International Journal of Automotive Technology, Vol. 16, No. 1, pp. 107−115 (2015)
DOI 10.1007/s12239−015−0012−7

pISSN 1229−9138/ eISSN 1976−3832

STEREO VISION ENABLING FAST ESTIMATION OF FREE SPACE ON TRAFFIC ROADS FOR AUTONOMOUS NAVIGATION

K. Y. LEE, G. Y. SONG, J. M. PARK and J. W. LEE*

Department of Industrial Engineering, Chonnam National University, Gwangju 500-757, Korea

(Received 26 December 2013; Revised 11 March 2014; Accepted 27 March 2014)

ABSTRACT−A novel algorithm capable of estimating free space for vehicle navigation is presented. When a disparity map – dense or sparse – from stereo matching and a longitudinal profile of the road surface on the disparity domain are provided, the free space is estimated precisely. According to the longitudinal profile of the road surface, the disparity map is classified into an obstacle disparity map and a road surface disparity map. After combining these two disparity maps through a score map, a border line separating the road surface and the non-road surface is estimated using dynamic programming on a u-disparity representation. The main contribution of the proposed approach is the robust detection of the free space and the distance between stereo cameras and obstacles, whereby the detection is sufficiently rapid for vehicle navigation. The validity of the proposed algorithm is demonstrated by experiments through many outdoor road images from various traffic scenarios.

KEY WORDS : Disparity histogram, Dynamic programming, Free space, Score map, Stereo vision

1. INTRODUCTION

In autonomous navigation, the route of the ego-vehicle (i.e., 'our car') is determined within the collision-free space (Badino *et al*., 2008; Kubota *et al*., 2007; Wedel *et al*., 2008; Oniga and Nedevschi, 2010). This paper presents an algorithm capable of estimating such space in front of an ego-vehicle using a disparity map from two-frame stereo correspondence algorithms. Such space is called 'free space'.

In the field of vehicle navigation, stereo vision has mainly been applied to the detection of obstacles, in which most approaches rely on a sparse or dense disparity map. Two well-known approaches have been presented, including the u- and v-disparity image based approach (Labayrade *et al*., 2002) and the digital elevation map (DEM) based approach (Oniga and Nedevschi, 2010). The former uses the stereo disparity directly in detecting the road surface and obstacles. The v-disparity representation basically provides the geometric content of the road scene. Accordingly, even under the presence of partial occlusion or disparity error, the robust extraction of the road surface and obstacles is expected. The DEM based approach proposed by Oniga and Nedevschi (2010) performs time consuming complicated and heuristic procedures in transforming the stereo disparity into DEM and detecting the road surface and obstacles. The method provides robust results but fails if obstacles are directly in front of the ego-vehicle. Furthermore, their method limits the detection depth to a relatively short distance from the ego-vehicle.

It is critical to reduce the computation time while maintaining robustness in calculating disparities. For these reasons, some approaches - such as top-view based methods (Lee *et al*., 2009; Bertozzi and Broggi, 1998) - do not rely on disparity map building. However, these methods assume that roads are flat. While the method introduced in (Lee *et al*., 2009) is fast and effective, it usually does not express the real shape of the road surface due to the flat-road assumption, which leads to false responses in detecting obstacles on non-flat roads such as downhill or uphill roads. The method in (Bertozzi and Broggi, 1998) exploits occlusion areas between left and right images. Since the different view-points of a stereo camera produce occlusions with respect to obstacles, the idea of this method is effective. However, in a real situation, many different obstacles that are large or small, and tall or short appear on roads. The baseline between stereo cameras can be either short or long. In addition, lighting conditions vary considerably. Therefore, the detection of occlusion areas often gives rise to errors, which eventually leads to errors in detecting obstacles. These two methods have inherent limitations in detecting obstacles on roads.

Disparity computation is time consuming because of the extensive correspondence searches and requires strong matching accuracy (Song and Lee, 2013). Nevertheless, we will use the disparity map in detecting the free space because, as mentioned in the previous paragraph, the alternative approaches have their own shortcomings.

**Corresponding author*. e-mail: joonlee@chonnam.ac.kr

In this paper, free space detection involves extracting the road surface without any standing obstacles. This detection is performed using the following procedures. First, using *vLine* and considering the height of the ego-vehicle, the disparity map is separated into an obstacle disparity map and a road surface disparity map. The *vLine* representing the profile of the road surface along the camera-viewing direction is estimated using the method introduced in (Lee *et al*., 2014). Second, along each column of the disparity map, a disparity histogram that is a frequency of a disparity in a range [0, $d_{\max}$] is built. The maximum value in each column of the histogram is likely formed by an obstacle because the single obstacle usually has similar disparities. According to this study, a search for the disparity with the maximum value can be considered reasonable. This search, however, may select a disparity by either a noise effect or a tall obstacle that is not the nearest obstacle to a stereo camera, and may lead to failure of the free space detection. To prevent such failure, this paper introduces a new concept of score map representing the possibility of obstacle presence. Third, the score maps are built from the disparity histograms of the obstacle disparity map and the road surface disparity map, and both score maps are combined. Finally, dynamic programming is applied to the combined score map to search for a disparity corresponding to the border separating the obstacles that are closest to a stereo camera and the road surface.

The remainder of the paper is structured as follows. Section 2 describes the estimation of the free space and Section 3 presents the experimental results. Finally, the conclusions are given in Section 4.

2. FREE SPACE DETECTION

2.1. Consideration of the Traveling Space of a Vehicle

Objects such as overpasses, road signs, traffic signals, and the top of the entrance to a tunnel can be recognized as obstacles. Such objects may or may not be obstacles depending on the height of the ego-vehicle. To avoid detecting such objects as obstacles, we first calculated the number of pixels corresponding to 1 meter-height for each disparity as shown in Figure 1. We then stored this calculation result in $N_{1m}(d)$.

Using $N_{1m}(d)$ and *vLine*, we can obtain the row number, $V_h(d)$, corresponding to the height of a traveling space of the ego-vehicle in the input image as follows:

$$N_{ts}(d) = h_{ts} \cdot N_{1m}(d) \tag{1}$$

$$V_h(d) = V(d) - N_{ts}(d) \tag{2}$$

where h_{ts} is the assumed height in metric units of a traveling space of the ego-vehicle, $N_{ts}(d)$ is the number of pixels corresponding to h_{ts}, and $V(d)$ is the row number corresponding to each disparity d on the *vLine* as shown in Figure 2, in which u is a column coordinate, v is a row coordinate, and d is a disparity. As h_{ts} can be chosen depending on the height of the ego-vehicle, h_{ts} is chosen as 2 meters considering passenger cars in this paper.

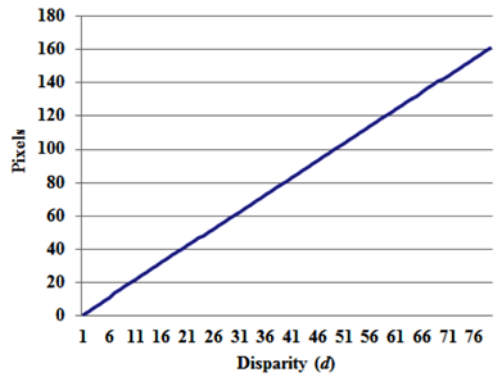


Figure 1. Number of pixels corresponding to 1-meter height for each disparity.

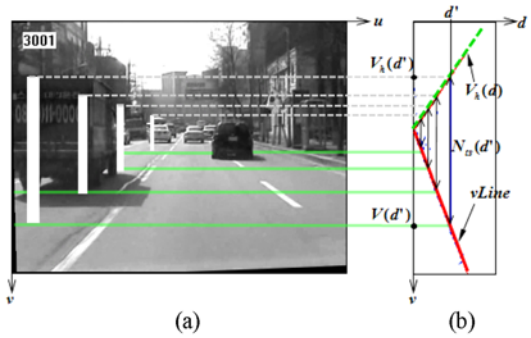


Figure 2. Height of a travelling space without obstruction from obstacles: (a) Input image and a 2-meter high column; and (b) v-disparity representation (Labayrade *et al*., 2002) and *vLine* (Lee *et al*., 2014).

Using $V_h(d)$ of Equation (2) in order to include the disparities within the traveling space, the disparity map $D(u, v)$ from stereo matching (Lee and Lee, 2011; Song and Lee, 2013) is filtered as follows:

$$D_{ts}(u,v) = \begin{cases} D(u,v) & \text{if } v \geq V_h(D(u,v)) \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

2.2. Separation of a Disparity Map

As the first step to extract the boundary between the road surface and obstacles, we separate $D_{ts}(u, v)$ into two maps - an obstacle disparity map, $D_o(u, v)$ and a road surface disparity map, $D_r(u, v)$ - as follows:

$$D_o(u,v) = \begin{cases} d & \text{if } v < V(d) - \varepsilon(d) \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

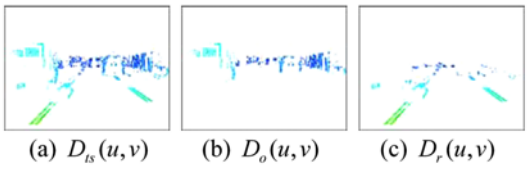


Figure 3. Obstacle and road surface disparity maps.

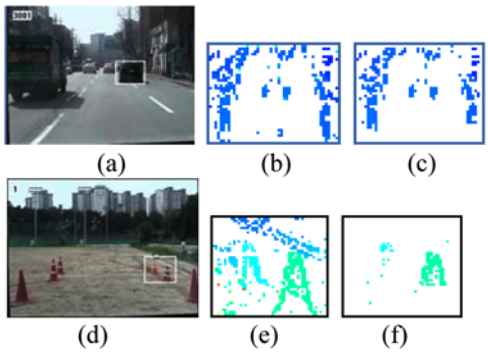


Figure 4. Effect of allowing the tolerance h_e: (a) A vehicle obstacle; (b) $D(u, v)$ of the vehicle; (c) $D_o(u, v)$ of the vehicle; (d) A cowl-head-cone obstacle; (e) $D(u, v)$ of the cowl-head-cone, and (f) $D_o(u, v)$ of the cowl-head-cone.

$$D_r(u,v)=\begin{cases} d & \text{if } V(d)-\varepsilon(d)\le v\le V(d)+\varepsilon(d) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $d = D_{ts}(u, v)$ and $\varepsilon(d) = h_e \cdot N_{1m}(d)$, in which h_e is the tolerance to compensate the error of *vLine*. We chose h_e as 0.3 through a number of experiments. If this tolerance is not allowed, disparities from the road surface can be included in the obstacle disparity map. Figure 3 shows $D_{ts}(u, v)$, $D_o(u, v)$, and $D_r(u, v)$.

On the other hand, because of the allowance for tolerance, the bottom part of the obstacles is severed from the body of obstacles. Figures 4 (c) and (f) show this phenomenon, in which two examples are given: a vehicle obstacle and a cowl-head-cone obstacle enclosed by a white rectangle in the raw image (Figures 4 (a) and (d)). Comparing $D(u, v)$ with $D_o(u, v)$ of both obstacles, it can be clearly seen that the bottom part of the obstacles is cut off. The short obstacles may not be detected without correction of this problem.

To solve the problem of the bottom part of the obstacles being cut off, we build a u-disparity image, which is a disparity histogram, as follows:

$$H_u(u,d)=\sum_{k=0}^{Row-1}\delta(D_{ts}(u,k),d) \tag{6}$$

where δ is the Kronecker delta function, *Row* is the number of pixels in a column of the disparity map $D_{ts}(u, v)$, d is the disparity from 0 to d_{max}, which is the maximum disparity set by the image size and camera parameters (Tsai, 1987), u indicates a column on the horizontal axis, and $D_{ts}(u, k)$ is the disparity at a pixel (u, k) on $D_{ts}(u, v)$. $H_u(u, d)$ is the frequency of the disparity on each column of a disparity map.

If $H_u(u, d)$ has a large value, it is likely to be from an obstacle because every corner of an obstacle has a similar disparity. Based on this argument, the separation error caused by Equations (4) and (5) can be corrected. For this correction, a threshold value is needed to determine whether or not a disparity is derived from an obstacle. Expecting that the threshold value is greater than the maximum frequency of a disparity from the road surface in $H_u(u, d)$, we use *vLine* in calculating this threshold value. For a disparity, d, the maximum frequency is set by $\tau = .5|V(d+1)-V(d-1)|$. We then select the threshold value of d as two times τ after considering the error of *vLine*.

The correction is performed using the following procedure. 1) The disparity from an obstacle is selected as follows:

$$o(d)=\begin{cases} 1 & if \;\; H_u(u,d)>2\cdot\tau \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

2) The obstacle disparity map and the road surface disparity map are reconstructed using $o(d)$ and the same conditions used in Equations (4) and (5) as follows:

$$D_o'(u,v)=\begin{cases} d & \text{if } o(d)=1 \text{ or } v<V(d)-\varepsilon(d) \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

$$D_r'(u,v)=\begin{cases} d & \text{if } con \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $con = o(d) \neq 1$ and $V(d)-\varepsilon(d) \le v \le V(d) + \varepsilon(d)$ and $d = D_{ts}(u, v)$. The result in Figure 4 is corrected considerably using Equations (8) and (9) as shown in Figure 5.

For the second step to extract the boundary between the road surface and obstacles, we construct u-disparity images of $D_o'(u,v)$ and $D_r'(u,v)$ as follows:

$$H_u^o(u,d)=\sum_{k=0}^{Row-1}\delta(D_o'(u,k),d), \tag{10}$$

$$H_u^r(u,d)=\sum_{k=0}^{Row-1}\delta(D_r'(u,k),d) \tag{11}$$

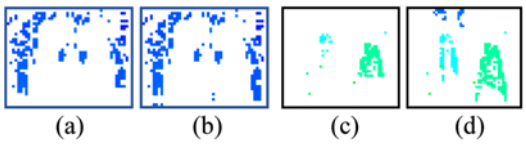
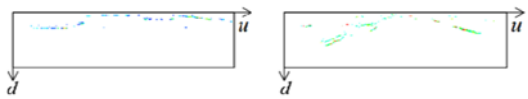


Figure 5. Correction of separation error: (a) Before correction for a vehicle obstacle; (b) After correction for a vehicle obstacle; (c) Before correction for a cowl-head-cone obstacle; (d) After correction for a cowl-head-cone obstacle.



(a) $H^o_u(u,d)$ (b) $H^r_u(u,d)$

Figure 6. u-disparity images $D'_o(u,v)$ of and $D'_r(u,v)$.

where $D'_o(u,k)$ and $D'_r(u,k)$ are disparities at a pixel (u,k) in $D'_o(u,v)$ and $D'_r(u,v)$, respectively. $H^o_u(u,d)$ and $H^r_u(u,d)$ for $D'_o(u,v)$ and $D'_r(u,v)$ are shown in Figure 6.

2.3. Score Maps

The third step to extract the boundary separating the road surface and obstacles involves constructing score maps. If $H^o_u(u,d)$ is large, the possibility of the presence of obstacles is also large. Besides this argument, some issues need to be considered: 1) some columns on the u-disparity image have no information if the disparities are not detected from stereo matching; 2) since a value in $H^o_u(u,d)$ is the frequency of a disparity, a short obstacle near a viewer may be missed if a tall obstacle is behind the short obstacle; and 3) how to combine $H^o_u(u,d)$ and $H^r_u(u,d)$.

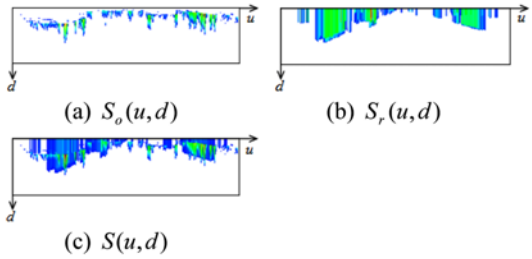
The first issue is solved by selecting the row of d=0 as the boundary. For the second and third issues, score maps of $H^o_u(u,d)$ and $H^r_u(u,d)$ are built, in which 'score' refers to the possibility of the presence of obstacles. The score is assigned depending on the value of $H^o_u(u,d)$ and $H^r_u(u,d)$, and on the strategies we determine. The strategies to build the score map of $H^o_u(u,d)$ are set as follows: 1) if a value of $H^o_u(u,d)$ is large, the score corresponding to this value is also large; 2) if $H^o_u(u,d)$ and $H^o_u(u,d+1)$ have values, the score at the pixel $(u, d+1)$ is larger than the score at the pixel $(u,d-1)$; 3) even though $H^o_u(u,d)$ has no value and $H^o_u(u,d-1)$ has a value, the score map at the corresponding two pixels, (u,d) and $(u,d-1)$, has a score, but the score at the pixel (u,d) is smaller than the score at the pixel $(u,d-1)$. The goal of the second and third strategies is to avoid missing the detection of obstacles near to a viewer. Reflecting these three strategies, the score map, $S_o(u,d)$ of $H^o_u(u,d)$, is built as follows:

$$S_o(u,d)=\begin{cases} S_o(u,d-1)+H^o_u(u,d) & \text{if } con1 \text{ is true} \\ H^o_u(u,d) & \text{if } con2 \text{ is true} \\ S_o(u,d-1)-1 & \text{if } con3 \text{ is true} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$S_o(u,0)=H^o_u(u,0)$$

where $con1=H^o_u(u,d)\neq 0$ and $S_o(u,d-1)\neq 0$, $con2=H^o_u(u,d)\neq 0$ and $S_o(u,d-1)=0$, $con3=H^o_u(u,d)=0$ and $S_o(u,d-1)\neq 0$, and d ranges from 1 to d_{max}.

The fact that $H^r_u(u,d)$ has a value means that the road surface relating to disparity d to d_{max} is visible without occlusion by obstacles. Accordingly, the score at the pixel (u,d) of $H^r_u(u,d)$ is set by the sum of $H^r_u(u,d)$ of the pixels of (u,d) to (u,d_{max}) as follows:



(a) $S_o(u,d)$ (b) $S_r(u,d)$

(c) $S(u,d)$

Figure 8. Score maps.

$$S_r(u,d)=S_r(u,d+1)+H^r_u(u,d),\quad S_r(u,d_{max})=H^r_u(u,d_{max}) \quad (13)$$

where d ranges from 0 to d_{max}–1. The third issue is solved by combining the score maps, $S_o(u,d)$ and $S_r(u,d)$, as follows:

$$S(u,d)=w(d)\cdot S_o(u,d)+S_r(u,d) \quad (14)$$

where, as a weight to emphasize the larger disparities rather than the smaller disparities, $w(d)$ is defined as $1+d/d_{max}$. The goal of $w(d)$ is to assist in the detection of obstacles that are close to the viewer. Figure 8 shows three score maps.

2.4. *uLine* Detection by Dynamic Programming

The fourth step involved in extracting the boundary between the road surface and the obstacles on the input image is to find the *uLine*, which is a border line between the road surface and obstacles in the combined score map, $S(u,d)$. This line is searched by dynamic programming (Hillier, 2006). Many paths exist starting from pixels on u=0 and ending at pixels on $u = Col$-1 in $S(u,d)$. Among them, a path with the maximum gain is selected as the optimal path. In the optimal path finding, all pixels in $S(u,d)$ become nodes. The optimal path is the combination of partial paths. Let $g(N_{m,n}, N_{i,j})$be the gain of the partial path from node $N_{m,n}$ to node $N_{i,j}$, and let $G(N_{i,j})$ be the gain of the optimal path from a node on $u = 0$ to node $N_{i,j}$, in which the node index represents the pixel position in $S(u,d)$. $G(N_{i,j})$ is defined recursively as follows:

$$G(N_{i,j})=\max\{G(N_{i-1,k})+g(N_{i-1,k},N_{i,j})\},\quad G(N_{0,j})=0, G(N_{i,0})=1 \quad (15)$$

where the range of i and j are $1 \le i \le Col-1$ and $0 \le j \le d_{max}$, respectively, and the range of k is $\max(0, j-1) \le k \le d_{max}$. The reason why k starts from j-1 is to ensure the optimal path passes through the boundary of the disparity occlusion area. No partial path can have a slope any steeper than 45° toward the upward-left. The initial condition of $G(N_{i,0})$ plays a role when there is no score on a column in $S(u,d)$. The gain, $g(N_{i-1,k}, N_{i,j})$ of a path in Equation (15) is defined by the sum of the current node $N_{i,j}$ and a value of smoothness constraint as follows:

$$g(N_{i-1,k},N_{i,j})=\omega_g\cdot S(i,j)+\Delta(j,k) \quad (16)$$

where $\Delta(j,k)$ is the value of smoothness constraint and ω_g is the weight to adjust the influence of $S(i, j)$ and $\Delta(j, k)$. $\Delta(j, k)$ is defined as follows:

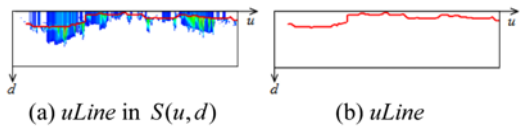
$$\Delta(j,k) = \max\{\beta^2 - (j-k)^2, 0\} \quad (17)$$

where β is the assumed maximum range of disparity change within a single obstacle. β is set as 3 throughout many experiments. Eventually, $\Delta(j,k)$ plays the role of preventing a large change in disparity between the current and previous nodes. ω_g in Equation (16) is also determined through many experiments. We chose ω_g as 0.5.

The optimal path is determined using a backward search after the gain computation of Equation (15) at all nodes in $S(u,d)$ is completed. On the last column, $u = Col\text{-}1$, a node with the maximum gain is searched for and the row number of the node is then memorized as follows:

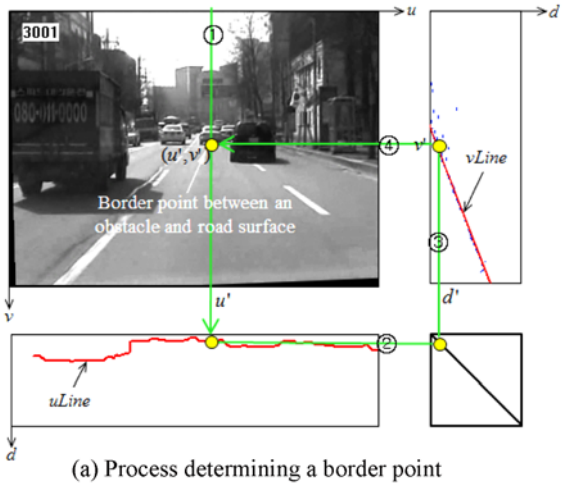
$$J = \underset{j}{\operatorname{argmax}} G(N_{Col-1,j}), \forall j \quad (18)$$

The backward search is continually carried out from the node $N_{Col\text{-}1,J}$ until the column number is less than the row number of a node being searched. This is because it is impossible for the row number representing a disparity d defined by $d = x_L - x_R$(in which x_L and x_R are column numbers of left and right images, respectively), to be larger the column number, $u = x_L$. Figure 16 shows the optimal path named *uLine* detected in $S(u,d)$.



(a) *uLine* in $S(u,d)$ (b) *uLine*

Figure 9. Detected *uLine*.



(a) Process determining a border point

(b) Border line and free space

Figure 10. Free space detection.

2.5. Free Space Detection

For the final step of extracting the boundary between the road surface and the obstacles, we use the *vLine* and the *uLine* as shown in Figure 10: 1) a column, u', is selected on the image; 2) this u' meets the *uLine* and determines a disparity, d'; 3) becoming a column of the v-disparity image, this d' meets the *vLine* and determines a row, v'; 4) (u', v') becomes a border point between an obstacle and the road surface. If this looping is applied to all the columns of the input image, a border line is created as shown in Figure 10 (b). The bottom part of this border line is the free space.

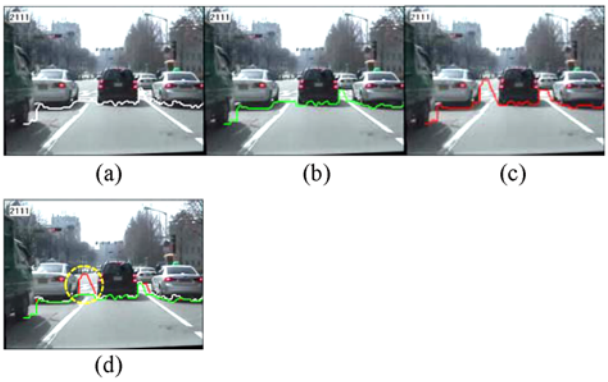
3. EXPERIMENTAL RESULTS

3.1. Comparison of $H_u^o(u,d)$, $S_o(u,d)$ and $S(u,d)$ for Free Space Detection

We compared $H_u^o(u,d)$, $S_o(u,d)$, and $S(u,d)$ in detecting the free space by applying the same optimal path finding algorithm. While they provided very similar results, except for the area indicated by a circle in Figure 11 (d), $S(u,d)$ gave the best result in terms of the definition of 'free space' as used in this paper. In Figure 11 (c), the border lines of the free space obtained from $H_u^o(u,d)$, $S_o(u,d)$, and $S(u,d)$ were overlapped on the input image.

3.2. Selection of Parameter Values

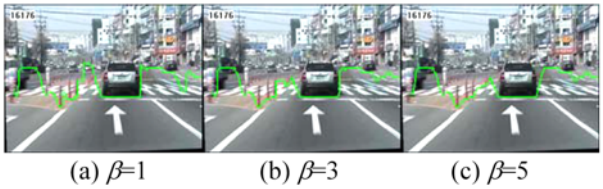
The parameters relating to *uLine* detection includes ω_g in Equation (16) and b in Equation (17). Experiments to select proper parameters were performed on many images by



(a) (b) (c)

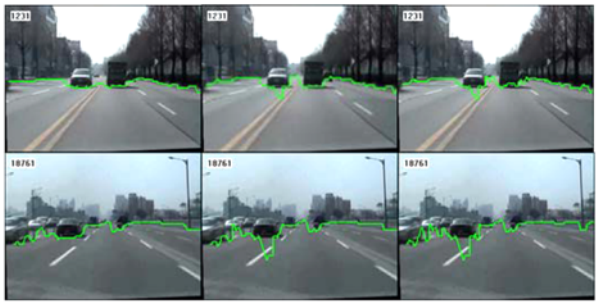
(d)

Figure 11. Comparison of $H_u^o(u,d)$, $S_o(u,d)$, and $S(u,d)$ for free space detection: (a) Result from $H_u^o(u,d)$; (b) Result from $S_o(u,d)$; (c) Result from $S(u,d)$, and (d) Overlap of three results.



(a) β=1 (b) β=3 (c) β=5

Figure 12. Border line of free space depending on β.



(a) ω_g=0.5 (b) ω_g=1 (c) ω_g=2

Figure 13. Border line of free space depending on ω_g.

altering the parameter values. For β, we chose β = 3 among the three cases, including β = 1, β = 3, and β = 5 through experimental results, as shown in Figure 12. Similarly, for ω_g, we also performed many experiments as shown in Figure 13 and chose ω_g = 0.5. In Figures 12 and 13, the blue line on the images represents the border line between obstacles and the road surface.

3.3. Comparison of Free Space Detection with the Result from Laser Scanner

The results from the proposed algorithm were compared with the results from a laser scanner (SICK LMS291-S05 with 180° for field of view, 0.25° for angular resolution, and 0 m~80 m for operating range). The scanner was installed at the height of 51.3 cm on the front bumper and at the center of the ego vehicle. The stereo cameras were DFK 21BU04 (6 mm lens, 42 cm for base line, and 5.37 mm for focal length). The cameras were also installed behind the room mirror in the ego-vehicle. The images used in this experiment have a disparity range of from 1 to 59. Table 1 shows part of the measurement depths corresponding to the disparity range.

Figure 14 shows the results of free space detection using the proposed algorithm and the laser scanner. In the figure, the result from the proposed algorithm is differentiated from the result from the scanner by colors of border lines of free space drawn on the images. The white line represents the border line from the scanner and the red line represents the border line from the proposed algorithm. While both lines seem to be quite close on each input image, they have a gap specifically at a far distance. As shown in Table 1, as the depth becomes increasingly longer, one pixel difference of disparity covers a larger range than in a nearby distance.

Table 1. Depth (meters) according to disparity, *d*.

d	depth	*d*	depth	*d*	depth	*d*	depth
1	151.96	6	25.99	11	14.19	16	9.75
2	77.19	7	22.29	12	13.01	17	9.18
3	51.73	8	19.51	13	12.01	18	8.67
4	38.89	9	17.34	14	11.15	19	8.21
5	31.36	10	15.61	15	10.40	20	7.80

Figure 14. Comparison of free space detection using the proposed algorithm and a laser scanner.

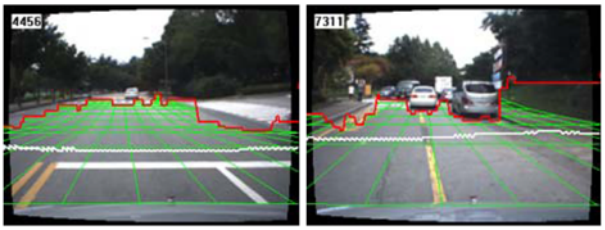


Figure 15. Free space detection by the proposed algorithm and a scanner under a pitch motion of the ego-vehicle.

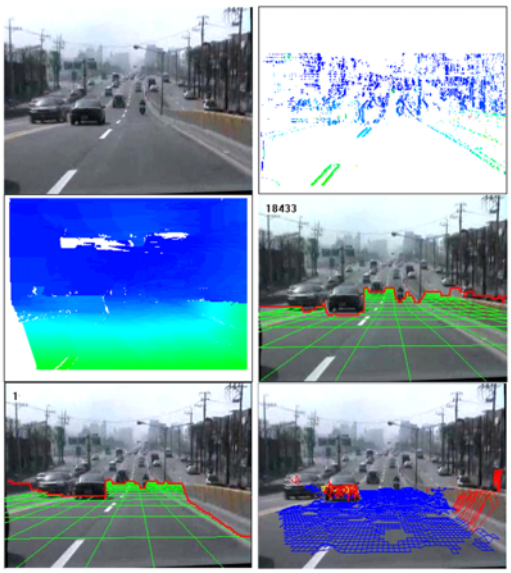
On the other hand, the scanner has better resolution than the stereo cameras. This causes a gap to appear at a far distance. From the first two results, we can see that the laser scanner did not detect the sidewalks as obstacles because the installation height is higher than the height of the curbs. In addition, the light emitted from the scanner is reflected on the surface of the vehicles as shown in the last result, which leads to an error in the free space detection.

The following results also compare the proposed algorithm and the laser scanner in coping with the pitch motion of the ego-vehicle. As shown in Fig. 15, the proposed algorithm is more robust in detecting the free space than the scanner for the pitch motion. When an abrupt pitch motion occurs, the scanner detects the road surface as an obstacle.

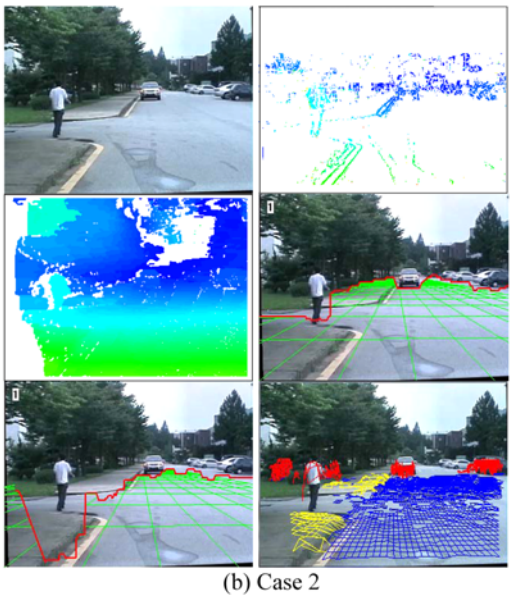
3.4. Comparison of the Proposed Algorithm with Other Algorithms

This comparison was performed by our algorithm and Oniga and Nedevschi's algorithm (2010). Figure 16 shows the results for the two cases: a general traffic road and a campus road. For each case in the figure, the left image from a pair of stereo images and a sparse disparity map constructed by the algorithm in (Lee and Lee, 2011) are shown on the left and right sides of the first row, respectively, a dense disparity map constructed by the algorithm in Song and Lee (2013) and a free space extracted from the sparse disparity map are shown on the left and right side of the second row, respectively, and a free space extracted from the dense disparity map by the proposed algorithm and the result of Oniga and Nedevschi's algorithm are shown on the left and right sides of the third row, respectively. In the result of Oniga and Nedevschi's algorithm, the green, red, and yellow meshes represent the road surface, obstacles, and traffic isles, respectively. In Figure 16, case 1 contains a concrete wall on the right side and case 2 contains sidewalks on the left side. In both cases, the surfaces of the wall and sidewalks are poorly textured, and the sidewalk has a height of less than 13cm. The wall and sidewalks are obstacles in terms of vehicle navigation. However, when we used the sparse disparity map as shown in these cases, the wall and sidewalks were included within the free space. Therefore, a study needs to be performed to determine how to enlarge the disparity pixels from such walls and sidewalks even for a sparse disparity map. On the other hand, when we used the dense disparity map in detecting the free space, the proposed algorithm and Oniga and Nedevschi algorithm detected the free space successfully.

However, as mentioned in section 1, Oniga and Nedevschi's algorithm is time consuming and requires only dense disparity map. In addition, their method is very limited in the detection depth.



(a) Case 1



(b) Case 2

Figure 16. Comparison of the proposed algorithm and Oniga and Nedevschi's algorithm.

3.5. Various Results of Free Space Detection

In Figure 17, we installed a 2 meter high upright virtual wall along the detected border line between the road surface and the obstacles to display the detected border line realistic for various images. In addition, we performed free space detection for the images acquired under various weather conditions. In Figure 18, we showed the results obtained from such images. The results were robust,

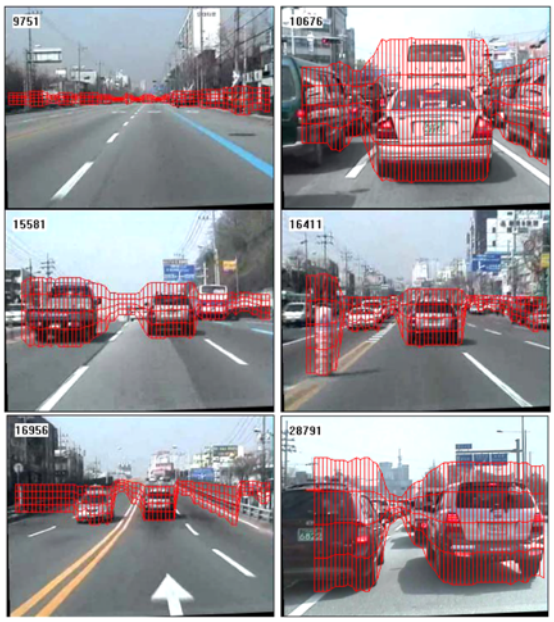


Figure 17. Representation of free space using a virtual wall for various images.

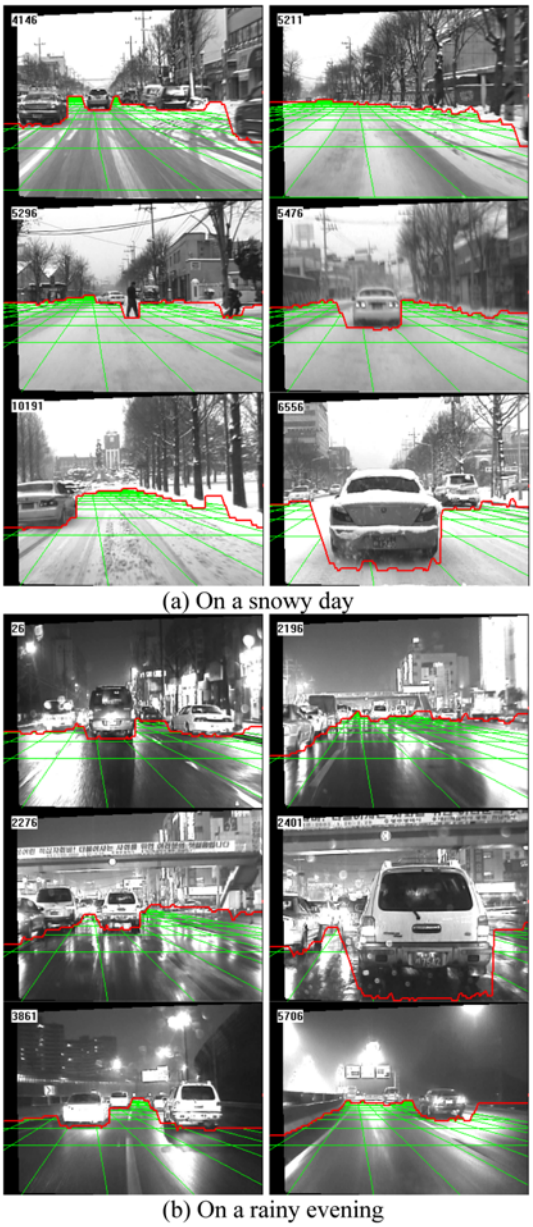


Figure 18. Free space detection under various weather conditions.

regardless of weather conditions.

3.6. Processing Time

The proposed algorithm was implemented using MFC Visual C++ on a personal computer with an Intel core 2 quad 2.4GHz with 2.0GB ram. The processing time of the estimation of free space detection was 6.0 ms per frame. The processing time was therefore very short.

4. CONCLUSION

Based on stereo vision, the proposed algorithm robustly detects the free space in front of the ego-vehicle for autonomous vehicle navigation. The algorithm formulates the detection process as the optimal path-finding problem and uses Dynamic programming to solve the problem. The algorithm uses images of 320×240 pixels and shows a detection performance of 6.0 ms per frame on a general PC.

As most applications based on stereo vision are affected by whether or not disparity is obtained, the proposed algorithm is also influenced by disparity generation in estimating free space. Therefore, future improvements to this algorithm should focus on supplementing a more accurate and dense disparity map.

The experimental results given in this paper showed the robustness of free space detection, which was the fundamental goal of our algorithm. As expected, the proposed algorithm detected the free space successfully, regardless of the weather conditions and the number, location, and type of obstacles.

ACKNOWLEDGEMENT–This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (grant number: 2011-0012323).

REFERENCES

Badino, H., Mester, R., Vaudrey, T. and Franke, U. (2008). Stereo-based free space computation in complex traffic scenarios. *IEEE Southwest Symp. Image Analysis and Interpretation*, 189–192.

Bertozzi, M. and Broggi, A. (1998). GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. Image Processing* **7**, **1**, 62–81.

Hillier, F. S. (2006). *Introduction to Operations Research*. 8/E. McGraw Hill. New York.

Kubota, S., Nakano, T. and Okamoto, Y. (2007). A global optimization algorithm for real-time on-board stereo obstacle detection systems. *IEEE Intelligent Vehicle Symp*., 7–12.

Labayrade, R., Aubert, D. and Tarel, J. P. (2002). Real time obstacle detection in stereovision on non flat road geometry through "V-disparity" representation. *Proc. IEEE Intelligent Vehicles Symp*., 646–651.

Lee, K. Y., Park, J. M. and Lee, J. W. (2014). Estimation of longitudinal profile of road surface from stereo disparity using dijkstra algorithm. *Int. J. Control, Automation, and Systems* **12**, **4**, 895–903.

Lee, K. Y. and Lee, J. W. (2011). Extracting of corresponding points of stereo images based on dynamic programming. *J. Institute of Control, Robotics and Systems* **17**, **5**, 397–404.

Lee, K. Y., Lee, J. W. and Houshangi, N. (2009). A stereo matching algorithm based on top-view transformation and dynamic programming for road-vehicle detection. *Int. J. Control, Automation, and Systems* **7**, **2**, 221–231.

Oniga, F. and Nedevschi, S. (2010). Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. *IEEE Trans. Veh. Technol.* **59**, **3**, 1172–1182.

Song, G. Y. and Lee, J. W. (2013). Accurate dense stereo matching of slanted surfaces using 2D integral images. *Proc. ICVS*, 828–833.

Tsai, R. Y. (1987). A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robotics Automat* **RA-3**, **4**, 323–344.

Wedel, A., Franke, U., Badino, H. and Cremers, D. (2008). B-spline modeling of road surfaces for free space estimation. *IEEE Intelligent Vehicle Symp*., 828–833.