Towards a Fast Moving Object Detection Method

Noha Sarhan, Yasser El-Sonbaty, Sherin Youssef
Arabic Academy for Science and Technology
Egypt
noha_sarhan@hotmail.com

ABSTACT: *Depth information plays an important role in many computer vision applications including moving object detection. In this paper, we introduce a fast moving object detection method using Kinect v2. Comparing to Kinect v1, the Kinect v2 provides a higher resolution for RGB images and adopts a ToF (Time-of-Flight) sensing mechanism for depth measurement. The upgraded depth sensing method yields depth images with less noises and holes, which allows us to simplify the pre-processing of eliminating noises and filling holes. Also, the depth information of the Kinect v2 is fruitful enough to detect the moving object without resorting to RGB image. These properties of the Kinect v2 lead us to devise a fast moving object detection method.*

Keywords: Moving Object Detection, Depth Image, Kinect v2, Background Subtraction

Received: 1 March 2015, Revised 29 March 2015, Accepted 4 April 2015

1. Introduction

Moving object detection in a video is a very important task in computer vision, especially for human detection and surveillance problems. Therefore, various algorithms have been proposed to improve the performance of the moving object detection. Depending on the specific target application, the detection methods are classified into online and offline approaches. The online applications [1-4] employ a fast algorithm to achieve a real-time processing, sacrificing the detection accuracy. In addition, the online methods rely on several past frames to estimate a foreground model, which may cause an interleaving region between the adjacent objects. Since only past frames are used for object detection, it is also vulnerable to a sudden appearance of an object in the video. On the other hand, the offline applications often require an exact detection of the moving objects, which forces to use background modeling [5] and background subtraction [6] schemes. One may combine background modeling [5] and background subtraction [6] methods to improve the detection accuracy at a slightly increased cost of the complexity. Basically, the offline approaches require to extract the moving objects rather than the background explicitly, which rather be a major task and usually takes a long execution time.

Most of the moving object detection methods are based on RGB information only, which obviously has a limitation in the accuracy of the object-detection. To improve the detection performance both RGB and depth images have been used. For

example, since the Kinect sensor provides both RGB and depth images, we can exploit the RGB image to fill the holes in the depth image and the hole-filled depth image to detect the moving objects [7]. Note that the algorithm in [7] is designed for Kinect v1 with a structured IR light, where the depth images suffer from noises and holes. On the other hand, the Kinect v2 adopts a ToF (Time-of-Flight) sensing mechanism, which suffers less from the noises and holes than the Kinect v1.

Figure 1 demonstrates the specific differences between the depth images from Kinect v1 and v2. Compared to the depth map of Kinect v1 in Figure 1(b), there are little holes in the depth map of Kinect v2 as shown in Figure 1 (d). This makes it possible to skip the preprocessing for the depth images used for Kinect v1.

Beside the mentioned drawbacks, detecting moving object based only on RGB information is almost an impossible task for night scenes or changing light conditions [8]. Thus, to be robust against the ambient illumination to detect the moving objects we definitely need the depth information. Luckily, as mentioned earlier, the Kinect v2 provides a depth image with a good quality, which allows us to skip the noise removal and hole filling processes. That is, the moving object detection can be achieved by the depth image only for the Kinect v2 without resorting to RGB color information. This in turn speeds up the detection of the moving objects. Specifically, a moving object can be easily extracted by just differentiating depth sequences and updating a reference background depth image in Kinect v2. The rest of this paper is organized as follows. The next section describes the proposed scheme, where we propose a new moving object detection method based on the depth information and reference background depth image. The performance evaluation in Section III shows our experimental results and discussions on the results. We give a conclusion in Section IV.

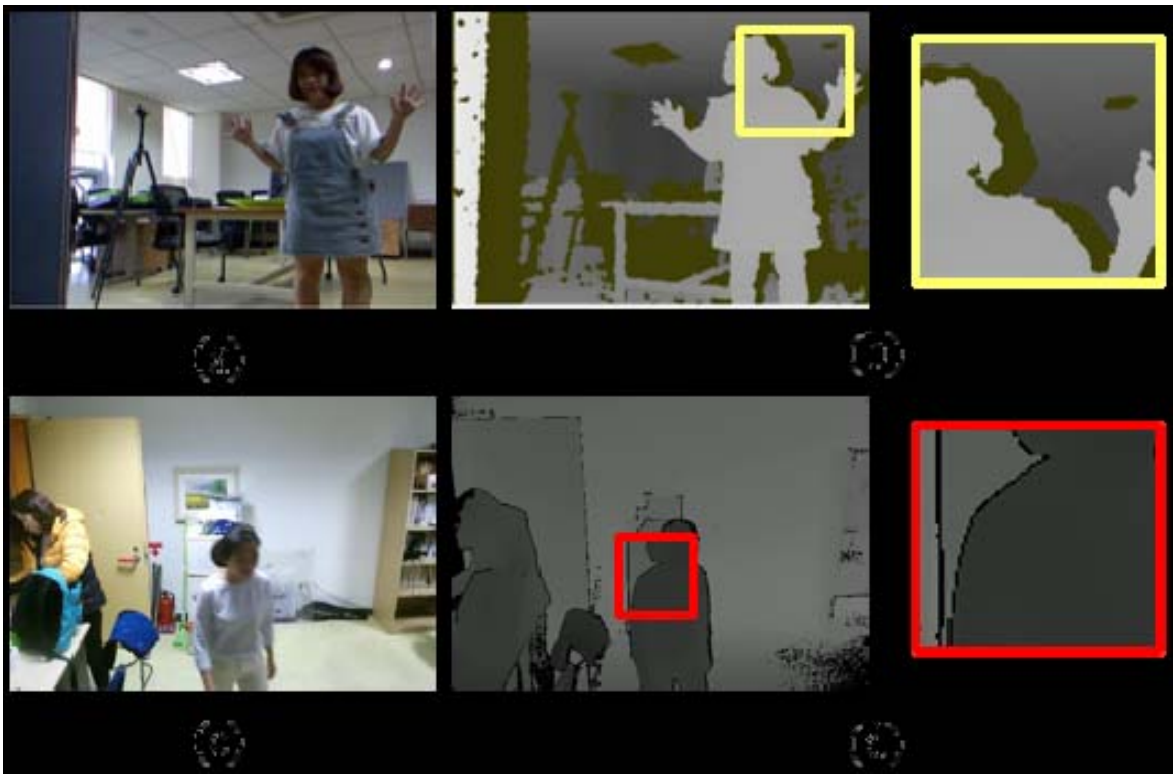

Figure 1. Comparisons of Kinect V1 and Kinect V2. (a) RGB image from Kinect v1 (b) Depth map from Kinect v1 (c) RGB image from Kinect v2 (d) Depth map from Kinect v2

2. Moving Object Extraction With Depth Information

As shown in the overall flowchart (see Fig. 2), we need a reference image *R* as a background model. This reference image is updated whenever we have a new static scene. Modeling the background with RGB information, we usually need a training sequence to determine pixels that belong to background. However, noting that the depth map contains information about the distance to the surfaces of objects from a viewpoint, instead of estimating background model based on a sequence of RGB frames, we can use a reference depth image as a background model.

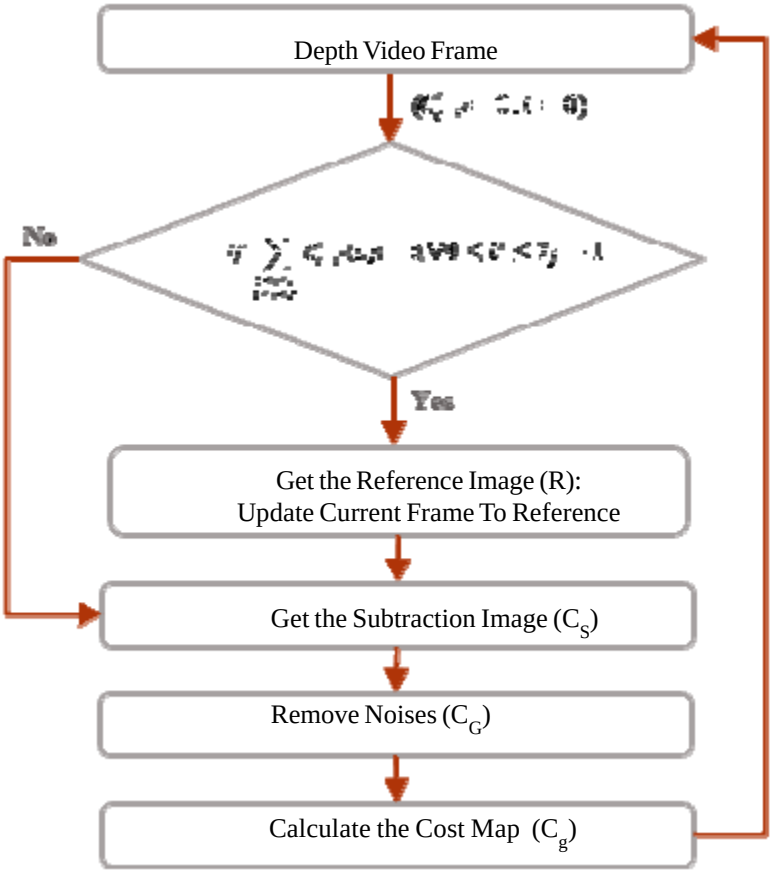

Figure 2. Flowchart of our proposed object extraction approach

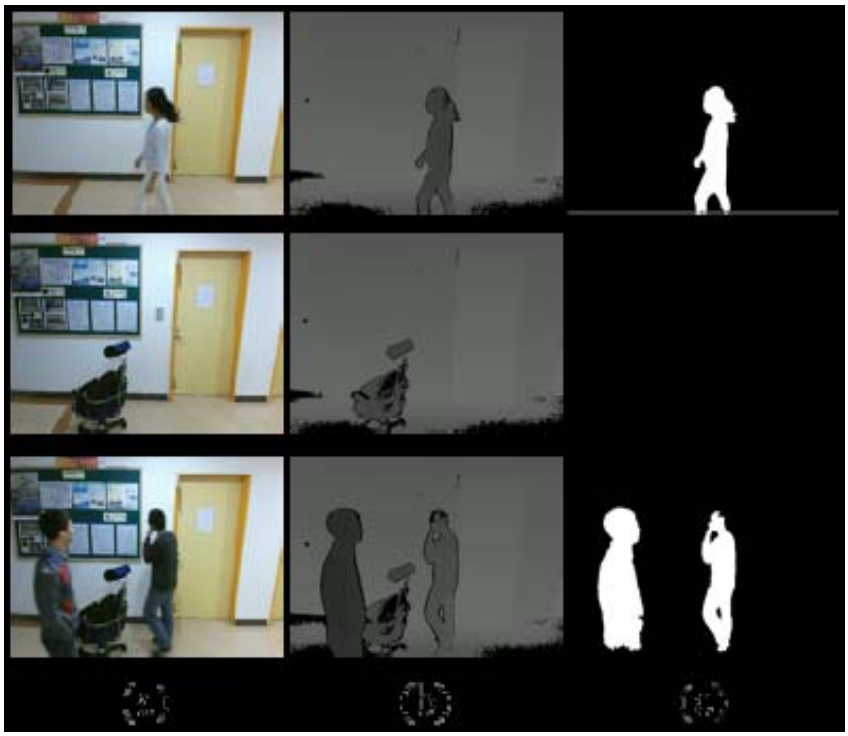

Figure 3. Moving object extraction. (a) Original RGB frames, (b) depth frames, (c) Cost maps *C*

After acquiring the reference image R, we can get the subtraction image C_S by differentiating the depth D_t and the reference image R, where t represents the current frame. To avoid the occluded pixels which appear in the hole regions, a threshold τ_S is used to eliminate the unreliable regions as follows

$$C_S(i,j) = \begin{cases} |D_t(i,j) - R(i,j)|, & if\ D_t(i,j) > \tau_S \,\&\, R(i,j) > \tau_S \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

where $I = \{I_t(i,j) \mid 1 \le t \le N, 1 \le i \le H, 1 \le j \le W\}$ represents the given video sequence, N is the number of frames, and H and W represent the size of each frame and $D = \{D_t \mid 1 \le t \le N, 1 \le i \le H, 1 \le j \le W\}$ is the corresponding depth image. As shown in Fig. 3(b), there are some regions of no depth information. This can happen when we have the reflection problem from the shiny floor. This is the reason why we need a threshold τ_S for removing unavailable and occluded pixels in (1). Now, the subtraction image C_S is to be blurred with Gaussian filter to remove noises as follows

$$C_G(i,j) = \frac{1}{W_G} \sum_{(i',j') \in G_W} G_\sigma(\|(i,j) - (i',j')\|) \times C_S(i',j') \quad (2)$$

where G_σ is a 2-D Gaussian kernel with the standard deviation σ . G_W indicates neighboring pixels of image $C_S(i,j)$ and W_G is the normalization factor which can be defined as $W_G \sum_{(i',j') \in G_W} G_\sigma$. Finally, the cost map C_t of frame I_t is defined as

$$C_t(i,j) = \begin{cases} 255, & if\ C_G(i,j) > \tau_C \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

where C_G is the blurred subtraction image with Gaussian filter from (2) and τ_C is the threshold to eliminate the unconnected pixels after removing noises. Through the above three steps (1)-(3), the cost map C_t can separate the background region from the moving-object region with two values of 0 or 255. An example of the overall flow of our method is shown in Figure 4. Given the reference frame R and current frame as shown in Figure 4 (a) and Figure 4(b), we can obtain the subtraction image C_S from two frames (see Figure 4(c)). Figure 4(d) shows the filtered subtraction image with the Gaussian filter. Lastly, Figure 4 (e) and 4(f) show the cost maps with the subtraction image C_S and C_G, respectively.

Because of the high visual quality, moving object extraction with reference depth image is very useful and effective in video processing applications. Nevertheless, it also becomes an issue in the unforeseen circumstances, for example, a sudden cessation of moving objects. Now, we have to determine whether the suddenly stopped objects belong to the background or not. To this end, we will check the current depth-frame D_t with the previous depth frame D_{t-1} to get a new cost map via equations (1) to (3) with the updated reference image in the iterative process of Figure 2. The difference between two consecutive depth frames is $C_{t\text{-}t'}$' (t:current frame, t' : past frame) with two values of 0 and 255, which denotes the similarity and dissimilarity, respectively. Then, the reference depth image is updated as follows

$$R = \begin{cases} D_t, & if \sum_{\substack{1 \le i \le H \\ 1 \le j \le W}} C'_{t-t'}(i,j) = 0, \forall 0 \le t' \le T_f - 1 \\ R, & otherwise \end{cases} \quad (4)$$

The equation (4) says that the reference depth image R will be updated by the T_f consecutive depth frames only when there is no change in the scene for T_f frames. An example of a sudden cessation of a moving object is shown in Figure 3(b). Note that the moving objects are represented by white region as illustrated in Figure 3(c) and the chair in Figure 3(b) is not considered as a moving object in Figure 3(b).

3. Experimental Results

3.1 Database

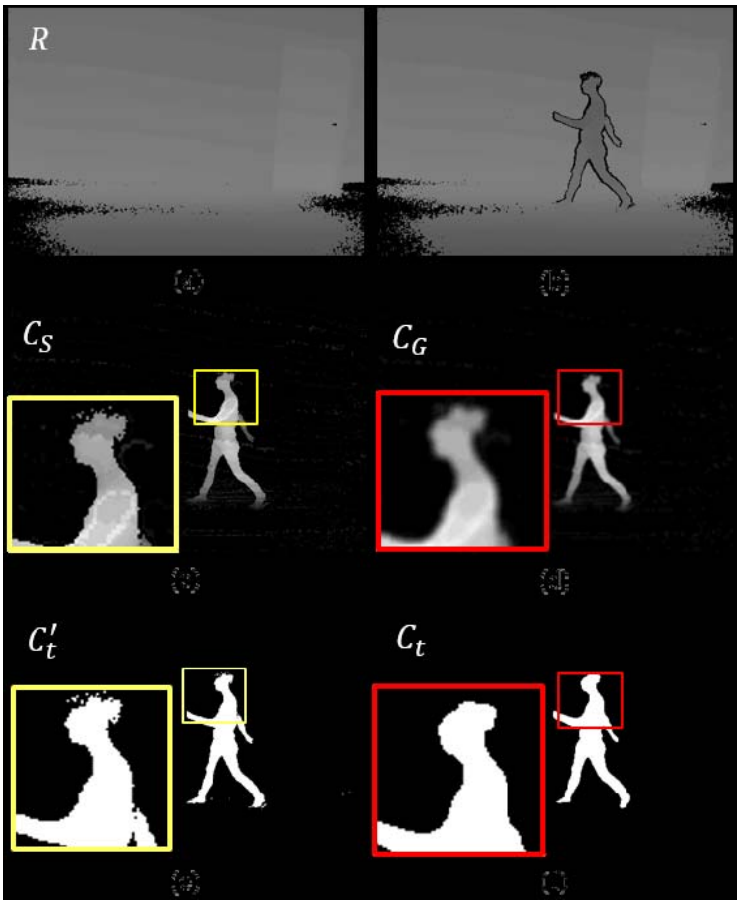

Figure 4. Illustration of each step of our method. (a) reference frame *R*, (b) current frame *Dt* , (c) subtraction image *CS* btained by (1), (d) filtered *CS* frame result of (2), (e) final cost map in (3)

We perform experiments on our Kinect v2 database. Kinect v2 is the new version of the original Kinect (Kinect v1), which adopts a ToF (Time of Flight) technology for depth measurements. RGB and Depth images have higher resolutions in the Kinect v2 than Kinect v1 [9]. Taking advantage of the advanced features of the Kinect v2, we can apply it as an indoor surveillance system and create a RGB-D database including five surveillance videos taken from four locations with different activities and different light conditions with a group of people. The database was recorded on a personal computer with 64-bit window 8, Intel Corei7, CPU 3.60-GHz and 8GB memory. The recorded length of videos is up to 3 minutes. Fig. 5 illustrates some examples of RGB-D frames from our database. As listed in Table I, there are five databases, “Door”, “Board”, “Lab” which are recorded in light scenes and “Night” is recorded in night scene; “Change” is recorded in changing light condition. Kinect has both the RGB and the IR cameras for sensing color and depth images. Because of the IR sensor, the depth can be measured without illumination and it helps to detect object accurately even at nights.

Name	**Length(minutes)**	**# of frames**
“Door”	2:19	4190
“Board”	1:12	2164
“Lab”	2:31	4554
“Night”	2:08	3869
“Change”	2:36	4707

Table 1. Our RGB-D database with Kinect v2

3.2 Results

The proposed method was implemented on a personal computer with 64-bit Window 8, Intel Core i7, CPU 3.60-GHz with 8GB memory. We fixed the parameters for Gaussian distribution and thresholds as follows. The standard deviation σ was set to 2, 5 x 5 window size and τ_s =10, τ_C = 5, which worked well in our experiments. Also we set the number of frame T_f as 10 in our experiments.

First, our foreground extraction method with depth only is compared with the conventional methods [3], [5], and [6] (offline method). Figure 6 (c) shows the result of our proposed method using the depth information. The white regions indicate the detected moving objects which are quite accurate compared to other methods as shown in Figure 6 (d) and Figure 6 (e). Our method is also appropriate for online applications because of the fast execution time as shown in Table II.

Method	**Database**				
	"Door"	"Board"	"Lab"	"Night"	"Change"
Online method[3]	98	51	106	93	123
Offline method[5-6]	62456	31562	68310	52589	71757
Proposed method	50	28	51	45	67

Table 2. Comparisons of execution time (seconds) with foreground extraction methods

Figure 5. Our Kinect v2 database. (a) "Door" , (b) "Board", (c) "Lab", (d) "Night", (e) "Change"

In [3], the background maintenance method, which combined the current frame with the previous background process, takes a short execution time. However, there are some cases with wrong detections. The cause of these errors is the retaining background which is illustrates in Figure 6(d). In addition, the method which contains background modeling [5] and moving object extraction scheme [6] based on MRF model also has the problem due to the object shadows. As shown in Figure 6(e), the visual quality is quite good but the shadow region is incorrectly detected as the object. For both online and offline methods, the detection performance is getting worse as the ambient illumination is getting darker only with RGB information. So, moving object extraction with RGB information only has limits in video surveillance. In comparison, our depth only method runs well for various illumination conditions and the execution time is quite fast.

Table III compares the calculated errors with our ground truth of Figure 7. The error is calculated by equation (5) and represents the degree of differences with the ground truth.

$$error = \frac{\Sigma_{i,j} C_t(i,j) \oplus G_t(i,j)}{H \times W} \qquad (5)$$

where H and W are the height and width of cost map respectively. Comparing our approach with other online and offline methods, it is clear that the errors are decreased drastically.

Error	**Database**				
	"Door"	"Board"	"Lab"	"Night"	"Change"
Online method [3]	8.62	9.81	13.73	12.06	13.05
Offline method[5-6]	6.56	4.32	4.68	7.74	9.93
Proposed method	0.67	0.67	0.56	1.17	1.07

Table 3. Error calculation in comparison with Ground Truth

Figure 6. Results of foreground extraction: (a) original RGB frames, (b) original depth frames, (c) results of our proposed method, (d) results of [3]. and (e) results of [5]&[6]

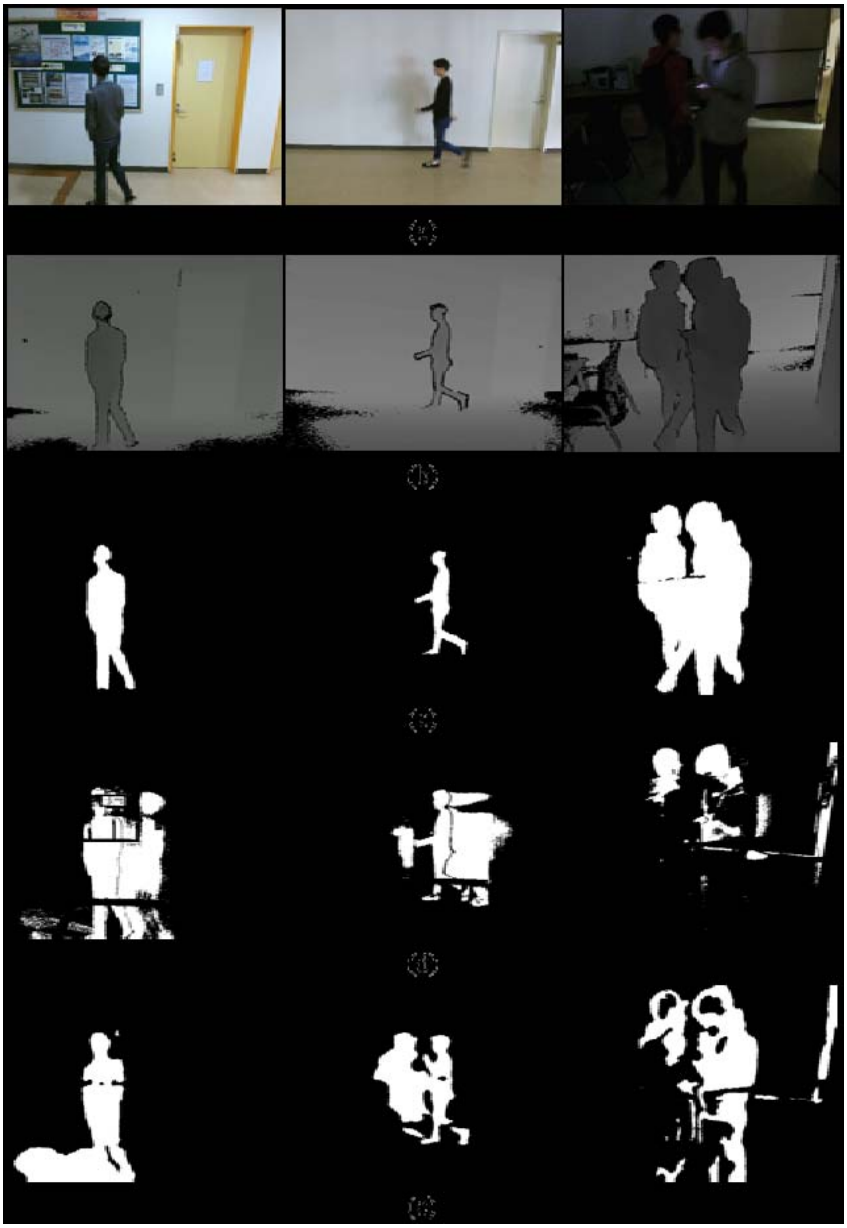

Figure 6. Results of foreground extraction: (a) original RGB frames, (b) original depth frames, (c) results of our proposed method, (d) results of [3]. and (e) results of [5]&[6]

4. Conclusion

In this paper we propose a moving object detection algorithm for Kinect v2. Our detection algorithm is based only on depth information of Kinect v2 and it simply uses the depth frame difference and an updated reference background depth image. Our experiments show that we can detect more accurate moving objects with faster execution times than the previous algorithms for Kinect v2. Since the proposed algorithm relies only on depth information, it can be used for all-day illumination independent surveillance system.

Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF- 2013R1A1A2005024).

Figure 7. Example of Ground Truth data

References

[1] Stauffer, C., Grimson, W.E.L. (1999). Adaptive background mixture models for real-time tracking, *In*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR)*, Fort Collins, 2246-2252.

[2] Sun, J., Zhang, W., Tang, X., Shum, H.-Y. (2006). Background cut, *Lecture Notes in Computer Science – ECCV*, 3952, p.628-641.

[3] Pal, S.K., Petrosino, A., Maddalena, L. (2012). Handbook on soft computing for video surveillance, 1st ed. CRC Press, Taylor & Francis Group, New York, 119-120.

[4] Liao, S., Kellokumpu, V., Pietikainen, M., Li, S.Z. (2010). Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes, *In*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, 1301- 1306.

[5] Elgammal, A., Duraiswami, R., Harwood, D., Davis, L. (2002). Background and foreground modeling using nonparametric kernel density for visual surveillance, *In*: Proceeding of the IEEE, 90 (7) 1151-1163, November.

[6] McHugh, J.M., Konrad, J., Saligrama, V., Jodoin, P.M. (2009). Foregroundadaptive background subtraction, *IEEE signal Processing Letters*, 16 (5) 390-393, May.

[7] Xu, K., Zhou, J., Wang, Z. (2012). A method of hole-filling for the depth map generated by Kinect with moving objects detection, *In*: *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Seoul, 1-5.

[8] Mirror Image. (2010). How Kinect depth sensor works-stereo triangulation?.[Online]. Available http://mirror2image.wordpress.com/ . Retreived March 16, 2011.

[9] Kinect for Windows Team. (2014). The Kinect for Window V2 sensor and free SDK 2.0 public preview [Online]. Available http://blogs.msdn.com