Removing Foreground Objects by Using Depth Information from Multi-View Images

Jaeho Lee and Changick Kim
School of Engineering, Information and Communications University
Munji-dong, Yuseong-gu, Daejeon, Korea

ABSTRACT

In this paper, we present a novel method for removing foreground objects in multi-view images. Unlike the conventional methods, which locate the foreground objects interactive way, we intend to develop an automated system. The proposed algorithm consists of two modules: 1) object detection and removal, and 2) detected foreground filling stage. The depth information of multi-view images is a critical cue adopted in this algorithm. By multi-view images, it is not meant a multi-camera equipped system. We use only one digital camera and take photos by hand. Although it may cause bad matching result, it is sufficient to detect and remove the foreground object by using coarse depth information. The experimental results indicate that the proposed algorithm provides an effective tool, which can be used in applications for digital camera, photo-realistic scene generation, digital cinema and so on.

KEYWORD: Image segmentation, Stereo matching, Multi-view images, Disparity/depth, Inpainting.

1. INTRODUCTION

Removing unwanted foreground objects is important in the digital picture/cinema production because obtaining foreground-removed background can be a critical issue. For example, protecting bars or fences surrounding a cultural property may need to be erased for better visual effect. But this procedure is not easy because we do not know the location of the foreground objects and the texture of the region which is concealed by foreground objects.

Existing works in this area can be placed into two classes. In the first class, several images are used to reconstruct the full background scene [1-4]. For example, Diego Ortin *et. al.* [2] present the generation of occlusion-free artificial view for the realistic texturing of 3D models. They reconstruct the geometric model of image by using several images at different view. Cormac Herley [3] combines the image sequence at the same view position to remove occlusions. At the different position of image sequence, occlusions vary continuously. So the “good” data that is the unobstructed view of the arch can be used for reference. In the second class, occlusion region is interpolated in image inpainting area [5-7]. Nielsen *et. al.* present an interactive system for removing undesirable object in digital pictures [5]. After selecting an undesirable object in image, a hole-filling technique is initiated to generate a seamless background portion. In [7], Criminisi *et. al.* propose for filling in the hole that is left behind in a visually plausible way. The actual color values are computed using exemplar-based synthesis.

But in most prior works, although they show good performance there are still some drawbacks. In the first case which uses several images, object region can not be selected automatically in image [2]. Or they use image sequence at fixed position. In this case only moving object can be detected automatically [3]. In the other case, inaccurate information can be used to fill the occluded region in image inpainting area [5, 7]. If the area to be filled is wide, the object-removed image may be somewhat unnatural. So in this paper, we propose a foreground object removal algorithm that operates automatically by using multi-view images without using any additional information.

As mentioned, our work consists of two parts:

1. Object detection and removal: Since the multi-view images system provides different view images, the foreground objects can be detected by using depth information. Note that the goal here is to locate the foreground regions, thus a

coarse depth map is enough. This coarse depth information is compared with homogeneous regions, which is obtained by image segmentation. Using this procedure we can detect the foreground object.

2. Detected foreground filling: To generate a foreground-removed background, the detected foreground regions need to be filled with appropriate texture, which can be obtained from the corresponding locations in the nearest view image. Since the depth value is not perfect, the mean disparity is computed from its neighbor pixels for finding corresponding locations.

The rest of this paper is organized as follows. Our object detection and removal module is presented in section 2. In section 3, details of detected foreground filling module are described. Section 4 shows several experimental results, and finally section 5 concludes the paper and brings up some future works.

2. OBJECT DETECTION AND REMOVAL

In this paper, our goal is to remove unwanted foreground objects. With this simple constraint, it is possible that the process of our algorithm works in an unsupervised manner. In this section, we propose an object removal algorithm, which consists of parts: image segmentation, stereo matching, region-based coarse depth map. Each step will be discussed in detail.

2.1 Image segmentation

Image segmentation is useful in many applications for dividing each image into some regions. In our approach, this technique is necessary for detecting accurate boundaries of the object region. Existing work in this area can be categorized into two classes.

One is color-based segmentation [8, 9]. Color quantization and spatial segmentation are critical cues in this approach. In most cases these papers have good performances but their operation is somewhat slow because the image pixels are quantized and replaced by their corresponding color level.

In the other class, a boundary between two regions is main clue. This edge-based image segmentation defines a predicate for measuring the evidence for a boundary [10-12]. Some weights on each edge measure the dissimilarity between pixels. So, similar regions are connected by the degree of variability in neighboring regions of image.

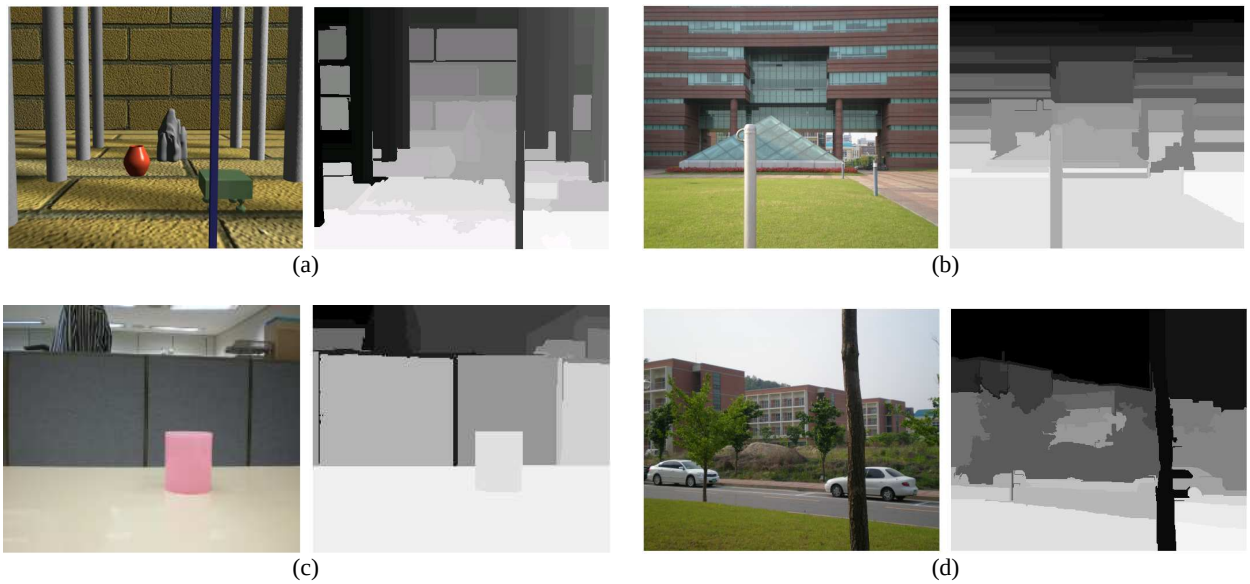


Fig. 1. (a) Inharmony, (b) School, (c) Cup, (d) Dormitory. Right images are the segmented images. There may be some unwanted foreground objects in a scene.

We adopt edge-based image segmentation [11] because this method also has good result and runs in time nearly linear according to the number of edges. First, we obtain edge information by using Sobel mask. Then we compare neighboring pixels at each pixel of edge. If the difference between neighboring pixels is smaller than such threshold, the regions which include that pixel are combined. There are two parameters σ and k. σ is a Gaussian blurring factor and k sets a scale of observation. In other words, the larger σ causes severe blurring effect and the larger k causes a preference for larger components. For a consistent result, we use the parameter value, σ = 1.0 and k = 300 for all test images. Figure 1 shows the segmentation results. Left one is a reference image and right one is a segmented image. The size of all images is 512x384.

2.2 Stereo matching

In multi-view camera system people can feel the sense of distance by the disparity of each object in image. According to the position of object in image, the object which is near from the camera has large disparity whereas the object which is far from camera has small disparity. There are many papers about generating disparity map by using this feature. Most existing works in this area are region-based for efficient and fast process [13-17]. Many attributes, such as pixel intensity, edge, corner and so on, are used for detecting corresponding pixel or area in different image. Note that the camera calibration does not matter in this multi-view camera framework.

As explained, only one camera is used in taking multiple pictures in our approach. Thus, conventional stereo matching algorithms are not suitable for this case which may have non-zero disparity in vertical direction. Since the goal in this paper is to delete foreground objects rather than estimating a dense depth map, just a coarse depth map is enough to achieve the goal.

In general, stereo matching methods dealing with only horizontal displacement are considered to have well-fitted epipole line. In our approach, however, we also allow for vertical disparity. Table 1 summarizes the disparity map generating algorithm in terms of pseudocode.

Table1. Pseudocode of the algorithm for generating disparity map.

```
( Parameter:  search_range_x = 48, searchrange_y = 12, blocksize = 4 )

for x = 0 to image_width {
  for y = 0 to image_height {
    best_matching = max_value;
    for m = x – search_range_x/2 to x + search_range_x/2 {
      for n = y – search_range_y/2 to y + search_range_y/2 {
        sum = 0;
        for a = a – blocksize/2 to a + blocksize/2 {
          for b = b – blocksize/2 to b + blocksize/2 {
            accum += abs(Ref_Img[y – blocksize/2, x – blocksize/2], Neigh_Img[b, a]);
            x++; }
          x = x – blocksize;
          y++; }
        if ( sum < max_value) { then,
          max_value = sum;
          disparity_x = m;
          disparity_y = n; }
        } }
    disparity = sqrt( (x – disparity_x)^2 + (y – disparity_y)^2 );
} }
```

The disparity map of our approach is shown in Fig. 2. Dark pixels denote small disparity values whereas bright pixels denote large disparity values. Although the obtained disparity maps contain erroneous pixels due to textureless region on background, illumination condition by different picture time, non-consensus epipole line, etc., it is shown that the

foreground object is well-located. In the following sub-section, we detect the foreground object using this disparity information.

(a)

(b)

Fig. 2. Disparity map under our system. (a) School, (b) Road.

2.3 Region-Based Coarse Depth Map

The previous two stages yield a segmented reference image and a coarse disparity map. Our next step is to convert the coarse disparity map into a region-based depth map. The region-based depth map is obtained by taking an average depth value on each segmented region, i.e.

$$RV_n = \frac{1}{N_n} \sum_{(x,y)\in R_n} D(x,y) \qquad for \;\; 0 \le n < region\_num$$

R_n means n-th segmented region, RV_n is average depth value of region R_n, N_n is the number of pixels belonging to R_n, and $D(x, y)$ is depth value at each pixel (x, y). The term *region_num* is the number of segmented region. Figure 3 shows the result of region-based depth map.

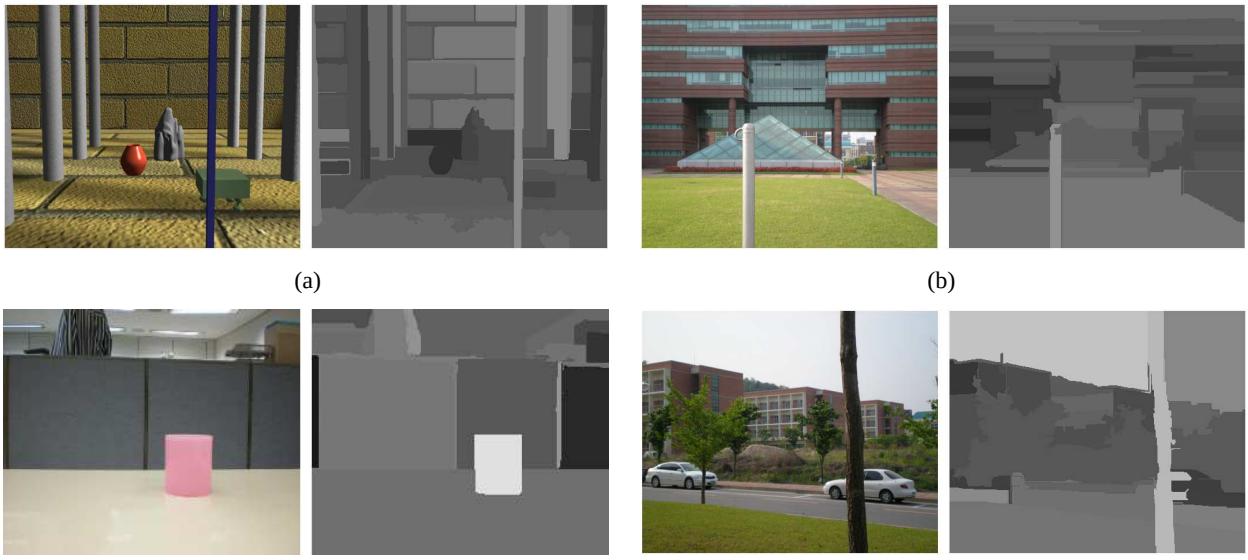
Since foreground object is expected to have the largest depth value, the region with the largest average depth value is selected us the foreground region. We denote a set of coordinates of points in the foreground object as *FO*, then it can be expressed as follows:

$$n_0 = arg \;\; \max_n \; RV_n \;, \;\; 0 \le n < region\_num$$

$$FO = \left\{ (x,y) \in R_{n_0} \right\}$$

Since the right most or left most side of the image may not have the matching region, we impose a constraint that the foreground objects are not located by the image boundary and it is expressed as follows:

$$FO = \left\{ (x,y) \in R_{n_0}, \;\; for \;\; \frac{1}{10} \times width < x < \frac{9}{10} \times width \right\}$$



(a) (b)

(c) (d)

Fig. 3. Reference image and Region-based depth image. (a) Inharmony, (b) School, (c) Cup, (d) Dormitory.

Based on above expression, object map $O(x, y)$ can be defined.

$$O(x,y) = \begin{cases} 1 & for\ (x,y) \in FO \\ 0 & for\ otherwise \end{cases}$$

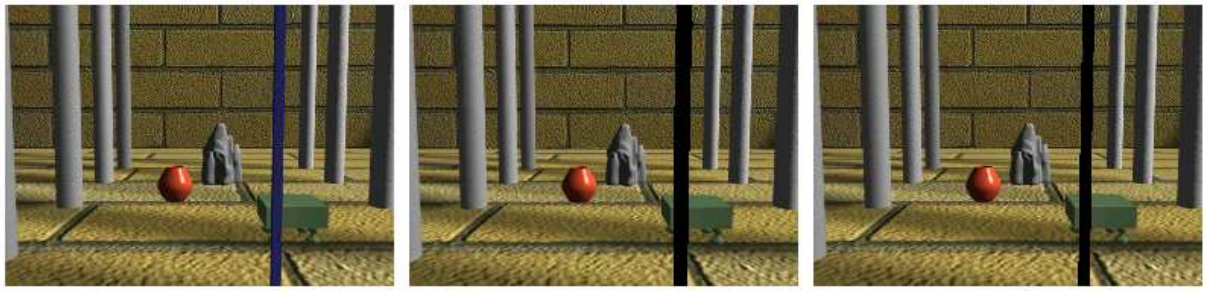
The final decision on the foreground object is marked in black in Fig. 4. Also, Table 2 describes the comparison between experimental result and ground truth map. In our approach, object map is expanded into 4 directions by 3 pixels because the border of the detected object may not be well-matched. Hence, it should be noted that the large false positive rates in Table 2 are due to this expansion. Since the expanded foreground regions need to be filled with corresponding texture in the other image, this expansion is not critical. The critical problem is how to deal with false negatives as shown in Table 2. We see, from experiments, the false negative rate is usually less than 3%. In the case of "School" image, the false negative rate is higher because the hook is segmented as a different region from the pole.

Table 2. Comparison of object detection with ground truth map.

	Inharmony	**School**	**Cup**	**Dormitory**
Total object pixel	8118	4479	8816	11369
False negative (non-detected object pixels)	30	171	4	321
False positive (detected non-object pixels)	1655	612	888	3806
False negative rate (%)	0.37	3.82	0.05	2.82
False positive rate (%)	20.76	17.48	10.07	36.30

3. FILLING DETECTED FOREGROUND OBJECTS

To generate foreground-removed background, the detected foreground regions need to be filled with appropriate texture, which can be obtained from the corresponding locations in the nearest view image. In order to find the disparity values in the foreground regions, we use the disparity values of the neighboring pixels.



(a)

(b)

(c)

Fig. 4. Object removal image. Center image is obtained by our algorithm. Right image is ground truth map by removing object manually. (a) inharmony, (b) school, (c) Cup.

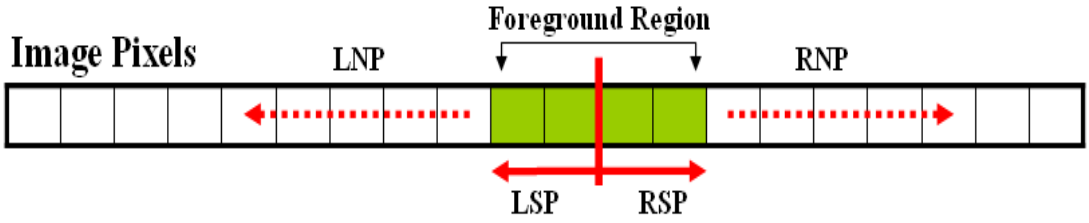
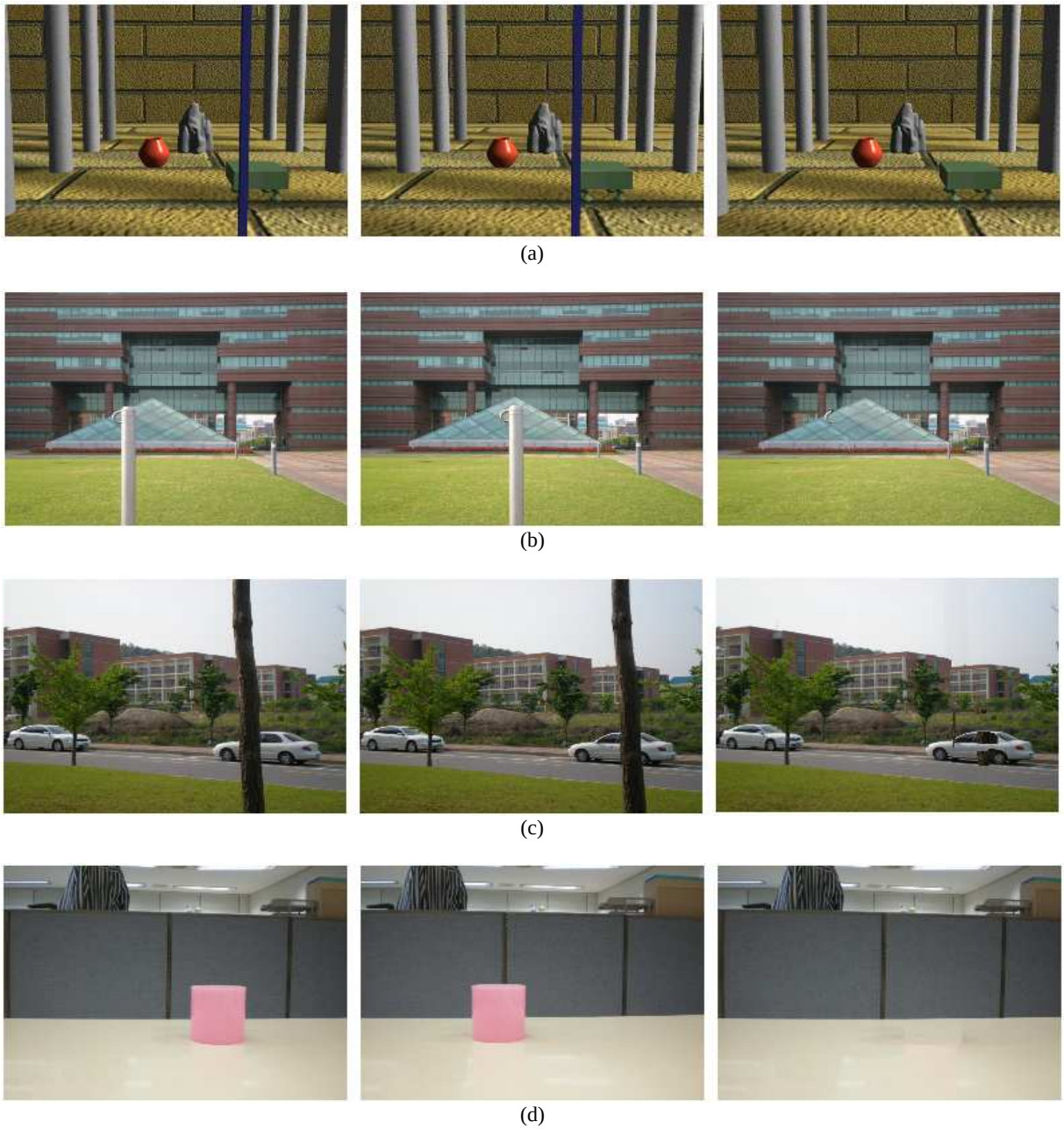


Fig. 5. The method map of filling detected foreground region.

The proposed filling process is simple and fast as shown in Fig. 5, in which pixels in green are detected foreground pixels whereas others belong to background region. Firstly, foreground pixels are partitioned into left side and right side pixels. Secondly, we average the disparity values of the left- and right- nearest five pixels (LNP, RNP). Finally, using these averaged disparity values, the corresponding pixel values of the foreground pixels are taken from the other image.

4. EXPERIMENTAL RESULTS

The proposed full automatic object removal algorithm has been tested on a variety of multi-view images. The size of all test images is 512x384 pixels. The test images are taken with a digital camera, Olympus FE-230. The pictures were captured by hand without using a tripod. A reference picture is taken first, and the image of the other view is taken after moving the camera aside a little. Note that, unlike other object removal algorithms [2, 5] which require a manual selection of the object region, our approach operates automatically owing to using multiple-view images. It is important to note that a perfect stereo camera system is not necessary for our approach.



(a) (b) (c) (d)

Fig. 6. The experimental results. Left and middle images are input multi-view images and the right image is object removed image. (a) Inharmony, (b) School, (c) Dormitory, (d) Cup.

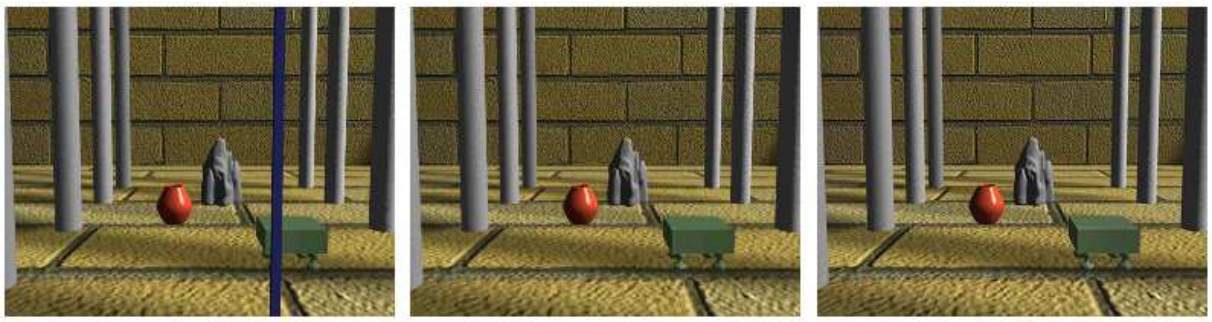


Fig. 7. Comparison of the result of the proposed algorithm (middle) and ground truth obtained manually(right).

The experimental results are shown in Fig. 6. In the "School" and "Dormitory" images, there are some filling errors near the hook and inside a car. This error is mainly due to mis-segmentation (refer to Fig. 1-(b), (d)). Figure 7 shows the comparison with ground truth of "Inharmony" image. We hardly see the difference between them.

5. CONCLUSION AND FUTURE WORK

In this paper, we present a fully automated method for removing foreground object in multi-view image system. Unlike the previous research work [2, 5], in which corresponding points are manually identified, the proposed algorithm allows automatic reconstruction of background since it uses depth information obtained from multi-view images. It should be noted that the proposed system does not require stereo camera setting; just a hand-held camera is good enough. Reducing false negatives are our on-going work. A better segmentation algorithm and efficient inpainting technique are inevitable for improving performance of the proposed system. These absolutely include our future research work.

The proposed scheme will be a useful tool for digital picture and cinema editing. This technique is also appropriate for applications, such as photo-realistic scene generation, content-based coding in MPEG-4 and so on.

ACKNOWLEDGEMENTS

This research was supported by the Ministry of Information and Communication (MIC), Korea, under the ITRC support program supervised by the IITA. (IITA-2006-C1090-0603-0017)

REFERENCES

1. John Boehm, "Multi-Image Fusion for Occlusion-free façade texturing," in *Proceedings of the XXth Conference of ISPRS '04*, vol. XXXV, part 5, pp. 867-872, 2004.
2. Diego Ortin and Fabio Remondino, "Occlusion-free image generation for realistic texture mapping," International Archives of the Photogrammetry, *Remote Sensing and Spatial Information Sciences*, 36(5/W17): 7 pages (on CD-ROM), 2005.
3. Cormac Herley, "Automatic Occlusion Removal from Minimal Number of Images," *IEEE International Conference on Image Processing*, vol. 2, pp. 1046-1049, Sept. 2005.
4. Fausto Bernardini, Ioana M. Martin and Holly Rushmeier, "High-Quality Texture Reconstruction from Multiple Scans," *IEEE Transaction on Visualization and Computer Graphics*, vol. 7, no. 4, pp. 318-332, 2001.
5. Frank Nielsen and Richard Nock, "ClickRemoval: Interactive Pinpoint Image Object Removal," *ACM Multimedia 2005*, pp. 315-318, 2005.

6. Manuel M. Oliveira, Brian Bowen, Richard McKenna and Yu-Sung Chang, “Fast Digital Image Inpainting,” in *Proceedings of the International Conference on Visualization, Imaging and Image Processing* (*VIIP 2001*), pp. 261-266, Sept. 2001.
7. Antonio Criminisi, Patrick Perez and Kentaro Toyama, “Region Filling and Object Removal by Exemplar-Based Image Inpainting,” *IEEE Transaction on Image Processing*, vol. 13, no. 9, pp. 1200-1212, Sept. 2004.
8. S. Belongie, C. Carson, H. Greenspan and J. Malik, “Color- and Texture-Baesd Image Segmentation Using EM and Its Application to Content-Based Image Retrieval,” in *Proceedings of International Conference of Computer Vision*, pp. 675-682, 1998.
9. Yining Deng and B.S. Manjunath, “Unsupervised Segmentation of Color-Texture Regions in Images and Video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no.8, Aug. 2001.
10. O. Pereira Bellon, A. Direne and L. Silva, “Edge Detection to Guide Range Image Segmentation by Clustering Techniques,” *IEEE International Conference on Image Processing*, Kobe, Japan, Oct. 1999.
11. Pedro F. Felzenszwalb, and Daniel P. Huttenlocher, “Efficient Graph-Based Image Segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, Sept. 2004.
12. Sappa, A.D. and Devy, M., “Fast range image segmentation by an edge detection strategy,” in *Proceedings of International Conference on 3-D Digital Imaging and Modeling*, pp. 292-299, 2001.
13. Nelson L. Chang and Avideh Zakhor, “View Generation for Three-Dimensional Scenes from Video Sequences,” *IEEE Transaction on Image Processing*, vol. 6, no. 4, Apr. 1997.
14. T. Kanade and M. Okutomi, “A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920-932, Sept. 1994.
15. C. L. Zitnick and T. Kanade, “A Cooperative Algorithm for Stereo Matching and Occlusion Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, July 2000.
16. Y. Tsin, S. B. Kang and R. Szeliski, “Stereo Matching with reflections and translucency,” in *Proceedings of International Conference on Computer Vision and Pattern Recognition* (*CVPR’03*), vol. I, pp. 702-709. June 2003.
17. C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder and R. Szeliski, “High-Quailty Video View Interpolation using a Layered Representation,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 600-608, Aug. 2004.