Vision-based navigation using top-view transform and beam-ray model

Qing Lin
Qingdao University
22 PUBLICATIONS **152** CITATIONS

Vision-Based Navigation Using Top-view Transform and Beam-ray Model

Qing Lin, Youngjoon Han, Hernsoo Hahn
Electronic Engineering Department, Soongsil University, Seoul, Korea

***Abstract*—This paper presents a vision-based navigation method for blind people in an outdoor sidewalk environment. Unlike many existing navigation systems that use stereo-vision based methods, the proposed method is able to get obstacle position as well as user motion information by using just single camera fixed at the belly of the user. To achieve this goal, a top-view image of the road is used for obstacle detection and user motion estimation, based on which a grid map is generated for navigation. Obstacles are detected by using a beam-ray model, while user motion is estimated by using optical flow in a user surrounding area. For navigation part, a step score is calculated on the grid map for evaluating the safety of next moving step. Experiments with several sidewalk video-clips show that the proposed navigation method is able to provide useful guidance instructions under certain sidewalk environments.**

I. Introduction

Recently, many electronic navigation devices have been developed to provide assistance to sight-impaired people in a certain local environment. These systems can be categorized depending on how the information is gathered from the environment and how to deliver to the user[1]. Information can be gathered with ultrasonic sensor, laser scanners, or cameras, and user can be informed via auditory[2]-[4] or tactile device[5]-[7]. Most existing vision-based navigation systems have been developed using stereo vision methods. In these systems, stereo cameras are used to create a depth map of the surrounding environment, and then this depth map is transformed into stereo sound or tactile devices for use of self-navigation by visually impaired people.

For instance, Meijer[4] developed a navigation device to transform depth map into a kind of stereo sound space, and delivered to the user via headphones. While the TVS system developed by Johnson and Higgins[5] transformed depth information into a tactile belt with 14 vibration motors laterally attached. The depth map is sliced into 14 vertical regions and each vibration motor is assigned to one region. The distance of the closest obstacle is transformed into vibration so that user can do self-navigation by sensing the vibration pattern of the array. Also the Tyflos navigator system[7] converts depth map into vibration sensing on a 2-D vibration array/vest that is attached on the user's abdomen or chest. The element of the array that vibrates represents the direction, where an object is detected and the different vibration levels represent the distance of the object.

Although many stereo-vision based travel-aids systems have been proved to be effective under certain environment, there are still some problems existed. First of all, due to the high computation load of stereo vision algorithm, most systems just simply run stereo vision algorithm and directly convey the depth information to the user without doing any further obstacle detection and avoidance algorithm. As a result, users have to do obstacle avoidance themselves by sensing and judging the transformed auditory or tactile pattern from the depth map, which makes the system less easy to use and requires much user training. Secondly, user's self-walking motion is seldom considered in the existing stereo-vision based system, which is obviously very helpful for the autonomous navigation. In addition, stereo cameras require camera synchronizations as well as special equipment for keeping the relative positions between two cameras, while single camera system is much simpler, cheaper and easy to carry.

To handle these problems existed in the stereo-vision based navigation system, a navigation method using just single camera and top-view transform is proposed in this paper. A general flowchart of the whole

Manuscript received September 15, 2011. This work was supported the Brain Korea 21 Project in 2011, and by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-2011-(C1090-1121-0010)).

Qing Lin is with the Electronic Engineering Department, Soongsil University, 156-743, Seoul, Korea

Youngjoon Han is with the Electronic Engineering Department, Soongsil University, 156-743, Seoul, Korea

Hernsoo Hahn is with the Electronic Engineering Department, Soongsil University, 156-743, Seoul, Korea

algorithm is shown in Fig.1.

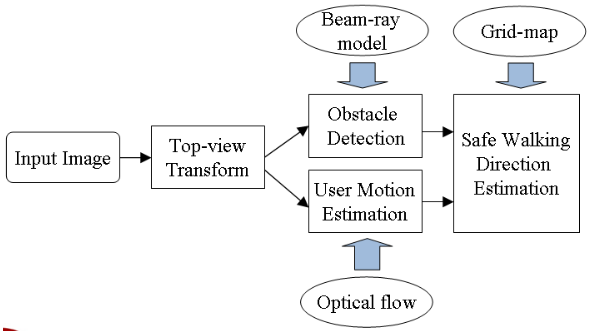


Fig. 1. General flowchart of the whole algorithm.

II. TOP-VIEW TRANSFORM

The first step of the whole algorithm is to map the input image onto a virtual plane which is located at the top of the road plane. This top-view transform will bring several advantages. First of all, the bottom point of erect obstacle is much easier to detect on top-view, which is very important for obstacle localization in single camera metrology. Secondly, obstacle edges are stretched while the road surface edges in the near field are suppressed. With more obvious bottom points and stretched edge components, it would be easier to detect obstacle position on the top-view. Finally, since perspective effect is removed on top-view image, the moving speed of image pixels with respect to the user's motion is more uniformly distributed from bottom to the top of image, which makes the user motion estimation much easier.

Inverse perspective mapping (IPM) [8] is used to generate a top-view from the input image. The estimation of the camera parameters is required for the application of the IPM transform.

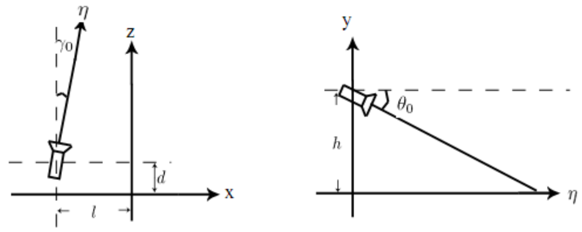


Fig. 2. Camera model for top-view transform.

Mathematically, IPM can be modeled as a projection from a 3D Euclidean space W, containing elements (x, y, z)∈R^3, onto a 2D planar subspace of R^3, denoted by I, with elements (u, v)∈R^2.

The transformation from W to I is given as:

$$u(x,0,z)=\frac{[\arctan(\frac{h\sin r(x,0,z)}{z-d})-(q_0-\partial_u)](m-1)}{2\partial_u} \quad (1)$$

$$v(x,0,z)=\frac{[\arctan(\frac{z-d}{x-l})-(r_0-\partial_v)](n-1)}{2\partial_v}$$

The important parameters in (1) are as follows:

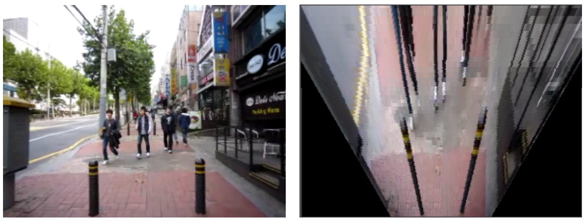
① Camera viewing point is defined by camera position in world coordinate system C = {l, h, d}.

② Camera viewing direction is described by two angles r_0 and θ_0, as shown in Fig.2.

③ Camera angular aperture is $2\alpha_u$ in row direction and $2\alpha_v$ in column direction. $\alpha_u = \arctan(\mu_0 / F)$, $\alpha_v = \arctan(\nu_0 / F)$, (μ_0, ν_0) is the camera's focal center, and F is focal length.

④ Camera resolution is n × m.

By using (1) and filling the corresponding camera parameters, the transform between perspective view and top-view domains can be easily carried out. Fig.3 shows an example of top-view transform using IPM.



(a) Perspective-view image (b) Top-view image

Fig. 3. An example of top-view transform.

III. OBSTACLE DETECTION

A. Beam-ray Model

In sidewalk environment, the most common obstacles are thin and erect obstacles like trees, poles and other pedestrians. These obstacles appear in a kind of beam-ray pattern on top-view image. As is shown in Fig.4, on top-view image, this beam-ray pattern is composed of two parts: the bottom point and two edge components ejecting from the bottom point. Therefore, the detection of obstacles is composed of two steps. The first step is to detect candidate bottom points, and then the edge components associated with each candidate bottom points are checked.

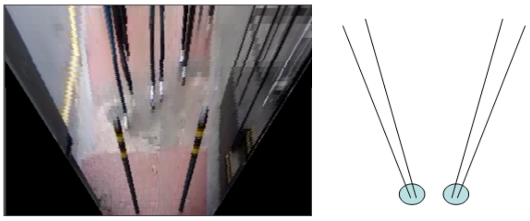


Fig. 4. Beam-ray model for obstacle on top-view domain.

B. Candidate Bottom Points Detection

It can be observed that the bottom points usually have different gray level values with its neighborhood pixels on the top-view image. To model this kind of feature, scale-invariant feature transform (SIFT) is generally used on the perspective view domain. However, in terms of top-view domain, the scale and rotation variation is not obvious. Therefore, to reduce the computation load of SIFT, these bottom points are just modeled as local extreme points of its neighborhood in terms of gray level values on the top-view image.

To extract these local extreme points, we group the pixels in a 5 by 5 window, and then fit a second order polynomial using the 25 points in this 5 by 5 neighborhood.

$$f(x,y) = p_1 + p_2 x + p_3 y + p_4 x^2 + p_5 y^2 + p_6 xy \tag{2}$$

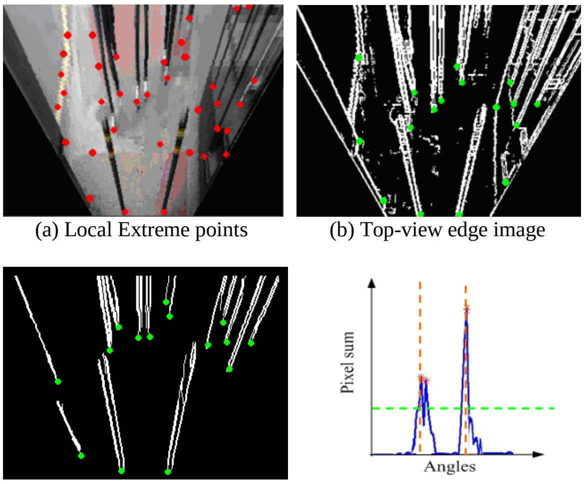
As is shown in (2), second order polynomial *f(x, y)* include six coefficients. Once the 6 coefficients of this polynomial function have been determined, the partial derivatives of this polynomial function *f(x, y)* can be checked using Hessian matrix *H(f(x,y))*:

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \tag{3}$$

The eigen-values of Hessian matrix correspond to the curvature of the gray value surface. A point is accepted to be a local extreme point if the absolute values of both eigen-values of the Hessian matrix are greater than predefined threshold. Fig. 5 shows an example of local extreme points extracted on top-view image.

C. Edge Component Verification

After these local extreme points are extracted, the edge components associated with each candidate bottom points are checked for beam-ray model verification.



(a) Local Extreme points (b) Top-view edge image

(c) Edge-link image (d) Edge-link histogram

Fig. 5. Obstacle detection using beam-ray model.

As is shown in Fig.5(c), at each candidate bottom point, an edge-linking operation [10] is performed to link edge pixels in one basic direction. The direction of edge-link is described by unit-direction and main-direction. The unit-direction is defined according to the orientation of one pixel's 8 neighborhoods, as shown in Fig.6. Based on the definition of unit-direction, the main-direction of edge-link is defined as the mean value of all unit directions between every two neighboring pixels. The example of main-direction calculation is also shown in Fig.6.

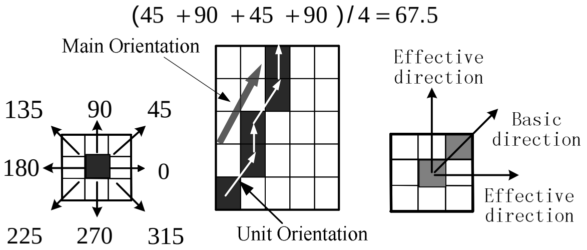


Fig. 6. The definition of edge-link direction.

One edge-link process continues until no more connected edge-pixels are found or the unit-direction of the next connected edge-pixels falls out of effective direction range. Effective direction is defined as the unit direction within ±45°to the basic direction. In this case, the current edge-link is terminated, and another new edge-link will be initialized from the candidate bottom point. After all the edge-links related to one candidate bottom point is found, the main-direction and length of each edge-link is checked.

On top-view image, the edges of vertical obstacles should lie in the direction of an imaginary connection line joining the camera projection on the ground plane and the basilar part of the obstacle in top-view image, as is illustrated in Fig. 7. According to this property, the main direction for obstacle edge-links can be estimated based on the position of the candidate bottom points. Therefore, a valid direction range can be defined for each candidate bottom point. If the edge-link's main direction falls out of this valid direction range, then this edge-link is discarded.

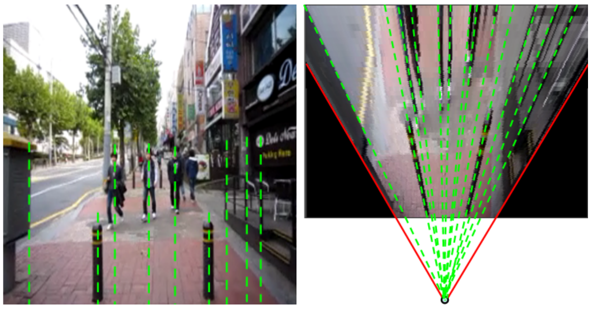


Fig. 7. Obstacle orientation on top-view domain.

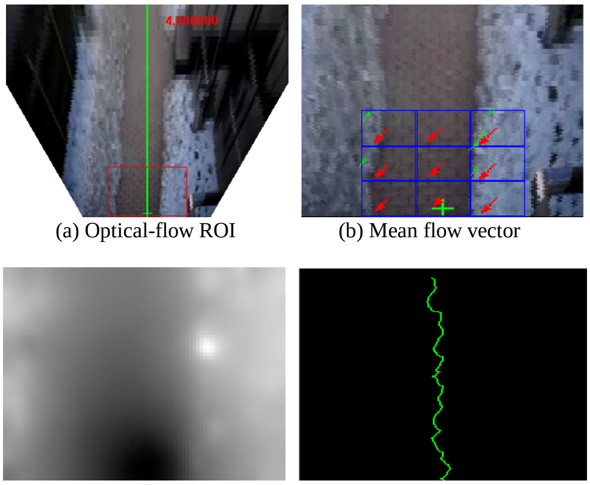
When this edge-linking process is completed for each candidate bottom point, an edge-link histogram is calculated for each candidate bottom point, as is shown in Fig.5(d). The horizontal axis represents edge-link direction while the vertical axis is edge-link length. If two distinct peaks are detected on the histogram, and the peak angle difference lie in a reasonable range, then this bottom point is accepted as real obstacle bottom point, otherwise the points are discarded.

IV. Safe Walking Direction Estimation

A. User Motion Estimation

Assuming that user is located at the bottom center of the top-view image, and a neighboring area close to the user's position is supposed to be composed of mainly road surface pixels. Therefore, we select a rectangular area around center bottom position for evaluating the optical-flow vector field as shown in Fig.8 (a).

Considering that the walking speed of user is not fast, and the ROI is very close to the user, there will be very little illumination variations for the corresponding pixels in consecutive images of an image sequence. In this way, constancy of the gray values, which is a very important assumption in optical-flow, can be ensured. In addition, in this walking scenario, there will not be big displacements. Based on these considerations, combined local-global method proposed by Bruhn, Weickert, Feddern, Kohlberger, and Schnörr [9] is used for calculating the optical-flow vector field.



(a) Optical-flow ROI (b) Mean flow vector

(c) Optical-flow field (d) Estimated user trajectory

Fig. 8. User motion estimation using optical-flow.

For each pixel location in this rectangle area, a motion vector can be obtained, the vector field strength is shown in Fig.8(c). As is shown in Fig.8(b), this region is divided into 9 sub-regions, and the mean motion vector is calculated in every sub-region, then by comparing the differences between those 9 mean vectors, we group the similar ones into clusters, and use the mean vectors included in the largest cluster to calculate final mean vectors. This final mean vector is then used to represent user's walking speed and walking direction. The user is supposed to walk in a speed proportional to the length of final mean vector, while in an opposite direction, by estimating user's walking speed and direction in each frame, the user walking trajectory can be generated, as is shown in Fig.8(d).

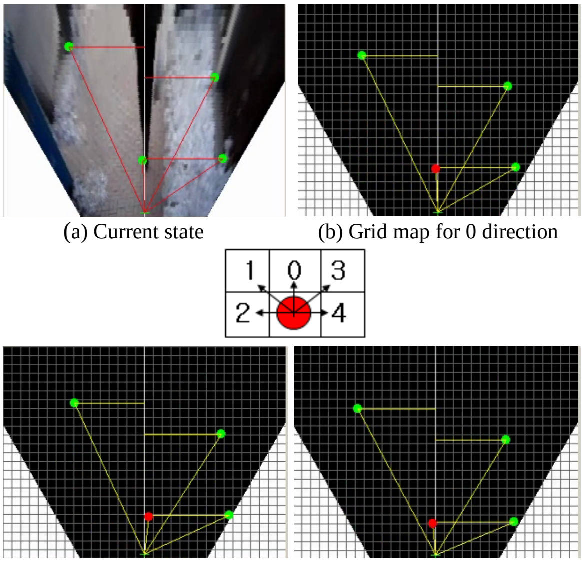
Since the camera is attached on user's belly, the camera will show some swing motions due to the movement of human body. These swing motions turn out to be noise motions added to the user motion. Comparing these noise swing motions to the situation when user is really moving to the left or right, it can be observed that the noise swing motions usually show a relatively small deviation to the left or right, and deviation to the left and right will appear alternatively. This is due to the fact that, the left and right swing motion of user body while walking usually only last for a very short period of time, and these left and right swing motions will appear in turn due to the walking gesture of human body.

Based on these observations, it is not difficult to smooth these swing noise motions, and extract real user motion. Here a one dimensional Kalman filter is used to filter slight deviations to the left or right direction. As most of these slight deviation observations are caused mostly by camera swing motions.

B. Safe Walking Direction Estimation

By knowing the obstacle positions as well as user's walking motion, the next step is to estimate safe walking direction based on a top-view grid map. The basic idea is to generate a top-view grid map for each possible moving direction that we defined. And then on the grid map, for each defined moving direction, a direction score is calculated, while the direction with the highest score is selected as the suggested safe moving direction for the next step.

Grid map is defined by n×n grids. The value of n is equivalent to the average human step length. The position of user in the graph is assumed to be located at the middle bottom of the image plane, while the obstacle locations are labeled using small circles on the graph.



(a) Current state (b) Grid map for 0 direction

(c) Grid map for 1 direction (d)Grid map for 3 direction

Fig.9. Grid map for possible moving directions.

On the grid map, five moving directions are defined, as is shown in Fig.9. Direction 0 indicates straight, direction 1 for left 45° , direction 2 for hard left, direction 3 for right 45° , and direction 4 for hard right. Based on the knowledge of user's motion in the scene, a grid map can be generated for each possible moving direction after a certain period of time. Fig.9 shows an example of grid map generation, where one grid map is generated for each possible moving direction.

Firstly, the grid map for 0 direction is generated, and direction score is calculated. If the direction score for 0 direction is larger than a warning threshold, then the user is suggested to keep the straight moving direction. Otherwise, the grid map for other moving directions will be generated and direction scores are calculated. And the direction with highest score is chosen as the next suggested moving direction. However, if the largest direction score is still less than a predefined "stop" threshold, then "stop" instruction is generated instead.

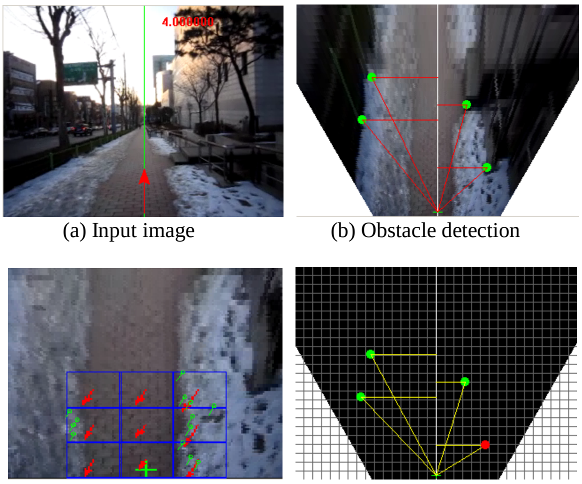
The direction score is calculated by considering the smallest triangular area formed by obstacle, user and user's walking direction. As shown in Fig.9, for each obstacle, a triangular can be formed. The smaller triangular area is, the more dangerous this obstacle will be; therefore, this smallest triangular area can be used as a direction score for evaluating the safety of the current walking direction. On the grid maps shown in Fig.9, the obstacle with smallest triangular area is considered as the most dangerous obstacle and is labeled in red color.

V. Experimental Results

The whole algorithm is developed using C++ under windows platform. To test the performance of the algorithm, we attached a camera on a belt and fix it at user's belly, looking a little bit downward to the road ahead of user. The camera captures images of the road environment, which is then used to test the algorithm. The algorithm is tested in different sidewalk environment around our campus.

Fig. 10 shows an example of the output result. In this scene, the detected obstacle bottom point is shown in Fig.10(b) while user motion is estimated by optical-flow field shown in Fig.10(c). Then based on the results of Fig.10(b) and Fig.10(c), grid map for the 0 direction is generated, and step score is calculated. The step score in this scene is higher than the alarming threshold, therefore, user is supposed to move straight forward. The recommended moving direction is indicated by red arrow shown in Fig.10(a).

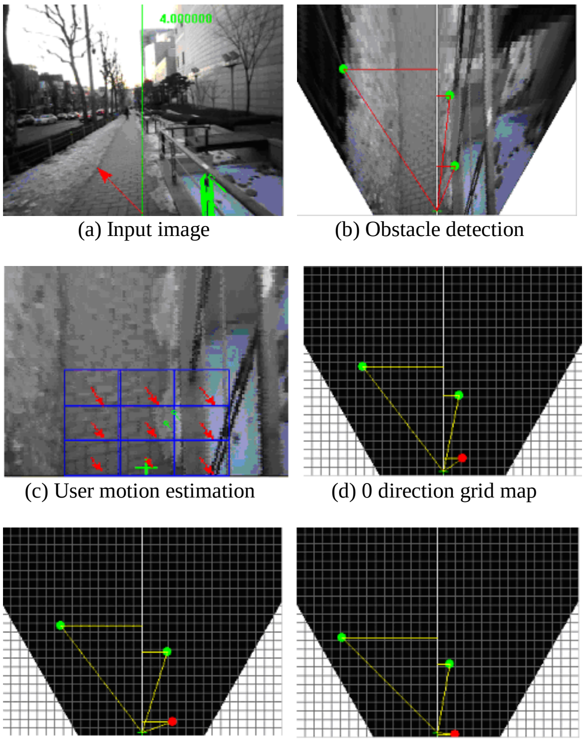
Another scene of experimental result is shown in Fig.11. In this scene, the grid map for straight direction is first generated, and the smallest triangular area is calculated. However, the smallest triangular area is less than the predefined warning level, therefore grid map for other moving directions are generated, and step scores for these moving directions are calculated, as is shown in Fig.11(e)(f). Since the left 45 degree moving direction has the largest step score on grid map, therefore, the suggested moving direction for next step is left 45 degree.



(a) Input image (b) Obstacle detection

(c) User motion estimation (d) Grid map generation

Fig. 10. Experimental results example 1#.



(a) Input image (b) Obstacle detection

(c) User motion estimation (d) 0 direction grid map

(e) 1 direction grid map (f) 3 direction grid map

Fig. 11. Experimental results example 2#.

The performance of the algorithm is evaluated in two parts: obstacle detection accuracy and safe walking direction accuracy. To evaluate the obstacle detection accuracy, we test the algorithm on a challenging side-walk video clip, with complex road side structures and cluttered road surface. Some sample images of the scene are shown in Fig.10 and Fig.11.

2000 frames from this video clip are used for testing. All the critical obstacle positions are manually labeled on the top-view images of these 2000 frames. If the obstacle is detected within a 10 pixels deviation from the ground truth position, then it is accepted as correct detection, otherwise it is counted as false detection. If the obstacle is not detected, it is counted as miss detection. If some road surface clutters are detected as obstacles, it is regarded as false detection. The obstacle detection result is shown in Table 1.

In the real testing, the Hessian matrix threshold in bottom point detection step is tuned to be a little lower to reduce the miss detection cases, since miss detection may cause serious mistakes in the following navigation steps. Therefore, miss detection case happens only when obstacle bottom part look very similar to the road surface in terms of gray level values.

TABLE I
OBSTACLE DETECTION RATE

Total obstacles	Correct Detection	False Detection	Miss Detection
9486	7788(82.1%)	1660(17.5%)	38(0.4%)

To evaluate the safe walking direction accuracy, we generate a virtual user walking trajectory by using the estimated safe walking direction and estimated user's self walking speed. And this virtual walking trajectory is then mapped to the top-view grid map with all the obstacles positions labeled. A segment of this synthesized map is shown in Fig.12.

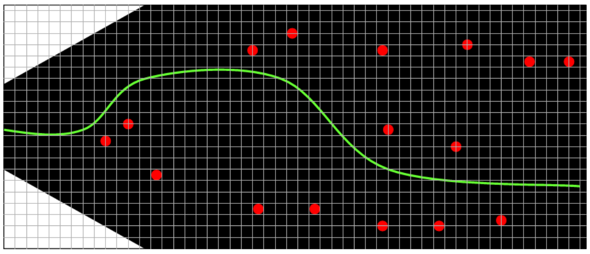


Fig. 12. Synthesized walking trajectory and grid map.

VI. CONCLUSION

A vision-based navigation algorithm for blind people in outdoor environment is proposed in this paper. Compared with general stereo vision based method, the proposed system handles this problem using just single camera. It not only can detect various erect obstacles with simple beam-ray model on top-view image, but also takes user motion into consideration, which can provide more helpful information for navigation. The proposed algorithm can be used in a travel-aid system for blind pedestrians in an outdoor sidewalk environment. The future work will involve developing an information transformation scheme to transform the information obtained from the image domain to the language domain, and deliver the navigation instructions via audio messages to the blind user in an appropriate way.

ACKNOWLEDGMENT

This work was supported by the Brain Korea 21 Project in 2011, and by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2011-(C1090-1121-0010)).

REFERENCES

[1] D. Dakopoulos and N. G. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: a survey," IEEE Trans. Syst., Man, And Cybern., vol. 40, no.1, pp. 25–35, Jan. 2010.

[2] G. Sainarayanan, R. Nagarajan, and S. Yaacob, "Fuzzy image processing scheme for autonomous navigation of human blind," Appl. Softw. Comput., vol. 7, no. 1, pp. 257－264, Jan. 2007.

[3] A. Hub, J. Diepstraten, and T. Ertl, "Design and development of an indoor navigation and object identification system for the blind," in Proc. ACM SIGACCESS Accessibility Computing, no. 77－78, pp. 147－152, Jan. 2004.

[4] P. B. L. Meijer, "An experimental system for auditory image representations," IEEE Trans. Biomed. Eng. vol.39, no.2, pp. 112–121, Feb. 1992.

[5] L. A. Johnson and C. M. Higgins, "A navigation aid for the blind using tactile-visual sensory substitution," in Proc. 28th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., NewYork, 2006, pp.6298－6292.

[6] N. Bourbakis, "Sensing 3D dynamic space for blind," IEEE Engineering in Medicine and Biology Magazine, Vol. 27, No. 1, 2008, pp. 49-55.

[7] D. Dakopoulos, S. K. Boddhu, and N. Bourbakis, "A 2D vibration array as an assistive device for visually impaired," in Proc. 7th IEEE Int. Conf. Bioinf. Bioeng., Boston, 2007, vol. 1, pp. 930–937.

[8] M.Bertozzi and A.Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," IEEE Tranc. Image Proc., vol.7, no.1, Jan. 1998.

[9] Bruhn, Weickert, Feddern, Kohlberger, and Schnörr, "Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods," International Journal of Computer Vision, vol.61, no.3, pp.211–231, 2005.

[10] Qing Lin, Youngjoon Han, Hernsoo Hahn, "Real-time lane departure detection based on extended edge-linking algorithm," in Proc. 2nd Int. Conf. on Comput. Res. and Dev. Kuala Lumpur, 2010, pp. 725-730.