf{R}^T\mathbf{t})\mathbf{R}\Delta\mathbf{q} - (\Delta\mathbf{q}^T\mathbf{R}^T\mathbf{t})\mathbf{R}\bar{\mathbf{q}}}{(1 + \bar{\mathbf{q}}^T\mathbf{R}^T\mathbf{t} + \Delta\mathbf{q}^T\mathbf{R}^T\mathbf{t})(1 + \bar{\mathbf{q}}^T\mathbf{R}^T\mathbf{t})}. \tag{23}$$

Then, substituting (23) and (17) for (22) obtains

$$\Delta\mathbf{P} = -\frac{1}{1 + \bar{\mathbf{q}}^T\mathbf{R}^T\mathbf{t} + \Delta\mathbf{q}^T\mathbf{R}^T\mathbf{t}}\mathbf{R}^T\mathbf{t}\Delta\mathbf{q}^T. \tag{24}$$

Equation (24) is directed to the expression of $\partial \Delta\mathbf{p}/\partial \Delta\mathbf{q}$ evaluated at $\Delta\mathbf{q} = \mathbf{0}$. We derive

$$\frac{\partial \Delta\mathbf{p}}{\partial \Delta\mathbf{q}} = \frac{1}{\kappa}\begin{bmatrix} \alpha_1 & 0 & 0 & \alpha_2 & 0 & 0 & \alpha_3 & 0 & 0 \\ 0 & \alpha_1 & 0 & 0 & \alpha_2 & 0 & 0 & \alpha_3 & 0 \\ 0 & 0 & \alpha_1 & 0 & 0 & \alpha_2 & 0 & 0 & \alpha_3 \end{bmatrix}^T, \tag{25}$$

where

$$\kappa \equiv -(1 + \bar{\mathbf{q}}^T\mathbf{R}^T\mathbf{t}), \tag{26}$$

$$\alpha_1 \equiv t_1r_1 + t_2r_4 + t_3r_7,\ \ \alpha_2 \equiv t_1r_2 + t_2r_5 + t_3r_8,$$

$$\alpha_3 \equiv t_1r_3 + t_2r_6 + t_3r_9, \tag{27}$$

Therein, $t_i (i = 1, 2, 3)$ and $r_j (j = 1, ..., 9)$ respectively denote the elements of $\mathbf{t}$ and $\mathbf{R}$.

2.3.2 Estimation Algorithm

Unfortunately, $\partial \Delta\mathbf{p}/\partial \Delta\mathbf{q}$ can not be constant in each iteration because κ depends on the current estimate $\bar{\mathbf{q}}$, which varies from iteration to iteration. However, κ is a simple scalar independent of pixels. Therefore, the final Gauss-Newton optimization algorithm can be written as

$$\Delta\mathbf{q} = -\kappa\mathbf{H}'^{-1}\mathbf{b}', \tag{28}$$

where

$$\mathbf{H}' \equiv \sum_{\mathbf{u}\in \mathrm{ROI}} \left\{ \left[\frac{\partial I}{\partial \mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}} \left[\kappa \frac{\partial \Delta\mathbf{p}}{\partial \Delta\mathbf{q}} \right] \right]^T \times \left[\frac{\partial I}{\partial \mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}} \left[\kappa \frac{\partial \Delta\mathbf{p}}{\partial \Delta\mathbf{q}} \right] \right] \right\}, \tag{29}$$

$$\mathbf{b}' \equiv \sum_{\mathbf{u}\in \mathrm{ROI}} \left\{ e \left[\frac{\partial I}{\partial \mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}} \left[\kappa \frac{\partial \Delta\mathbf{p}}{\partial \Delta\mathbf{q}} \right] \right]^T \right\}, \tag{30}$$

$$e \equiv I[\mathbf{u}] - I'[\mathbf{w}(\mathbf{u}; \mathbf{p}(\bar{\mathbf{q}}))]. \tag{31}$$

Compared with (10), we simply replace $\partial \Delta\mathbf{p}/\partial \Delta\mathbf{q}$ by $\kappa\partial \Delta\mathbf{p}/\partial \Delta\mathbf{q}$ which is a constant 9×3 matrix composed by the elements of $\mathbf{R}$ and $\mathbf{t}$, as described in (25). This replacement renders the product (1×3 column vector) of $\partial I/\partial \Delta\mathbf{w}$, $\partial \Delta\mathbf{w}/\partial \Delta\mathbf{p}$, and $\kappa\partial \Delta\mathbf{p}/\partial \Delta\mathbf{q}$ at each pixel constant in each iteration. Therefore, $\mathbf{H}'$ and its inverse can be pre-computed before the iteration process.

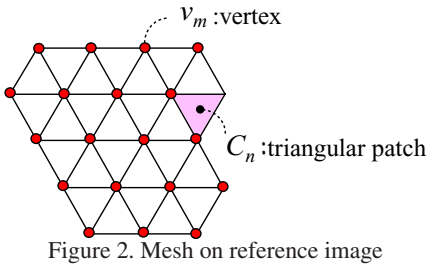


Figure 2. Mesh on reference image

Note that $\partial\Delta\mathbf{w}/\partial\Delta\mathbf{p}$ and $\kappa\partial\Delta\mathbf{p}/\partial\Delta\mathbf{q}$ are independent of the scene. The product (2×3 matrix) of the two matrices at every pixel can be computed when the cameras are calibrated. Additionally, the computational costs of κ are negligible compared to $\mathbf{b}'$, which requires per-pixel computations in each iteration. Therefore, we can consider that the per-parameter computational costs of this algorithm are almost equivalent to those of the ICIA algorithm; the presented plane parameter estimation algorithm is faster than the ICIA algorithm of homography estimation.

3. Piecewise-Planar Surface Reconstruction

We can say that the plane parameters are the 3D model parameters of the scene; to estimate plane parameters is to estimate the 3D parameters of the scene modeled by a single plane. In this section, we extend the single plane model to a multiple plane model.

3.1. Triangular Mesh

We first generate a 2D triangular mesh on the reference image. The triangular mesh includes M vertices $v_m (m = 1, \cdots, M)$ and N triangular patches $C_n (n = 1, \cdots, N)$, as shown in Fig. 2. The position of a vertex on the reference image is specified by $\mathbf{u}_m = (u_m, u_m)^T$, and its depth is represented as d_m.

Several approaches are useful for generating a 2D triangular mesh, such as simple meshing by regular triangles or Delaunay algorithms using corner-like feature points. Many types of meshing algorithms are applicable to our direct method.

In our experiments, the mesh is simply composed of regular triangles. A simple mesh is sufficient for hierarchical estimation of a target surface. We can roughly estimate the surface using a mesh with large patches. According to the extent of shape details required for a user application, the surface can be expressed more precisely using smaller patches.

3.2. Surface Reconstruction by Vertex Depth Estimation

We assume that the deformation of the mesh between the stereo images is specified by homographies. The homography that deforms each triangle has 3 d.o.f. under epipolar constrains. This number corresponds to the number of elements of plane parameters and to the number of vertices that define a triangle. This is readily apparent from the fact that, under epipolar constraints, the displacement of each vertex between two images has 1 d.o.f., which corresponds to its depth variation. We formulate these relationships for directly estimating all vertex depths; the SSD value between the reference image and the image wapred from the input image by the homographies determined by the depths is minimized.

3.2.1 Relationship of Plane and Depth Parameters

We begin with re-formation of the algorithm presented in Section 2 for estimating the three depths of a single triangular patch.

Let $\{i, j, k\}$ be the indices of the three vertices of the n-th patch. Because the image coordinates are expressed using canonical image configuration, the 3D positions of the vertices on the target surface in the reference camera coordinate frame are written as $\mathbf{x}_m = (d_m u_m, d_m v_m, d_m)^T$, where $(m = i, j, k)$. The relationship between the plane parameters $\mathbf{q}_n$ of the n-th patch and the depths d_m of the vertices is written as

$$\mathbf{q}_n = \mathbf{L}_n \boldsymbol{\gamma}_n, \tag{32}$$

$$\text{where} \quad \boldsymbol{\gamma}_n \equiv (1/d_i, 1/d_j, 1/d_k)^T, \tag{33}$$

$$\mathbf{L}_n \equiv \begin{bmatrix} u_i & v_i & 1 \\ u_j & v_j & 1 \\ u_k & v_k & 1 \end{bmatrix}^{-1}. \tag{34}$$

Consequently, we easily extend (9) for estimating the depth parameter vector $\boldsymbol{\gamma}_n$ of the n-th patch. Let $\boldsymbol{\gamma}_n = \bar{\boldsymbol{\gamma}}_n + \Delta\boldsymbol{\gamma}_n$, where $\bar{\boldsymbol{\gamma}}_n$ and $\Delta\boldsymbol{\gamma}_n$ respectively represent a current estimate of $\boldsymbol{\gamma}_n$ and a vector with small elements. We can thereby re-write (9) as

$$\sum_{\mathbf{u}\in C_n} \{I[\Delta\mathbf{w}(\mathbf{u}; \Delta\mathbf{p}_n(\Delta\mathbf{q}_n(\Delta\boldsymbol{\gamma}_n)))] - I'[\mathbf{w}(\mathbf{u}; \bar{\mathbf{p}}_n(\bar{\mathbf{q}}_n(\bar{\boldsymbol{\gamma}}_n)))]\}^2. \tag{35}$$

The relationship between $\Delta\mathbf{q}_n$ and $\Delta\boldsymbol{\gamma}_n$ is obtainable from the linear equation (32).

3.2.2 Estimation of All Depths

The extension to our surface reconstruction method by estimating all depths is obtained straightforwardly from (35).

Let $\mathbf{\Gamma} \equiv (1/d_1, 1/d_2, \cdots, 1/d_M)^T$, and $\mathbf{\Gamma} = \bar{\mathbf{\Gamma}} + \Delta\mathbf{\Gamma}$. Then we re-write (35) as

$$\sum_n \sum_{\mathbf{u}\in C_n} \{I[\Delta\mathbf{w}(\mathbf{u}; \Delta\mathbf{p}_n(\Delta\mathbf{q}_n(\Delta\mathbf{\Gamma})))] - I'\left[\mathbf{w}\left(\mathbf{u}; \bar{\mathbf{p}}_n(\bar{\mathbf{q}}_n(\bar{\mathbf{\Gamma}}))\right)\right]\}^2. \quad (36)$$

Applying Gauss-Newton optimization to (36) yields

$$\Delta\mathbf{q} = -\mathbf{H}^{-1}\mathbf{b}, \quad (37)$$

where

$$\mathbf{H} \equiv \sum_n \frac{1}{\kappa_n^2} \times \sum_{\mathbf{u}\in C_n} \left\{ \left[\frac{\partial I}{\partial \Delta\mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}_n} \left[\kappa_n \frac{\partial \Delta\mathbf{p}_n}{\partial \Delta\mathbf{q}_n}\right] \frac{\partial \Delta\mathbf{q}_n}{\partial \Delta\mathbf{\Gamma}} \right]^T \times \left[\frac{\partial I}{\partial \Delta\mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}_n} \left[\kappa_n \frac{\partial \Delta\mathbf{p}_n}{\partial \Delta\mathbf{q}_n}\right] \frac{\partial \Delta\mathbf{q}_n}{\partial \Delta\mathbf{\Gamma}} \right] \right\}, \quad (38)$$

$$\mathbf{b} \equiv \sum_n \frac{1}{\kappa_n} \times \sum_{\mathbf{u}\in C_n} \left\{ e_n \left[\frac{\partial I}{\partial \Delta\mathbf{w}} \frac{\partial \Delta\mathbf{w}}{\partial \Delta\mathbf{p}_n} \left[\kappa_n \frac{\partial \Delta\mathbf{p}_n}{\partial \Delta\mathbf{q}_n}\right] \frac{\partial \Delta\mathbf{q}_n}{\partial \Delta\mathbf{\Gamma}} \right]^T \right\}, \quad (39)$$

$$e_n \equiv I[\mathbf{u}] - I'[\mathbf{w}\left(\mathbf{u}; \mathbf{p}_n\left(\bar{\mathbf{q}}_n(\bar{\mathbf{\Gamma}})\right)\right)], \quad (40)$$

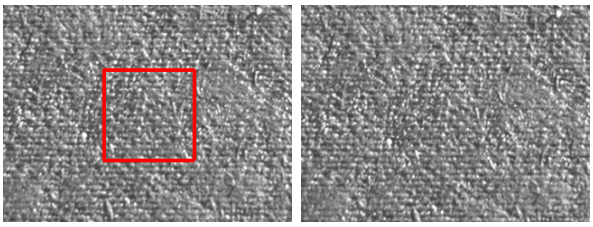
$$\kappa_n \equiv -(1 + [\bar{\mathbf{q}}_n(\bar{\mathbf{\Gamma}})]^T \mathbf{R}^T \mathbf{t}). \quad (41)$$

Because the relationship between $\Delta\mathbf{q}_n$ and $\Delta\mathbf{\Gamma}$ is linear, the product of the four matrices $\partial I/\partial\Delta\mathbf{w}$, $\partial\Delta\mathbf{w}/\partial\Delta\mathbf{p}_n$, $\kappa\partial\Delta\mathbf{p}_n/\partial\Delta\mathbf{q}_n$, and $\partial\Delta\mathbf{q}_n/\partial\Delta\mathbf{\Gamma}$ at each pixel is constant in each iteration.

In each iteration, a $1 \times M$ vector at each pixel, which is the product of the four matrices, has only three non-zero elements because $\partial\Delta\mathbf{q}_n/\partial\Delta\mathbf{\Gamma}$ is affected only by three depth values of the n-th triangular patch. Consequently, the additional computation of this piecewise-planar surface reconstruction to the plane parameter estimation of one plane is the inverse of an $M \times M$ matrix $\mathbf{H}$. The costs in each iteration can be roughly estimated as $O(3P) + O_{\text{inv}}$, where P is the number of pixels in the entire mesh and O_{inv} is the computational cost for the inverse. Note that O_{inv} can be application specific; if a roughly estimated surface is required, O_{inv} can be ignored. We can easily control the cost using hierarchical meshing.

4. Experimental Results

We first show the experimental results of the plane parameter estimation method presented in Section 2. Then we show some surface reconstruction results using the method presented in Section 3. All algorithms were implemented in C-language and run on a Linux PC (Pentium-IV 2.8 GHz).



(a) Reference image (b) Input image

Figure 3. Stereo images used in simulation for plane parameter estimation: (a) is an example of reference images. The reference images are created by warping the input image (b) using randomly generated plane vectors. The rectangle denotes a 100 × 100 ROI whose position is fixed on the reference image.

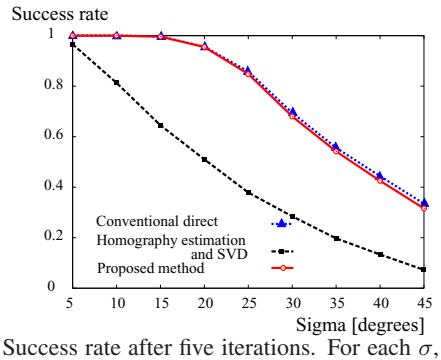


Figure 4. Success rate after five iterations. For each σ, 5000 sets of plane parameters were generated randomly .

4.1. Plane Parameter Estimation

The presented method for plane parameter estimation will be used for obtaining the 3D parameter of the ground plane from stereo cameras mounted on a bipedal robot or a vehicle for real-time control. We demonstrate the capability of the presented method using a comparative study of three methods that estimate plane parameters. These methods are the conventional direct method described in Section 2.2, a method which combines the ICIA algorithm of homography estimation [1] with singular value decomposition (SVD) for obtatining the plane parameters as presented in [10], and our method presented in Section 2.3.2. The convergence stability and the computational time of the three methods were evaluated.

We first set the plane parameters of a reference plane as $\mathbf{q}_0 = \mathbf{n}_0/d_0$, where $\mathbf{n}_0 = (0, 0, 1)^T$ and $d_0 = 15.24$ (in meters). We also set the origin of the object coordinates of the reference plane at the cross point of the optical axis of the reference camera and the plane. For setting a target plane, the reference plane was moved by a set of three values that were generated randomly by zero-mean Gaussian noise of a certain standard deviation σ.

Using each set of the three values, the reference plane was rotated with respect to its x and y axes (in degrees),

Method	pre-computation	per iteration	total(5 iterations)	total(100 iterations)
Conventional direct	—	15.19	75.95	1519
Homography estimation and SVD	11.09	2.613	24.16	272.4
Proposed Method	2.526	1.312	9.087	133.5

Table 1. Computational time (ms) in plane parameter estimation: Averaged over 100 trials for a 100 $\times$ 100 ROI. All algorithms were implemented using C-language and run on a Linux PC (Pentium-IV 2.8 GHz).

and moved along with the optical axis (at meters/20 increments). Then we computed the plane parameters $\mathbf{q}$ of the target plane and warped the input image, as shown in Fig. 3(b), by the target plane parameters for creating a reference image, as in Fig. 3(a). We added Gaussian noise of standard deviation 4 gray levels to both images. Then we run the algorithms of the three methods starting from $\mathbf{q}_0$.

For each σ, we evaluated the frequency of convergence over 5000 trials. We judged that an algorithm succeeded after five iterations, if the angle between $\mathbf{q}$ and the estimates was less than 0.5 degrees. We evaluated the angle because applying SVD to a homography matrix recovers only the plane normal direction. The results are shown in Fig. 4.

Figure 4 shows that the two direct methods for plane parameter estimation give far higher success rates than the method for homography. Under the epipolar constraints of the stereo images, the input image deformation is restricted to the epipolar lines. This restriction is remarkably profitable for acquiring high stability.

Table 1 shows the computational time of the three methods. Comparing our method with the others shows that the five-iteration time of our method is about one-eighth as long as the conventional method, and about one-twice as long as the homography estimation method.

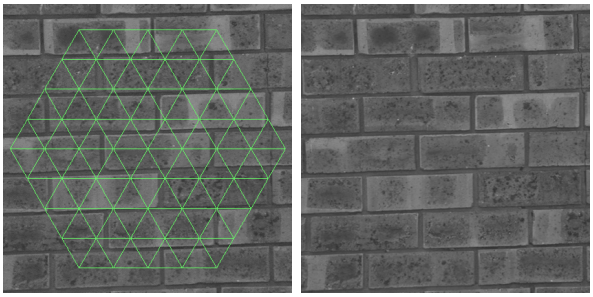
4.2. Surface Reconstruction

In this subsection, we present the experimental results of the proposed method for surface reconstruction. The algorithm is implemented in C-language. We used the CLapack library for computing inverse matrices.

4.2.1 Synthetic Images

We created a textured 3D object and stereo images, as shown in Fig. 5. The object shape was a large sphere with the center at $\mathbf{x} = (0, 0, 15)^T$ and radius 7.7 m. The mesh generated on the reference image comprised 61 vertices and 96 triangles with sides that were 50 pixels long. We run the algorithm of the proposed method starting from all depths with 10 m.

Figure 6 shows the errors of depths for each iteration. Figure 7 also shows the sequence of the estimated surface during the iterations. All depths converged to good values after 20 iterations.



(a) Reference image with mesh (b) Input image

Figure 5. Synthetic stereo images 1: (a)(b) 420 $\times$ 420 size stereo images with baseline 0.3 meters. The mesh, which comprises regular triangles with sides that are 50 pixels long ($M = 61, N = 96$) is shown in (a).

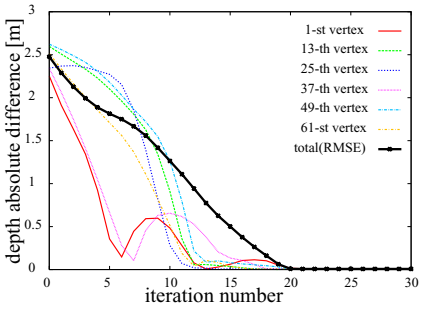
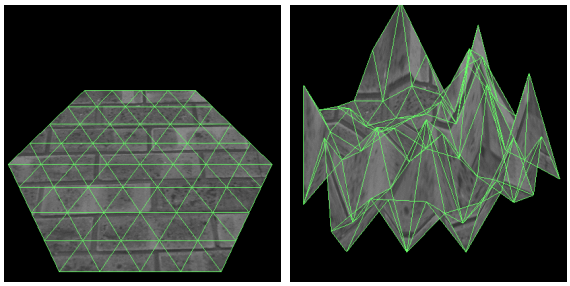


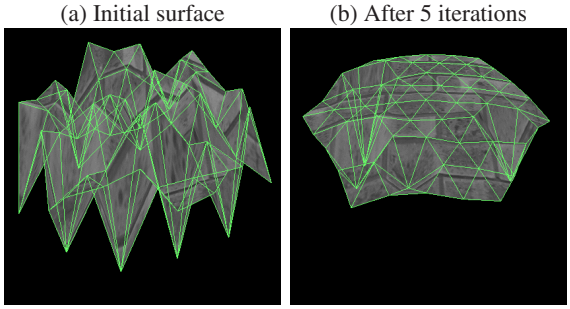
Figure 6. Depth error in each iteration: The error of each vertex depth (only six depths) is shown. The total error (RMSE) of all 61 depths is shown by the thick graph.

Figure 8 shows the result of a hierarchical meshing approach. We created another synthetic stereo images with the size of 420 $\times$ 420 pixels. For such a target with similar texture patterns on different surface positions, a fine mesh diverges when the algorithm starts with inappropriate initial depths. We first generated a mesh composed by regular triangles with sides 200 pixels long (level 0); then we hierarchically divided each triangle into four regular triangles with half sides. The final mesh had 217 vertices and 384 triangles with sides that were 25 pixels long. We obtained a satisfactory surface after 20 iterations for each meshing level.

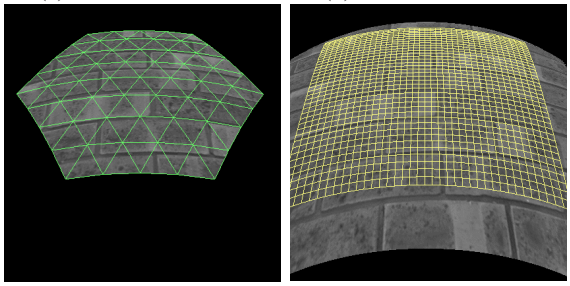
In this experiment, the computations of all meshing levels for 20 iterations take 0.6 s, except 0.9 s in level 3. These



(a) Initial surface (b) After 5 iterations

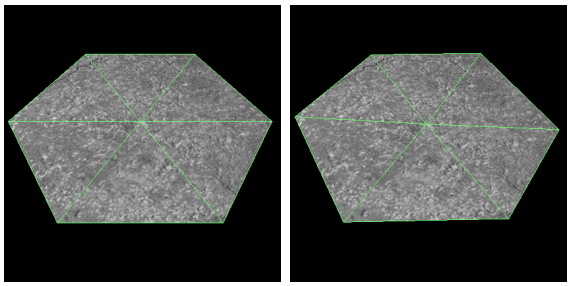


(c) After 10 iterations (d) After 15 iterations

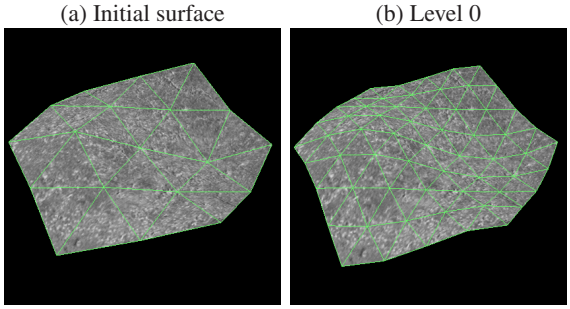


(e) After 20 iterations (f) Actual 3D shape

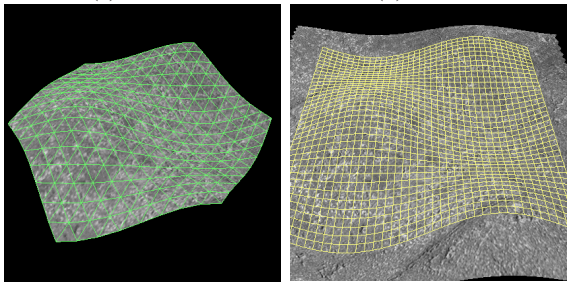
Figure 7. Surface estimated in each iteration: Our algorithm, which uses about 10^5 pixels in the whole mesh, requires 0.61 s for 20 iterations.



(a) Initial surface (b) Level 0



(c) Level 1 (d) Level 2



(e) Level 3 (f) Actual 3D shape

Figure 8. Surface estimated at each meshing level after 20 iterations. the computations of all meshing levels for 20 iterations take 0.6 s, except 0.9 s, which was required for level 3.

timing results are promising for various navigation applications.

4.2.2 Real Images

We used stereo images (1280 $\times$ 720 pixels large) as shown in Fig. 9. The target has a moon-like surface, which is simulated on a sandy plane. For reconstructing the surface, we also use a hierarchical approach. We first generated a mesh comprising regular triangles with sides that are 464 pixels long (level 0). The final mesh has 817 vertices and 1536 triangles with sides 29 pixels long (level 4). In this experiment, we use a condition for each iteration process to stop: $|\Delta\Gamma| < 10^{-4}$. The plane parameter estimation method described in Section 2 was first employed to set initial vertex depths over the whole region.

Figure 10 shows results of the hierarchical approach. At each level, the target surface was preferably approximated. The final mesh recovered valleys in the center and craters in the left-top area well.

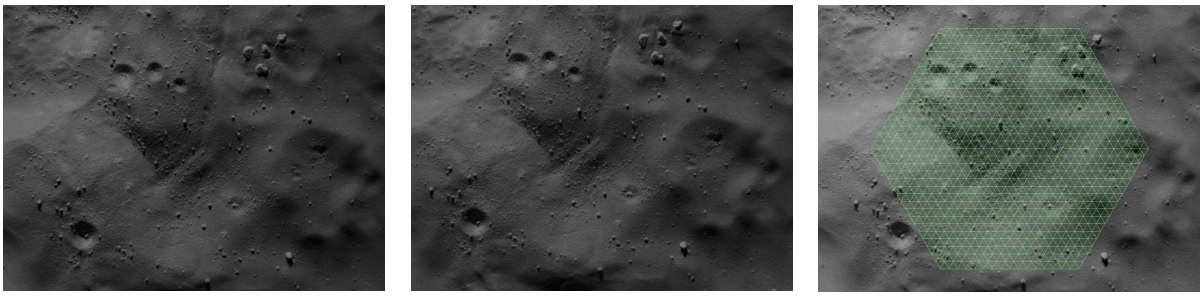
5. Conclusions

We have proposed a direct and efficient method for surface reconstruction. A fast direct method for estimating the three parameters of a plane was proposed and subsequently extended to the surface reconstruction method. The experimental results are promising from the standpoint of the surface representation ability and computational efficiency.

Many options are available in our method for better surface reconstruction, such as multi-baseline stereo strategy and uses of a-priori information. The evaluation of these options will be studied during future research work.

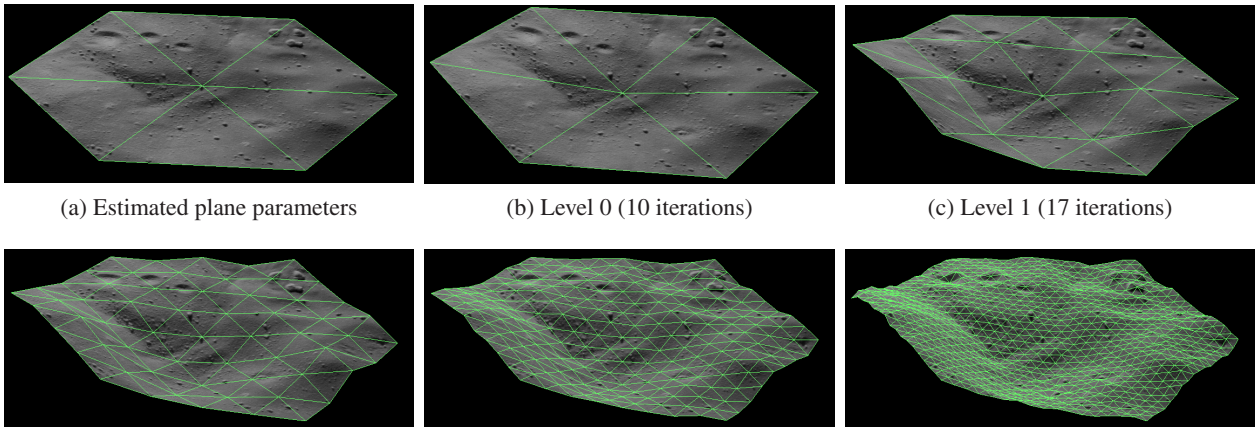
Acknowledgement

We would like to thank Dr. Yasuhiro Katayama with Japan Aerospace Exploration Agency (JAXA) for supplying the real images that we used in the experiments.



(a) Reference image (b) Input image (c) A mesh

Figure 9. 1280 $\times$ 720 size real images: A moon-like surface was simulated on a sandy place.



(a) Estimated plane parameters (b) Level 0 (10 iterations) (c) Level 1 (17 iterations)

(d) Level 2 (4 iterations) (e) Level 3 (3 iterations) (f) Level 4 (4 iterations)

Figure 10. Surface estimated in each iteration (real images)

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004.

[2] D. Cobzas and H. Zhang. Planar patch extraction with noisy depth data. In *International Conference on 3-D Digital Imaging and Modeling*, pages 240–245, 2001.

[3] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *Report de Recherche de l'INRIA*, 1988.

[4] O. D. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *ECCV*, volume I, pages 379–393, 1998.

[5] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *IJCV*, 16(1):35–56, September 1995.

[6] S. B. Kang, A. Johnson, and R. Szeliski. Extraction of concise and realistic 3-D models from real data. Technical report, Digital Equipment Corporation, Cambridge Research Lab, CLR 95/7, 1995.

[7] Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *CVPR*, volume 1, pages 390–397, June 2003.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge Univ Press, 1992.

[9] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, volume 1, pages 519– 528, 2006.

[10] A. Seki and M. Okutomi. Robust obstacle detection in general road environment based on road extraction and pose estimation. In *IV2006*, pages 437–444, 2006.

[11] L. Zhang and S. Seitz. Image-based multiresolution shape recovery by surface deformation. In *In SPIE: Videometrics and Optical Methods for 3D Shape Measurement*, pages 51–61, 2001.