OCCLUSION DETECTION OF VIRTUAL TARGET FOR AUGMENTED REALITY

Gyeyoung Kim , Changjin Suh, Sangjun Lee, Soowon Lee.

School of Computer Science and Engineering, Soongsil University,
511 Sando-Dong, Dongjak-Gu, Seoul, 156-743, Korea
E-mail: {gykim11, cjsuh, sangjun, swlee}@ssu.ac.kr

ABSTRACT

In this paper, we present several techniques for displaying a virtual target, in which a target is integrated into a 3-D real environment. The first result involves formation of wireframe using DEM of the experimental realm and registration with real image using visual clues. In the second effect, we use enhanced snake algorithm for extracting the outlines of object and proportional relation expression for obtaining of 3D information, and the 3D information was calculated at point where occlusion problem for a moving target. We used calibrated data derived from the real image, displayed validity of presented methods under the environment in which partial occlusion occurs.

Key words: *Augmented Reality, Occlusion, DEM, Enhanced snake algorithm, Proportional relation expression*

1. INTRODUCTION

In augmented reality, the computer provides additional visual information that enhances or augments a user's view of the real world. While virtual reality offers a virtual world in which users are completely immersed, augmented reality brings the computer out of the desktop environment and incorporates the computer into the reality of the user. Interest in augmented reality has increased in the past years and is being researched and applied to many areas. Augmented reality has following three characteristics: combines real and virtual, interactive in real time, registered in 3-D[1]. A key problem in the AR field is how to best depict occluded objects in such a way that the viewer can correctly infer the depth relationships between different real and virtual objects [b],(E. Chen, 1995). This paper studied on the development of a realistic simulated training model through the display of virtual targets in the input images of CCD camera mounted in a vehicle. For realistic simulation, it is essential to determine the occlusion areas of the virtual object produces after registering real image and virtual object. But if the accuracy or density of the created map is insufficient to estimate the boundary of object, it is difficult to determine the occlusion area. For solving this problem, we formed a 3D wireframe using the DEM of the experiment area and then registered it to real CCD images with visual clues. And we also presented enhanced snake algorithm for extracting object boundary. Next, to decide the occlusion area regardless of the density of map, we acquire the reference 3D information of the occlusion points using picking algorithm. And then we infer the 3D information of other boundaries using the proportional relations between 2D and 3D. We also use MER (Minimum Enclosing Rectangle) method which compares the rectangle area of the objet in the camera's angle of vision and virtual target for improving performance.

2. MODELING THE REAL WORLD

In order for real and virtual objects to properly interact, they must be placed into the same computational framework. Models of real objects must be first created and brought into the virtual world. We restricted the real environment to some area and constructed that area's scene Model using 3D information. The topographical information DEM (Digital Elevation Model) is used to map the real

ISSN : 1978-774X

world coordinates to each point of the 2D CCD image. DEM has information on the latitude and longitude coordinates expressed in X and Y and heights in fixed interval. The DEM used for this experiment is a grid-type DEM which had been produced to have the height information for 2D coordinates in 1M interval for the limited experiment area of 300 m x 300 m. The DEM data are read to create a mesh with the vertexes of each rectangle and a wireframe with 3D depth information.

3. BOUNDARY EXTRACTION ALGORITHM BY NEIGHBORING EDGE SEARCH

3.1. Enhanced Boundary Extraction Algorithm

The Snake algorithm (Lilian, 2002), (Charles, 2006) is a method of finding the outline of an object by repeatedly moving to the direction of minimizing energy function from the snake vertex input by user. But existing snake algorithm cannot accurately extract the contour information when the object form is complex because the direction of the energy function appears as a composite vector of the current, previous, and the next snake points, and shrinks toward the center of these points. To solve this problem, this paper proposes a method to form a edge map using the Gradient Vector Flow (GVF) algorithm (C. Xu, 1997),(Xu, 1998),(Xu, 1998), and add a new energy term that indicates the distance between the searched edge point and snake point so as to extract an accurate contour. The GVF algorithm can measure the contour of complex objects using the gradient of edge, and move to the concave contour regardless of initialization. Further, the gradient vector of the edge map has a larger value as it is near edge, and approaches zero as it is farther. This paper uses the edge information of the gradient vector flow to search the proximal edge point, and when there is an edge, adds a new energy term ($E_{edge\text{-}distance}$) that shows the distance from the reference point to the searched edge as equation (1). Here, α, β and γ are all set to 1 without exhaustive adjustment

$$E_{Enhanced\ \ Snake} = \int_0^1 E_{continuity}(v(s)) + E_{curvature}(v(s)) + E_{image}(v(s)) + E_{edge-distance}(v(s))ds \qquad (1)$$

First it searches edge points while rotating around the axis d which is the connection between current and previous snake points v_i and $v_{i\text{-}1}$. In other words, if the angle formed by the three points v_i, $v_{i\text{-}1}$, and $v_{i\text{-}2}$ is ϕ, to prevent the situation where the axis meets with or passes by $v_{i\text{-}2}$ and meets v_i again, it searches the edge point v_i' where the image strength ∇I is greater than the threshold while rotating only by $\frac{\phi}{2}$ and adds a new energy term using the value of the distance d' between v_i and v_i' to the existing algorithm. This paper determined the rotation direction for accurate search by assuming the following two facts: First, it was assumed that the initial snake points form a closed curve that encloses the object. Second, it was assumed that the points were arranged sequentially in one direction. The reason for this was because to search proximal edge, it must move inside the contour, but the direction may be wrong due to the diversity of object forms if simply the direction to the object center is set.

3.2. Calculation of $E_{edge\text{-}distance}$

Fig. 1 is an example of calculating the distance between an arbitrary snake point v_i' and the edge v_i' around it.

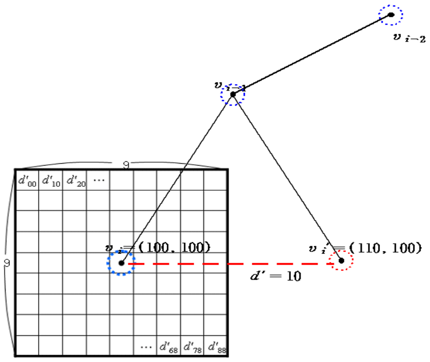


Figure 1. distance between a point of snake and edge

If we surround the arbitrary point v_i with a 9×9 window and assume that its distance with a new edge is d'_{mn}, the height and width of the window are s, and the horizontal and vertical positions of the snake point in the window are m and n, the

d'_{mn} can be obtained with the equation (2) by the Euclidean theorem, and the energy term to be added can be defined as the equation (3) by applying the distance value instead of the brightness value of the image term.

$$d'_{mn} = \sqrt{\left(\frac{2(|v_x - v'_x| + m) - s + 1}{2}\right)^2 + \left(\frac{2(|v_y - v'_y| + n) - s + 1}{2}\right)^2} \quad (2)$$

$$E_{edge-distance} = (|v_i - v'_i| - d'_{min})/(d'_{max} - d'_{min}) = (d'_{mn} - d'_{min})/(d'_{max} - d'_{min}) \quad (3)$$

Added new energy term $E_{edge\text{-}distance}$ is expressed together with continuity and curvature energy terms. In conclusion, the flow of the enhanced snake energy function to which the proximal edge energy function is added can extract the edge exactly in complex situations by approaching the edge more closely. Fig. 2 shows process of proposed boundary extraction algorithm using GVF.

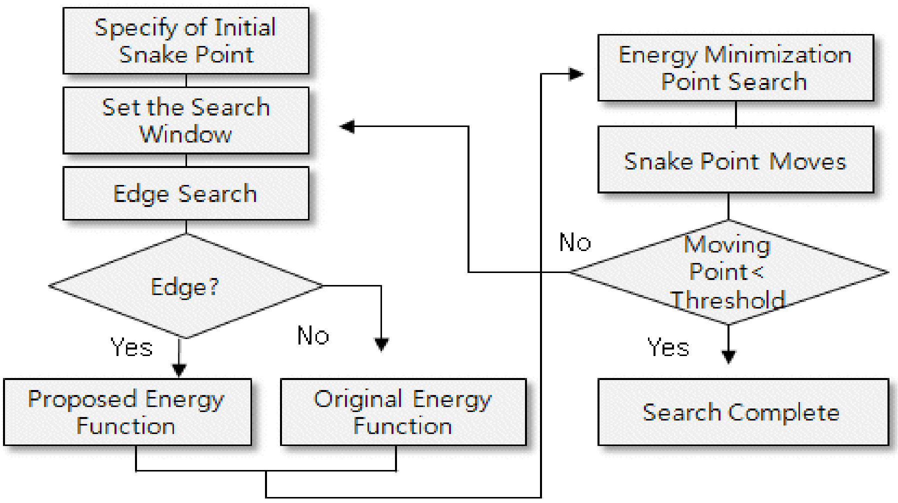


Figure 2. proposed boundary extraction algorithm

4. PICKING ALGORITHM FOR ACQUIRING 3D INFORMATION AND OCCLUSION PROCESSING

4.1. Acquisition of 3D Information Using the Picking Algorithm

In order to acquire the 3D information of the extracted vertexes, this paper used the Picking algorithm which is a well-known 3D graphics technique (Wu, 2003). It finds the collision point with the 3D wireframe created by DEM that corresponds to the points in 2D image and provides the 3D information of the points. The picking search point is the lowest point of the vertexes of the objects extracted from the 2D image. The screen coordinate system that is a rectangular area indicating a figure that has been projection transformed in the 3D image rendering process must be converted to the viewport coordinate system in which the actual 3D topography exists to pick the coordinate system where the mouse is actually present. First, the conversion matrix to convert viewport to screen is used to obtain the conversion formula from 2D screen to 3D projection window, and then the ray of light is lengthened gradually from the projection window to the ground surface to obtain the collision point between the point to search and the ground surface.

4.2. Creation of 3D Information Using Proportional Relational Expression

The collision point, or reference point, has 3D coordinates in DEM, but other vertexes of the snake indicated as object outline cannot obtain 3D coordinates because they don't have a collision point. Therefore, this paper suggested obtaining a proportional relation between 2D image and 3D DEM using the collision reference point and then obtaining the 3D coordinates of another vertex. Fig. 3 shows the proportional relation between 2D and 3D vertexes.

To get P_m that passes the center of the screen using the coordinates of the reference point obtained above, t' must be obtained first. As the t value is given by the picking ray, the given t value and y_B are used to get θ_B and t' is obtained using this θ_B in Expression (4).

$$\theta_B = \sin^{-1}(\frac{\Delta P_{y_B}}{t}),\ t' = |t_B|\cos(\theta_B)\quad t' = |t'| \qquad (4)$$

To get t_m, ϕ_B is obtained from Expression (4) which is the angle between t' and t_m. t_m can be obtained using ϕ_B from Expression (5).

$$\phi_B = \tan^{-1}(\frac{\Delta P_{x_B}}{t'}),\qquad t' = |t_m|\cos(\phi_B) \qquad (5),$$

$$|t_m| = \frac{|t'|}{\cos(\phi_B)}\qquad t_m = |t_m|$$

Because $t_m = p_{z_m}$, $P_m = (0,0,t_m)$.

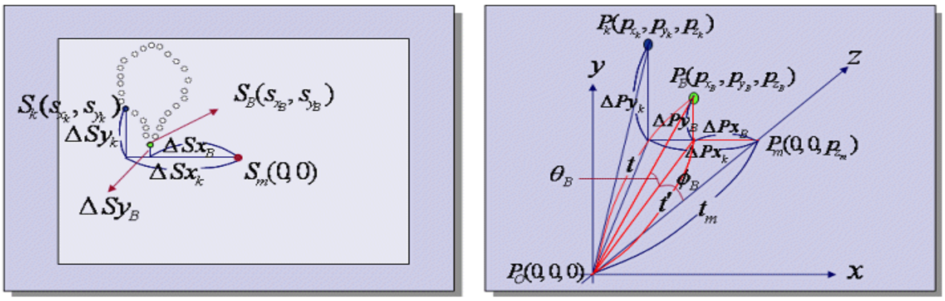


Figure 3. Proportional Relation of the Vertex in 2D and 3D

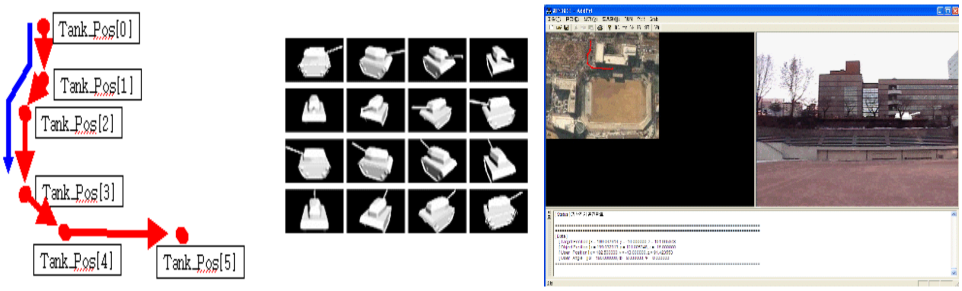


Figure 4. Moving Route Creation and Appearance of Virtual Target(left), Result (right)

4.3. Creation of virtual target path and selection of candidate occlusion objects using MER (Minimum Enclosing Rectangle)

To test the proposed occlusion-resolving algorithm, we created the movement path of a virtual target, and determined the changes of the direction and shape of the target as well as the 3D position of the target. First, the beginning and end points of the target set by instructor were saved and the angle of these two points was calculated, and the direction and shape of the target were updated in accordance with the change of the angle. Further, the remaining distance was calculated using the speed and time of the target, and the 3D coordinates corresponding to the position after movements were determined. Fig. 4 (left) shows movement path of the virtual target which trainee sets and the various virtual targets created to display the targets changing with movement on the image. Fig. 4 (right) shows the result.

5. EXPERIMENTAL RESULTS

We restricted the real environment to some area and constructed that area's scene Model using 3D information. We used experimental vehicle for simulation, we send steering, acceleration, brake data to car driving controller through Bluetooth using remote car controller. Vehicle can be controlled by transmitted data and we can get feedback of present car location data by mounted sensor system. RS232 communicator is interface between vehicle driving controller and sensor fusion system. And it receives instructions from sensor system. CCD camera views the environment. The camera may be static or mobile. In mobile case, the camera might move around by being attached to a vehicle, with their locations tracked by GPS and INS. Table 1 shows frame number of errors occurred per 100 frames depending on camera position change. We can see the low accuracy as position of camera farther away from the object because error range of extracted object's 3D information is great.

Table 1. Camera Position and Accuracy

Location of Camera	Total Error	Accuracy (%)
50M	4	96
150M	12	88
300M	29	71

We tested iteration number and accuracy of proposed energy function. Table 2 shows the result. Proposed algorithm stopped search at the 120th round, and the accuracy was 0.6 while the Kass and greedy algorithms showed the search count 40 and 30 and the accuracy 0.4 and 0.1, respectively. Therefore, we can conclude that the proposed algorithm continued search process and had higher accurate results than existing algorithms. The proposed method is finding optimal contours that have similar properties of edge components with minimal loss. But, existed methods cause errors because of local minima generation. In fact, outdoor images have comparatively low accuracy compared to the indoor images and we could get good accuracy (0.9) when proposed algorithm is applied to simple object like .

Table 2. Iteration number and Accuracy of Energy Function

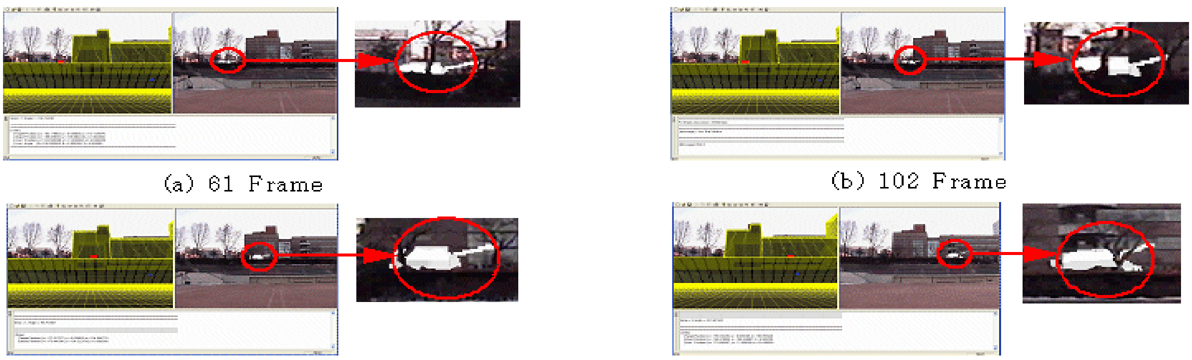
Location of Camera	Proposed Method	Greedy Algorithm	Kass Algorithm
Iteration num. of Energy function	120	40	30
Accuracy	0.6	0.4	0.1

Table 3 shows the speed comparison between the case of using snake vertexes and the case of using the proposed MER to select objects in the image to compare with virtual targets. The proposed method contributed to performance improvement.

Table 3. Speed Comparison

Method	Total frame	Used object	Speed (sec)	Frame per sec.
Snake vertexes	301	10	112	2.687
MER(proposed)	301	10	67	4.492

Fig. 5 shows the virtual images moving along the path by frame. We can see that as the frames increase, it is occluded between the tank and the object.



(a) 61 Frame (b) 102 Frame
(c) 155 Frame (d) 301 Frame

Figure 5. Experimental Results of Moving and Occlusion

ISSN : 1978-774X

6. CONCLUSION

In this paper, we proposed several methods to solve the problem of occlusion that occurs when virtual targets are moved along the specified path over an actual image. We created 3D virtual world using DEM and coordinated this using camera images and visual clues. Moreover, the Snake algorithm and the Picking algorithm were used to extract an object that is close to the original shape to determine the 3D information of the point to be occluded. To increase the occlusion processing speed, this paper also used the method of using the 3D information of the MER area of the object, and proved the validity of the proposed method through experiment. In the future, more research is required on a more accurate extracting method for occlusion area that is robust against illumination as well as on the improvement of operation speed. We also hope to study more in real time environment and to overcome complicated factors that were beyond our control, such as sensor error in the current settings, the brightness difference of same image.

7. ACKNOWLEDGEMENT

This research was supported 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-2010-C1090-0803-0006) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012047687).

8. REFERENCES

(a) Bimber, O. and Raskar, R., (2005). Spatial Augmented Reality: A Modern Approach to Augmented Reality, Siggraph, Los Angeles USA

(b) J. Yong Noh and U. Neumann. (2001). Expression cloning. In SIGGRAPH'01, pages 277-288.

(c) E. Chen. (1995). Quicktime VR-an image-based approach to virtual environment navigation. Proc. of SIGGRAPH.

(d) Lilian Ji, Hong Yan, (2002), "Attractable snakes based on the greedy algorithm for contour extraction", Pattern Recognition 35, pp.791-806.

(e) Charles C. H. Lean, Alex K. B. See, S. Anandan Shanmugam, (2006). "An Enhanced Method for the Snake Algorithm," icicic, pp. 240-243, First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06).

(f) C. Xu and J.L. Prince, (1997). “Gradient Vector Flow: A New External Force for Snakes,” Proc. IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR), Los Alamitos:Comp. Soc. Press, pp. 66-71.

(g) Xu. Chenyang, and J. L. Prince, (1998). "Snakes, Shapes, and Gradient Vector Flow," IEEE Transactions in Image Processing, vol. 7, no. 3, Mar 1998.

(h) Xu. Chenyang, and J. L. Prince, (1998). "Generalized Gradient Vector Flow External Forces for Active Contours," Signal Processing, vol. 71, issue. 2, Dec., pp. 131-139.

(i) Wu, S.-T., Abrantes, M., Tost, D., and Batagelo, H. C. (2003). Picking and snapping for 3d input devices. In Proceedings of SIBGRAPI 2003, 140-147.