A Footstep-Plan-Based Floor Sensing Method Using Stereo Images for Biped Robot Control

Minami Asatani, Shigeki Sugimoto, and Masatoshi Okutomi

Abstract—In this paper, we propose a floor sensing method using stereo cameras mounted on a biped robot. In the proposed method, we first determine multiple regions of interest (ROI) in a reference image from footstep positions up to several steps, scheduled by a current footstep plan. Then the 3D plane parameters of the floor with respect to each ROI are estimated by a direct method using stereo images. We adopt the fast plane parameter estimation method [5], along with the compensation for the errors of the initial parameters by using the internal state of the robot, for the enhancement of the robustness and efficiency in the optimization process. Additionally, we estimate the shape of the floor including slopes from the set of the estimated plane parameters, and feedback the results for updating the footstep plan. The validity of the proposed method is demonstrated through on-line experiments using stereo cameras mounted on the body of a biped robot traversing a real environment.

I. INTRODUCTION

The recent developments of biped robot control researches have led to the remarkably enhanced traversability of biped robots. On the other hand, to realize the autonomous walking of biped robots with the help of the traversability, it is essential to sense the unknown walking environments in real-time. For this purpose, several approaches have been presented recently ([1] ~ [4]).

In [1], the robot communicates with a ceiling mounted camera which detects colored obstacles and extract stepable areas used for planning footsteps. In [3], the body-mounted stereo cameras are used for obtaining depth maps and extracting the areas with the same level height for detecting stepable areas. The stereo depth maps obtained by body-mounted cameras are also used for walking on stairs in [2], where the ‘height map’ of the walking environment is estimated by a depth map segmentation technique.

These sensing methods are the approach subject to the walking on level planes (with the normal direction parallel to the gravity). The restriction of the walking only on level planes permits relatively easy data segmentation for noise-riddled depth maps. On the other hand, to realize stable walking on slopes, it is necessary to accurately estimate the inclinations of the slopes and edge positions where the inclination changes. These parameters are indispensable for deciding footstep positions. However, it is not easy to extract them from the noisy depth map in real time.

In this paper, we propose a real-time floor sensing method for biped robots. In the context of this paper, a ‘floor’ means a walking environment including slopes. We first determine regions of interest (ROI) from the scheduled footstep positions up to several steps. Then the 3D plane parameters (i.e. the distance to the origin and the plane normal) of the floor in each ROI are estimated by a direct method using stereo images. The direct method enables us to estimate target parameters in high precision even though objects are not textured enough. Thus we don’t use any markers on the floor. The fast plane parameter estimation method [5] is adopted for the real-time estimation with multiple ROIs. In addition, we compensate the initial errors due to robot movements, by using the internal state of the robot, for the enhancement of the robustness and efficiency of the estimation. Moreover, for updating the scheduled footstep positions, we estimate the shape of the floor, which may include slopes, from the set of the estimated parameters by using a two-plane model. The importance and the validity of the proposed method are demonstrated through on-line experiments using stereo cameras mounted on the body of a biped robot traversing a real environment.

II. PLANE PARAMETER ESTIMATION USING FOOTSTEP POSITIONS

In this section, we describe the plane parameter estimation using scheduled footstep positions which determine the regions of interests (ROI) in a reference image. We first roughly review the fast plane parameter estimation method using stereo images [5]. Then the compensation for the errors of the initial parameters and the role of scheduled footstep positions are explained.

A. Direct Plane Parameter Estimation

A 3D plane \( \Pi \) is expressed by \( \mathbf{n}^T \mathbf{x} = d \), where \( \mathbf{n} \) and \( d \) respectively denote its unit normal and the distance from the origin, and \( \mathbf{x} = (x, y, z)^T \) indicates 3D positions on the plane in a reference camera coordinate frame, as shown by Fig. 1. In this section, we refer \( \mathbf{m} = \mathbf{n}/d \) as the plane parameter vector to be estimated.
Let \( I[u] \) and \( I'[u'] \) be the reference image and the other image, respectively, where \( u = (u, v)^T \) and \( u' = (u', v')^T \) respectively denote the corresponding image positions in \( I \) and \( I' \). For avoiding complexity, let \( u \) and \( u' \) be in the canonical image coordinates. Thereby, the relationship between \( u \) and \( u' \) can be written as follows [6]:

\[
\tilde{u}' = P\tilde{u},
\]

where \( P = R + tm^T \). \hspace{1cm} (1)

Therein, \( \tilde{u} \) and \( \tilde{u}' \) denote homogeneous coordinates of \( u \) and \( u' \), respectively, and the matrix \( P \) denotes a 3x3 homography matrix. \( R \) and \( t \) respectively indicate the rotation matrix and the translation vector between the two camera coordinate frames. We assume that \( R \) and \( t \), along with the camera intrinsic parameters, are calibrated beforehand.

Let \( w(u; p(m)) \) denotes the homography warps derived from (1), where \( p = (p_1, p_2, \ldots, p_9)^T \) is a homography parameter vector which is a function of \( m = (m_x, m_y, m_z)^T \), as indicated by (2). In a conventional direct method for estimating the plane parameter vector \( m \), the following SSD (Sum of Squared Differences) function is minimized by a Gauss-Newton optimization algorithm:

\[
\sum_{u \in \text{ROI}} (I[u] - I'[w(u; p(m))])^2,
\]

where ROI signifies the region of interest in the reference image.

This direct method, where every pixel in the ROI directly contributes to the estimation, is more preferable than a feature-based approach, since typical floors have weakly textured and have few feature points. On the other hand, the iterative computations in the Gauss-Newton optimization algorithm require large computational costs. However, in [5], an alternative to (3) is formulated for remarkably fast computing the plane vector without loss of precision. We adopt this method and use the SSD function as follows:

\[
\sum_{u \in \text{ROI}} (I[\Delta w(u; \Delta p(\Delta m))] - I'[w(u; \overline{p}(\overline{m}))])^2,
\]

where \( \overline{m} \) is current estimates and iteratively estimated \( \Delta m \) is a small element vector defined by \( m = \overline{m} + \Delta m \), and \( \Delta w(u; \Delta p(\Delta m)) \) is defined by:

\[
w(u; p(m)) = w(u; \overline{p}(\overline{m})) \circ \Delta w(u; \Delta p(\Delta m))^{-1},
\]

where \( \circ \) denotes the composition of the two warps. In this case, we can iteratively estimate \( \Delta m \) as follows:

\[
\Delta m = -\kappa H^{-1}b,
\]

where

\[
H = \sum_{u \in \text{ROI}} \begin{bmatrix} \frac{\partial I}{\partial \Delta w} & \frac{\partial I}{\partial \Delta p} \end{bmatrix} \begin{bmatrix} \frac{\partial \Delta p}{\partial \Delta m} \end{bmatrix}^T.
\]

\[
b = \sum_{u \in \text{ROI}} \begin{bmatrix} I(u) - I'[w(u; \overline{p}(\overline{m}))] \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial \Delta w} & \frac{\partial I}{\partial \Delta p} \end{bmatrix} \begin{bmatrix} \frac{\partial \Delta p}{\partial \Delta m} \end{bmatrix}^T.
\]

\[\kappa = -(1 + \overline{m}^T R \overline{t}).\]

Therein, \( \partial I/\partial \Delta w \) denotes a 1x2 column vector which represents image gradients at \( u \), and \( \partial \Delta w/\partial \Delta p \) denotes a 2x9 Jacobean written by:

\[
\begin{bmatrix} \partial \Delta w/\partial \Delta p \end{bmatrix} = \begin{bmatrix} u & v & 1 & 0 & 0 & 0 & -u^2 & -uv & -u \end{bmatrix}.
\]

Moreover \( \kappa \times (\partial \Delta p/\partial \Delta m) \) denotes a constant matrix which depends on \( R \) and \( t \):

\[
\kappa \times (\partial \Delta p/\partial \Delta m) = \begin{bmatrix} \alpha_1 & 0 & 0 & a_5 & 0 & 0 & a_9 & 0 & 0 \\ 0 & a_1 & 0 & 0 & a_4 & 0 & 0 & a_8 & 0 \\ a_2 & 0 & 0 & a_5 & 0 & 0 & a_9 & 0 & a_7 \\ 0 & a_3 & 0 & a_4 & 0 & a_7 & a_8 & 0 & 0 \\ a_6 & 0 & a_5 & a_8 & 0 & a_7 & a_9 & 0 & 0 \\ 0 & a_6 & a_7 & a_8 & 0 & a_9 & 0 & 0 & 0 \\ a_7 & a_8 & a_9 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_8 & a_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},
\]

where \( a_1 = r_1k_1 + r_2k_1 + r_3k_2 \), \( a_2 = r_1k_3 + r_2k_3 + r_3k_4 \), \( a_3 = r_1k_5 + r_2k_5 + r_3k_6 \). Therein \( k_{ij} \) (\( i = 1, \ldots, 9 \)) and \( k_{ij} \) (\( j = 1, 2, 3 \)) respectively denote the elements of \( R \) and \( t \).

In this case, \( \partial I/\partial \Delta w \), \( \partial \Delta w/\partial \Delta p \), and \( \kappa \times (\partial \Delta p/\partial \Delta m) \) are constant during the iterations. Therefore \( H \) and its inverse, required in each iteration process, are also constant. This results in remarkably fast plane parameter estimation (see [5] for more details).

### B. Initial Error Compensation by Internal State of Robot

In the method mentioned above, the initial values of \( \overline{m} \) are given, then the precision is enhanced from iteration to iteration. To successively estimate the parameters that lead the SSD function to the global minimum, and to reduce the
iteration number of the optimization algorithm, it is important to set the initial values as near to the real values as possible.

When a direct method is applied for obtaining the plane parameters of the slopes in a road scene by using car-mounted stereo cameras, as presented in [7] and [8], it is possible to obtain relatively good estimates if the initial values are set to the last values at the preceding time, since there are relatively small changes of the plane parameters while a car cruises. On the other hand, a biped robot dramatically changes its pose relative to the floor while it walks. For example, the upper body goes lower when both feet are in touch with the floor than when only one foot is in touch. When a leg is carried forward, the upper body is twisted so as to cancel the reactive force. Consequently the preceding estimates are no longer good initial values at the current time. For obtaining better initial values from the preceding estimates, we compensate the errors due to the robot movements by using the internal state of the robot.

The data types of the internal state of the robot used for the error compensation are shown by Fig. 2. We use these data for obtaining the relationship between the global coordinate frame and the camera coordinate frame at the current time, where the robot starting position is the origin of the global frame. First the current footstep position in the global coordinate frame is computed by integrating every footstep from the starting position (i). Then the camera position and the camera posture relative to the footstep position are computed by using the robot posture command (ii). Finally we obtain the camera posture relative to the global coordinate frame by using the body inclination measured by the inclinometer mounted on the robot body (iii).

These processes enable us to determine the relationship between the global frame and the camera frame at every moment. Consequently, the relationship between the preceding camera frame and the current camera frame can be obtained. We adopt this relationship for compensating the errors between the preceding estimates and the current plane parameters.

C. Determining ROI using Scheduled Footstep Positions

In the plane parameter estimation method mentioned above, it is required to set a ROI in the reference image. If a large region is set for the estimation, the region tends to include not only multiple planes but also the areas superfluous for robot walking. On the other hand, it is essential for robust walking to obtain the plane vectors of the places where the robot steps on. Therefore we use scheduled footstep positions which are provisionally determined by a footstep plan.

In the proposed method, we project the scheduled 3D footstep positions up to several steps into the reference image, and set each ROI which includes the surroundings of the projected point. Then the fast estimation method is applied for every ROI.

Fig. 3 shows an example of the image ROIs determined by the scheduled footstep positions. In this figure, respective four steps of the robot are indicated by red, green, blue, and yellow quadrilaterals. These colors are repeated for more steps.

III. SHAPE ESTIMATION FOR FLOOR INCLUDING SLOPES

For walking on a floor including slopes, it is necessary to estimate not only the plane vector for each ROI, but also the position of the boundary edge between two different planes and the length of each plane. These parameters should be estimated as precisely as possible, since they are required by footstep planning which realizes robust walking. Each footstep position of the robot has to be properly adjusted in consideration of dynamic balance and also avoiding stepping on the boundary edge because of the flat soles of our robot. In this section, we describe how to estimate the shape of the floor including slopes.

A. Floor Shape Modeling

As described in II.A, a plane parameter vector provides the 3D plane equation. Consequently, the result of the plane parameter estimation for multiple ROIs determines the respective planar pieces in the 3D space. We can represent the
floor shape by extending the respective pieces. However, this modeling remains the question which region includes a boundary edge. Even though we adopt the direct method better for a weak textured ROI, it is difficult to judge whether the ROI includes a boundary edge and to detect the edge position, if the observed planes are not textured enough or the two plane vectors are close to each other.

In this paper, we assume that the robot walks on a floor including a few slopes each of which has a large flat area. Then we model the floor in the field of view of the reference camera by two planes, as shown in Fig.4. Then we compute the two plane vectors of both planes by the method described in the next subsection, from the estimated multiple plane vectors for the ROIs.

**B. Floor Shape Estimation**

Let the number of the ROIs in the reference image (the number of scheduled footstep positions observable in the reference image) be $N$, and the estimated plane vector for each region be $\mathbf{m}_i (i = 1, 2, \cdots, N)$, where the index increases from near to far. We compute a step number $k$ which minimize the following cost function:

$$
\sum_{i=1}^{N} |\mathbf{m}_i - \overline{\mathbf{m}}_{\text{front}}|^2 + \sum_{i=k+1}^{N} |\mathbf{m}_i - \overline{\mathbf{m}}_{\text{back}}|^2 ,
$$

where $\overline{\mathbf{m}}_{\text{front}} = \frac{1}{k} \sum_{i=1}^{k} \mathbf{m}_i$ and $\overline{\mathbf{m}}_{\text{back}} = \frac{1}{N-k} \sum_{i=k+1}^{N} \mathbf{m}_i$.

After that, for each of two plane vector groups, we compute the mean of the plane vectors by averaging both $\mathbf{n}_i$ and $d'_i$.

Then the angle between the two averaged vectors is computed. If the angle is larger than a pre-defined threshold, we judge that the floor is composed of two planes, otherwise one plane. In the case of two planes, we easily get the position of the boundary edge between the two planes from the two plane vectors. If the edge crosses with a particular ROI, the vector averaging is performed again without the ROI. This process contributes to enhancing the accuracy of the mean vector and the boundary position.

In our experiments, the stereo cameras are mounted on a biped robot at about 90 cm in height, and the looking-down angle of the cameras about 45 degrees. In this case, the reference camera cannot see far enough, and seldom or never see more than two planes in the field of view. When the scheduled footstep positions are projected in the reference image, the ROI number, which is equivalent to the number of scheduled footsteps in the field of view, is seven at the maximum. Thereby it is reasonable to divide the floor into two planes. Since this floor modeling approach is very simple, we can easily divided extend our approach into the case of more than two planes.

**IV. EXPERIMENTAL RESULTS**

In our on-line experiments, we mounted stereo cameras on the upper body of the robot (Fig. 5). The size of the stereo images was 640x480 pixels, and an external computer with AMD Opteron 2.2GHz (4 cores) was used for estimating the plane vectors and the floor shape. The floor was set as shown in Fig. 6. The surface of the floor was covered with carpets. We set the number of the scheduled footsteps was seven (the first footstep is never observed by the camera because its position is too near). The angular threshold for the judgment of the plane number was set to 1.5 degrees (see Section II.B). For pre-processing the stereo images, we compute the difference of two averaging filters (15x15, 7x7 pixels) for removing image noise and the bias between the stereo images. Additionally, for reducing the errors of the plane vector for each ROI, we use a temporal smoothing filter, as follows:

$$
\mathbf{m}^{(t)} \leftarrow \alpha \mathbf{m}^{(t)} + (1-\alpha) \mathbf{m}^{(t-1)},
$$

where $\mathbf{m}^{(t)}$ denotes the plane vector at time $t$, and $\alpha = 0.14$.

Fig. 7 shows how the scheduled footstep positions were being updated. First, the robot was walking on the ground level plane, as shown by (a). Immediately after that, a slope plane was viewed, and the slope shape was estimated by the proposed sensing method. This result is shown by (b). The dashed-horizontal line in the image shows the detected boundary edge, which precisely corresponds to the edge between the ground plane and the slope plane. Also, the scheduled footstep positions updated by the proposed method are shown. The positions were preferably updated from the proposed sensing method, and the robot avoided stepping on
the boundary edge. After that, the robot was walking on the slope plane, as shown by (c). Readers may notice that the distances between steps on the slope plane are naturally shorter than those on the ground floor. Then other plane was viewed and detected by the proposed method, as shown by (d). Again, the scheduled footstep positions were updated preferably for avoiding stepping on the boundary edge.

We also show the results of a comparative study. We compare the walking stabilities and behaviors between two cases. In both cases, the actual 3D shape of the floor (Fig. 6) is given in advance to the robot. And we compare (1) the robot walks without the proposed method, and (2) the robot walks while the 3D shape is updated at every moment by using the proposed method.

The following information of the floor shape was given to the robot: (a) Starting position of the slope (distance from the starting position of walking), (b) Inclination of the slope, (c) Length of the slope.

Even though the actual floor is given to the robot, the robot walking tends to be unstable due to the errors caused by slippage of the feet or the integration error of the real control values. The errors are caused in particular when the robot walks on a slope plane.

Fig. 8 shows the time-serially normalized angle in
anteroposterior (x-axis) inclination of the upper body of the robot, while the robot walked over the floor. The longitudinal axis indicates the normalized angle of the inclination. The value was computed by: $\theta' = (\theta - \theta_0) / \theta_{\text{max}}$, where $\theta$ denotes the actual angle, and $\theta_0$, $\theta_{\text{max}}$ indicate, respectively, the mean angle and the maximum angle during the normal walking on a level floor. The horizontal axis indicates millisecond time from the start of walking. The case with the proposed method is indicated by red, and the other is indicated by black. In the case using the proposed method, the robot walked stably and the angle stably fluctuates without deviation from a certain range. On the other hand, although the robot without the proposed method walked stably from the level ground plane and the slope plane, the robot suddenly inclined to the backward at 11200ms, when the robot’s left foot stepped on the highest plane and mistook the footstep position because of the error accumulated during the walking on the slope plane.

Fig. 9 shows the floor reactive force of the left foot when the robot reached the highest plane. In the case without the proposed method (black), the treading timing of the robot was faster by 17ms than the case using the proposed method (red). The undesirable impact caused by the foot treading engendered the sudden body incline. These experimental results show the importance of the floor sensing and the validity of the proposed method.

The processing during walking was performed at 20.7fps (3-stage pipeline) with latency about 110ms.

V. Conclusions

In this paper, we have proposed a floor sensing method using stereo images for the biped robot walking. The proposed method has successfully realized a real-time floor sensing which is beneficial to updating the footstep plan for traversing on a floor including slopes. The importance and the validity of the proposed method have been demonstrated by on-line experiments on a real slope.

The computation for the proposed method is performed by an external computer outside the robot. We are now planning to realize a stand-alone robot by a special hardware.

REFERENCES