Human Extraction from a Sequence of Depth Images Using Segmentation and Foreground Detection

Lan Anh Trinh
Electronics Engineering Department-
Posts and Telecommunications Institute of Technology
Ho Chi Minh City
lanatrinh2512@gmail.com

Nguyen Duc Thang
Biomedical Engineering Department-
International University-
Vietnam National Unviersity-
Ho Chi Minh City
ndthang@hcmiu.edu.vn

Hoang-Hai Tran
School of Information & Communication Technology-
Hanoi University of Science and Technology
haith@soict.hut.edu.vn

Tran Cong Hung
Science Technology Department,
Post and Telecommunication Institute of Technology-HCM
conghung@ptithcm.edu.vn

ABSTRACT
This paper investigates on an approach of how to extract and track multiple subjects from a sequence of depth images. The Kinect camera is used to obtain a depth image revealing the depth information of a scene. Our proposed system includes the object clustering module to segment different isolated regions correspondent to objects in a depth image and foreground detection module to find moving regions from a sequence of frames. The combination of the two modules let us know which object is moving within a sequence of frames to locate a human subject. In order to extract the depth silhouettes of multiple subjects during time, we propose the use of matching algorithm between two consecutive frames to track their movements. We evaluate the algorithm with a long sequence of frames within a complex environment containing backgrounds with furniture and show how the algorithm is able to precisely extract and separate different human subjects with a fast processing speed. Therefore, the proposed approach is suitable for widely practical applications working with human activity recognition, human pose estimation and human tracking from depth images.

Categories and Subject Descriptors
I.4.6 [**Segmentation**]: *Region growing, Partitioning.*

General Terms
Algorithms, Measurement, Performance, Experimentation, Theory.

Keywords
Human extraction, depth image, clustering, power iteration, foreground detection.

1. INTRODUCTION
During the last decade, automatically extracting and recognizing human subjects from an image has emerged as an important research in computer vision with applications in numerous areas. Typically, human information recognized over a short duration of time provides inputs to control external devices such as computers and games. Meanwhile, a long-term human pose and activity recognition adapts to proactive computing, human health-care, and discovering human lifestyles. Actually, human-healthcare applications are motivated by efforts to address the exponential growing of the world population. People are living longer, increasing amount of older persons within an adult age. Thus, human health-care services especially for the elderly people play an important role more than ever before. For offline healthcare services, a huge amount of patient data is recorded by video camera and analyzed by doctors to suggest a user a lifestyle change or proper diet. In online healthcare, a system which automatically monitors a human and provides proactive action to prevent dangerous event is a preferred solution.

Recently the vision of human aware technology has just become possible although it been receiving many efforts from research communities for a long time ago. In fact, an obstacle for achieving human aware technology is due to the limitations of software and hardware infrastructure. Some attempts developed marker-less systems to estimate human information from a sequence of monocular images (or 2-D Red/Green/Blue images). Because the 3-D information of the subject is lost, the movement of different objects in 3-D space is inaccurately monitored due to ambiguity as well as occlusion. Therefore, other marker-less systems utilize multiple cameras to capture 3-D human motion. Through such systems, the 3-D information of the observed human subject is captured from different directional views, thereby providing better results of recovering human motion in 3-D. However, many cameras may require complicated setup with extra software and hardware to support the transfers of large video data from multiple cameras over a network. Thus, there are always tradeoffs between the flexibility of using a single camera and an ability to get 3-D information using multiple cameras.

It is possible to obtain useful information including depths via a single camera. A stereo camera achieves depth perception in a manner similar to human eyes by finding the correspondences between two images from the left and right cameras to estimate the disparity images. The disparity let us know how far from the interested points to the camera through perspective projection. However, finding the correspondences of pixels from the two images is not an easy task. If the global view of image is concerned, it consumes a lot of time to process the whole image. Meanwhile, if just local areas of an image are taken into account, the correspondence from one pixel to others cannot be exactly evaluated. A time of arrival (TOA) camera emits a beam of laser and receives the light reflection from the object surfaces to reveal depths. With complicated implement, TOA camera is expensive. The structured light-based camera like Kinect uses a structured-dot pattern to light an object. The deformation of dot-light let us know the distance from a point in 3-D to the camera. With current implementation, the structured light-based camera achieves a better quality of depths with a lower price than the old ones. The Kinect camera is limited for indoor environment and for measuring distance within near distance (less than 10 meters), yet with regards to a low-cost and easy-setup system this camera is an option for human computer interface applications. Besides, the infrared-based technology integrated in Kinect allows it to operate

SoICT '14, December 04 - 05 2014, Hanoi, Viet Nam
Copyright 2014 ACM 978-1-4503-2930-9/14/12…$15.00
http://dx.doi.org/10.1145/2676585.2676624

in both day and night time, thereby appropriate for most personalized human aware systems.

In this work, we concern the uses of Kinect camera for human detection and segmentation from an image since it is an important discipline in extracting information of human subjects. Note that the depth information acquired by a depth camera is given by a depth image where a 16-bit value of a pixel of a depth image reveals a distance in millimeters to a point in 3-D. Many approaches estimated the edge maps from a depth image to find the boundary of an object [1,2,3]. The binary silhouette used for the human detection was presented in [4,5]. Apart of the human body like faces [6] and Shirt-shapes [7] suggested the location and presence of a human subject, yet the human extraction was still limited with this approach. Other efforts designed a classifier to find a human or non-human subject wihthin a region of interest. In order to summarize the features of each region, Haar-like features [8], Histogram of Oriented Gradient (HOG) [9], Scale Invariant Feature Transform (SIFT) [10], Local Ternay Direction Pattern [11] etc. are mainly applied. Along with depth, colour features provided extra cues to track the human across the video frames when an RGB camera is attached along with Kinect [12,13]. The combination of depths and color resulted better human recognition but limited the computational time since the size of an RGB image is three times larger than that of a depth image.

Although the aforementioned algorithms helped us to localize the human subject in a depth image, additional human segmentation is necessary to extract the body silhouettes. Background removal performed before detection aims at precisely determining the boundary of monitored subjects and the task of detection is mitigated by avoiding the ambiguity of interested regions from the background. Background removal has been developed for decades targeting to the specific domain of an RGB and a gray image. A static background can be averaged over a sequence of frames and subtracted by the current video frame to reveal the foregrounds. Lately, each pixel of the background image is modeled by Gaussian mixture distribution to allow the dynamic changes of captured environments [14, 15]. These approaches usually face foreground missing when tracked subjects stand still for a long duration of time. Some other works using both color and depth for background estimation were found in [16], but they were complicated and slow for real-time applications.

In this work, we propose a pipeline approach for the human extraction and detection of multiple subjects from complicated backgrounds. Different regions appeared in a depth image are isolated using the depth similarity of neighbor pixels. Such regions are merged into the bigger presenting objects that are later recognized as human though motion characteristics. Human subjects are tracked from frame-by-frame basis to avoid occlusion of multiple subjects. With the integration of effective background segmentation, the proposed approach is appropriate for the task of extracting and detecting human subjects from a depth image. Furthermore, the algorithm is fast for real-time applications.

The rest of the paper is organized as follows. Section 2 presents the methodology of the proposed approach. Section 3 shows our experimental results. We finalize the paper with discussions in Section 4.

2. METHODOLOGY

The overall system illustrated in Fig. 1 consists of the two certain components. The object segmentation component aims at distinguishing regions in a depth image. Each isolated region is correspondent to an object. Meanwhile, the foreground detection locates moving areas from a sequence of frames. The combination of the two allows us to separate moving objects from a depth image that is considered a human subject in our proposed method. Finally, extracted users are assigned identified numbers to track their depth silhouettes from frame-by-frame basis.

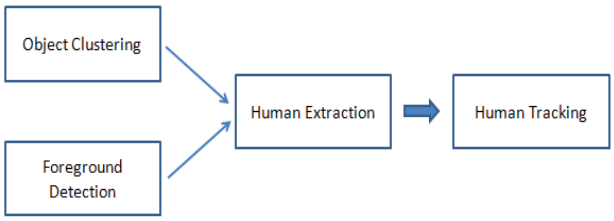


Figure 1. Step-by-step processing of our proposed approach.

2.1 Object Segmentation

The task for segment objects from image is challenging due to the presence of complicated backgrounds with furniture, overlapping, and occlusion as well as movements of multiple subjects. Fortunately, with available information of depth, each object can be presented by a cloud of points in 3-D. Usually, the human subjects stand on a floor. Therefore, after a floor is removed from a depth image, different objects in depth images are correspondent to isolated regions. These regions are found by the clustering algorithm which consists of the region growing algorithm to form connected regions namely superpixels [17] and the graph clustering to merge superpixels to reconstruct objects. The whole procedure to segment objects includes the two steps that are described as follows.

2.1.1 Floor removal

In this step, we deal with the problem of how to remove a floor from a depth image. Usually, a floor belongs to a lower part of a depth image, therefore, it is unnecessary to estimate the floor plane equation from the whole image. A low part of a depth image is extracted and its pixels are transformed into a set of 3-D points in a real world coordinate system (Fig. 2) as

$$X = \frac{(u - u_0)Z}{f} Y = \frac{(v - v_0)Z}{f} \quad (1)$$

where u and v are the row and column index of a pixel, u_0, v_0, and f are the parameters configured by a depth camera.

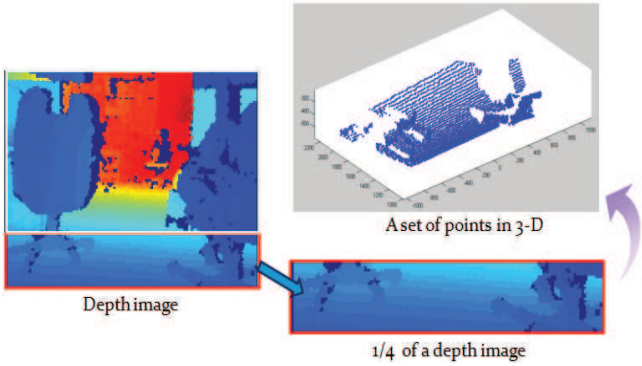


Figure 2. Extract 3-D points of the floor plane from a depth image.

Later some 3-D candidates of the floor are found to compute an equation of the plane. In order to find the candidates of the floor, a set of 3-D points will be cut by paralleled planes Z_k with an equation $z=a$ as in Fig. 3 in which a is a constant value determining the location of a plane along the z-axis. Continuously, the 3-D points are projected into the closest plane to find the points belonging to the floor in each plane Z_k.

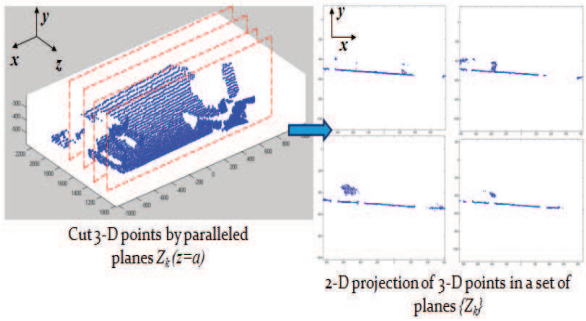


Figure 3. Project 3-D points into different *Z*-planes.

In each plane, the lower convex hull of 3-D points projected into Z_k is detected by the Graham Scan algorithm. Obviously, a set of points close to the lower convex hull are concerned the candidates belonging to a floor. Let a set of detected points be $X_i = \{x_i, y_i, z_i, i = 1, \dots, n\}$ that is used to formulate the floor plane equation $Y = aX + bZ + c$. The coefficients a, b, and c are found by the mean square estimation to minimize $\sum_i (ax_i + bz_i + c - y_i)^2$,

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum x_i^2 & \sum x_i z_i & \sum x_i \\ \sum x_i z_i & \sum z_i^2 & \sum z_i \\ \sum x_i & \sum z_i & n \end{pmatrix}^{-1} \begin{pmatrix} \sum x_i y_i \\ \sum y_i z_i \\ \sum y_i \end{pmatrix}. \quad (2)$$

After the plane equation is formulated, a point (x, y, z) with its distance $d = |ax + bz + c - y|/\sqrt{a^2 + b^2 + 1}$ to a plane less than a threshold is considered a floor pixel and eliminated as shown in Fig. 4. Note that, the floor removal is only useful if the number of candidate floor pixels is larger than a pre-determined value. Unless, the plane detected from an image is not big enough to be filtered and the floor removal function is disable.

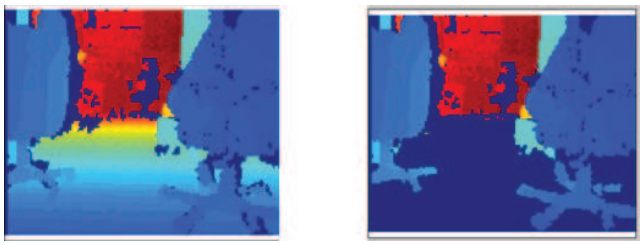


Figure 4. Floor is removed from a depth image.

2.1.2 Region growing and object clustering

Actually, the points in 3-D can be clustered by K-means to form objects but it is slow to handle a larger number of pixels in a depth image. Therefore, an extra step to reduce the number of inputs to the clustering algorithm is necessary. The region growing is utilized in our implementation to segment a depth image into non-overlapping superixels. The region is grown from a point x_i to its adjacent point x_{i+1} if the depth value of x_{i+1} is approximately equal to the depth value of x_i (the depth differences between them is less than a certain threshold). Just a single scan is sufficient for the whole image. However, due to occlusion, a single object might be separated into several superpixels. Therefore, a clustering is necessary to merge superixels to create an object.

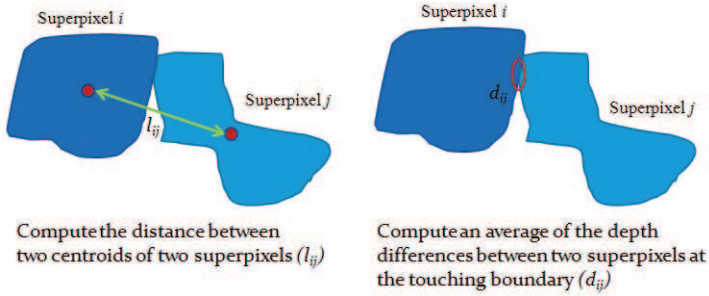


Figure 5. Two certain factors affect the decisions of combining superpixels.

Considering criteria to combine superpixels of the same object, we determine the centroids of regions. After that, the distance between the centroids l_{ij} and the average of the depth differences d_{ij} between superpixels i and j at the touching boundary are estimated (Fig. 5). Combining these two measurements, we evaluate the similarity between each pair of superpixels as

$$s(i,j) = \left(\frac{1}{1 + e^{-\alpha(d_{ij} - d_0)}}\right)\left(\frac{1}{1 + e^{-\beta(l_{ij} - l_0)}}\right) \quad (3)$$

where d_0 and l_0 are the two thresholds on the centroid distance and boundary depth difference to determine when the two superpixels should be merged together and α, β are the growth rates of the logistic function. Usually, there are about tens of superpixels in a depth image, hence the clustering task is mitigated with the remarkable reduction of the clustering inputs. We apply the spectral clustering via its advantage over K-means to handle non-convex regions. With n superpixels, we generate an affinity $n \times n$ matrix $\boldsymbol{A}$ with $a_{ij} = \begin{cases} s(i,j) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$. Correspondingly, a degree matrix $\boldsymbol{D}$ is constructed from $\boldsymbol{A}$ as a diagonal matrix with each element $d_{ii} = \sum_j A_{ij}$ and the normalized affinity matrix is $\boldsymbol{L} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$. In conventional spectral clustering, the k-largest eigenvectors $x_1, x_2, \dots, x_k$ of $\boldsymbol{L}$ is found and stacked in a column to form a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times k}$. Each row of $\boldsymbol{X}$ is normalized to have a unit length to present the mapping of the correspondent superpixel in an embedded space $\mathbb{R}^k$. Finally, the hierarchy clustering is performed to group close points within an embedded space: Two close points with their Euclidean distance not over a predefined threshold are merged into a group. The centroids of groups are recalculated to generate a new set of points. The produce is repeated until there are no new groups made. In our work, to save the computational time of eigenvector decomposition, we apply the power iteration procedure to estimate the largest eigenvector of $\boldsymbol{W} = \boldsymbol{D}^{-1}\boldsymbol{A}$ as described in Table 1.

Actually, the final convergence of the eigenvector found by the power iteration leads to a constant vector with respect to the eigenvalue $\boldsymbol{\lambda_1} = \boldsymbol{1}$. Alternatively, the algorithm described in Table 1 aims at finding the local convergence of the largest eigenvector to make sure the elements of the same cluster are

close yet the different clusters still not approach to the same value at the final convergence.

Table 1. Algorithm to estimate the largest eigenvector of the normalized affinity matrix.

Input:

$n \leftarrow$ the number of superpixels

A $\leftarrow$ the affinity matrix evaluates the similarity between superpixels

Algorithm:

1. Construct $n \times n$ degree matrix $\boldsymbol{D}$ and the normalized affinity matrix $\boldsymbol{W} = \boldsymbol{D}^{-1}\boldsymbol{A}$
2. Initialize eigenvector v_0 with the size size $n \times 1$ randomly
3. Update v_{t+1} as $v_{t+1} \leftarrow \boldsymbol{W}v_t$
4. Normalize $v_{t+1} \leftarrow v_{t+1}/\|v_{t+1}\|$
5. Compute $acceleration = |\sigma_{t+1} - \sigma_t|$ where $\sigma_{t+1} = |v_{t+1} - v_t|$
6. If *acceleration* is less than a threshold, the algorithm is stopped, otherwise go back to step 3

2.2 Foreground Detection

The purpose on this stage is to estimate a foreground image of pixels changing in a given sequence of depth images. The equivalent problem is how to dynamically obtain the background images from a frame sequence. Corresponding foreground pixels are the differences between a current frame with a background image, which satisfy the condition $|frame_i - background_i| > \varepsilon$, where ε is a predefined threshold.

Previously, many approaches [15] have been proposed for the task of foreground detection or background estimation. For instance, background is defined as a previous frame, as the maximum depth image among *n*-previous frames, as a running average, as a joint probability density function (PDF) estimated by nonparametric density estimation, as the smallest eigenvector of covariance image estimated by principal component analysis, and as a mixture of Gaussian distribution. The details of these techniques are illustrated in Table 2.

In this work, we estimate a background using a mixture of Gaussian clusters to discover a foreground image for both forward as well as backward movements of a human subject and to reduce the size buffer to store the history depth images.

2.3 Combination of Object Segmentation and Foreground Detection for Human Extraction

We consider a segmented object containing changing pixels (moving) is a human subject. The algorithm performed in Section 2.1 segments a depth image into non-overlapping regions correspondent to objects. At the same time, we estimate the foreground images consisting of changing pixels using a sequence of history frames (in Section 2.2). Finally, in order to completely remove a background, detected regions consisting of many foreground pixels are retained, the others are eliminated.

2.4 Tracking Multiple Subjects

This part answers a question of how to track a moving subject from frame to frame basis. According to our provided techniques, we match the segmented regions from one frame to the next frame as shown in Fig. 6. Hence, two overlapping regions in two consecutive frames receive the same identified number. By tracking the identified number of different regions, we can separate different subjects and track their identified number on a frame-by-frame basis. In order to match each region in the current frame with every region in the previous frame, we compute the intersection of each pair of regions of the two depth frames.

In general, each pixel is assigned a linear index locating the row and column of this pixel in a depth image. Therefore, a region is stored by a linked-list of pixel indices. Finding intersection set of the two regions A and B requires $|A|log|A| + |B|log|B|$ computations, where $|A|$is the number of pixels within the region A and $|B|$for the region B. Actually, this computational time is cost for sorting the indices of pixels of the two sets A and B. We calculate a ratio $\frac{|A \cap B|}{\max(|A|,|B|)}$ to measure the matching of the two areas. If the matching ration is over a predefined threshold, we consider that the two areas are of the same subject.

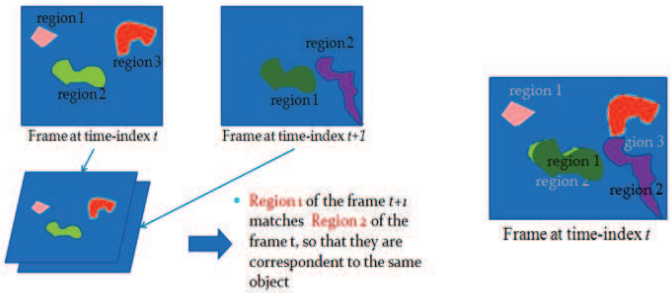


Figure 6. Tracking regions from frame to frame.

In case of tracking a human subject, a region detected as changing is assigned a user identified number and is tracked. By applying our proposed method, even when a human subject stands still, the method is effective to detect which region is corresponding to the subject and to separate this region from the background.

There are the problems that when the human touches another object from the background, their depth silhouettes are merged together as the results of region growing. To avoid so, we stored a link-list of background objects obtained from previous frames. When a human region is detected, this region will be matched with background regions to eliminate the redundant of the intersection of the two objects.

3. EXPERIMENTAL RESULTS

3.1 Evaluation of Foreground Detection Algorithm

In this section, we conduct a series of experiments to find a suitable foreground algorithm that is capable of detecting moving pixels from a sequence of depth images. Alterative algorithms implemented in our experiments estimate a foreground (a changing image) by comparing the differences between the current frame with the background defined by a previous frame, a maximum depth image, a running average image, an image with nonparametric density distribution (NDD), an eigen-background, and an image with mixture Gaussian distribution. Qualitative evaluation of different algorithms on a sequence of depth images is shown in Fig. 7. Obviously, using a previous frame, a running average image, an image with nonparametric density distribution, and an eigen-background results a foreground including static regions, that is unexpected in our approach since we want to detect only moving pixels from an image. An amount of moving pixels later play a critical role to determine which object will be assigned as a human subject and tracked along the frame sequence.

The method using a maximum depth image among history depth image stored in a buffer can extract foregrounds free from the presence of static regions. However, due to the limited size of the buffer, this approach may skip the backward movements. Finally, as described in Fig. 8 an approach of estimating a background image using the mixture Gaussian distribution is an appropriate solution for our implementation since it not only takes above advantages of using a maximum depth image but also can detect both forward and backward movements of the subjects.

An average computational speed of differences algorithm is given in Table 3. As we can see, except the algorithms computing a background using PCA and nonparametric density estimation, all of them are suitable for realtime implementations.

Table 3. An average computational speed(fps) of different algorithms defining a background as

Previous frame	Maximum depth image	Running average	NDD	Eigen-background	Mixture of Gaussian
810	*320*	*650*	*10*	*12*	*200*

3.2 Evaluation of Human Extraction from a Sequence of Depth Images

We evaluated our proposed algorithm to extract human subjects from a sequence of frames recorded by the camera Kinect. The two different scenes including a room and corridor have been tested in our experiments. We recorded 800~1500 frames for each depth sequence. The human extraction results are analyzed through subjective evaluation. That means we computed the extraction accuracy by counting how many frames were correctly processed over the total number of depth images within a sequence. The overall accuracy results are given in Table 4 and the human extraction are shown in Fig. 9 respectively. Here, the maximum number of persons are allowed to enter a sense is two. Examples of failed detection are given in Fig. 10. Obviously, the proposed algorithm is able to detect and extract human subjects within complex background. However, touching between subjects as well as a subject with big regions of the background causes the region growing failed to determine an exact boundary of the human subject.

In overall, the whole system is implemented on a laptop using Intel Core i5 2.5 GHz CPU with multiple cores where a single thread on a single core is utilized for the computations. In average, the algorithm is able to process 60fps with a depth image of QGVA 320× 240 resolution, that is fast for realtime applications.

Table 4. Accuracy performance of our proposed approaches.

Scenes	A Single Subject	Two Subjects
Room	92.3%	98.5%
Corridor	97.3%	98.1%

4. CONCLUSION AND DISCUSSION

In this paper, we have proposed an efficient and promising method to remove static regions from depth images to extract and track multiple users from frame to frame. The proposed approach poses some advantages: It dynamically estimates the background image, so that we do not need to initialize a background image for the first time; The algorithm is able to track multiple users and separate them apart using different identified numbers; The method is fast with the processing speed of around 60fps that is suitable for real time applications; Besides, our proposed algorithm can be operated within various environments.

Some drawbacks of our proposed algorithm can be improved in further works. Firstly, the identified number of human subjects are lost if they occlude each other. We suggest the use of tracking algorithm with Kalman Filter to locate a human subject for duration of time even the subject disappears from a scene. Secondly, when an individual touch others, their depth silhouettes are merged into one. In order to address such a problem, matching the current frame with the previous ones to separate a complex region of human subjects into a part is an efficient solution. When an individual touches objects from complicated background, a sequence of frame is used to evaluate whether or not a set of pixels belong to background objects. Besides, unexpected objects such as fans, moving chairs, etc. can be rejected by applying shape classification algorithm. Finally, we realized that moving a camera leads to difficulties of extracting a foreground image. Therefore, we plan to integrate the moving information of the camera to our human detection and extraction algorithm.

5. ACKNOWLEDGMENTS

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2013.11

6. REFERENCES

[1] Schafer, H., Lenzen, F., Garbe, C.: Depth and Intensity Based Edge Detection in Time-of-flight Image. In: 3DV-Conference, (2013).

[2] Lejeune, A., Pierad, S., Droogenbroeck, M. V., and Verly, A.: A New Jump Edge Detection Method for 3D Cameras. In: International Conference on 3D Imaging (ICA3D), (2011).

[3] Xia, L., Chen, C.-C., Aggarwal J.K. Human Detection Using Depth Information by Kinect. Computer Vision and Pattern Recognition Workshops, (2011).

[4] Xu, F., Fujimuar, K.: Human Detection Using Depth and Gray Images. In: Advanced Video and Signal Based Surveillance, (2003).

[5] Zhao, L., Thorpe, C.: Stereo and Neural Network-based Pedestrian Detection. IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 3, pp. 148--154 (2000).

[6] Munoz, R., Aguiire, E., Garcia, M.: People Detection and Tracking Using Stereo Vision and Color. Image and Vision Computing, vol. 25, no. 6, pp. 995-1007 (2007).

[7] Southwell, B. J., Fang, G.: Human Object Recognition Using Colour and Depth Information from an RGB-D Kinect Sensor. International Journal of Advanced Robotic Systems, vol. 10, no. 171, (2013).

[8] Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: IEEE Computer Vision and Pattern Recognition, (2001).

[9] Dalal, N., Trigs, B.: Historgrams of Oriened Gradients for Human Detection: In: IEEE Computer Vision and Pattern Recognition, (2005).

[10] Lowe, D.: Object Recognition from Local Scale-invariant Features, In: IEEE International Conference on Computer Vision, (1999).

[11] Shen, Y., Wang P., Ma, S., Liu, W.: A Novel Human Detection Approach Based on Depth Map via Kinect. IEEE Conference on Computer Vision and Pattern Recognition Workshops, (2013).

[12] Zhao, Y., Liu, Z., Yang, L., Cheng, H.: Combining RGB and Depth Map Features for Human Activity Recognition. Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), Asia-Pacific, (2012).

[13] Salas, J.,Tomasi, C.: People Detection Using Color and Depth Images. Lecture Notes in Computer Science, (2011).

[14] Langmann, B., Ghobadi, S., Hartmann, K., Hoffeld, O., Multi-modal Background Subtraction Using Gaussian Mixture Models. In: IAPRS, (2010).

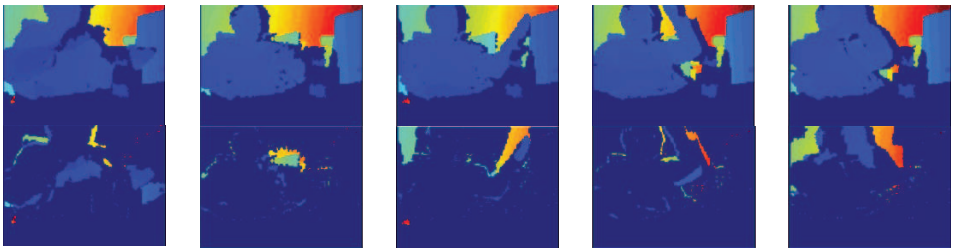
[15] Brutzer, S., H¨oferlin, B., Heidemann, G.: Evaluation of Background Subtraction Techniques for Video Surveillance, IEEE Conference on Computer Vision and Pattern Recognition Workshops, (2011).

[16] Crabb, R., Tracey, C., Puranik, A., Davis, J.: Real-time Foreground Segmentation via Range and Color Imaging. In: Computer Vision and Pattern Recognition Workshops, (2008).

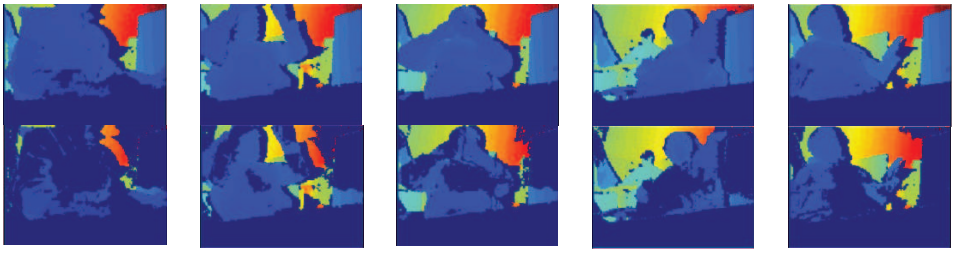
[17] Toyoda, T., Hasegawa, O.: Random Field Model for Integration of Local Information and Global Information. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 30, no. 8, pp.1483--1489. (2008).

Table 2. Comparisons between different background detection algorithms.

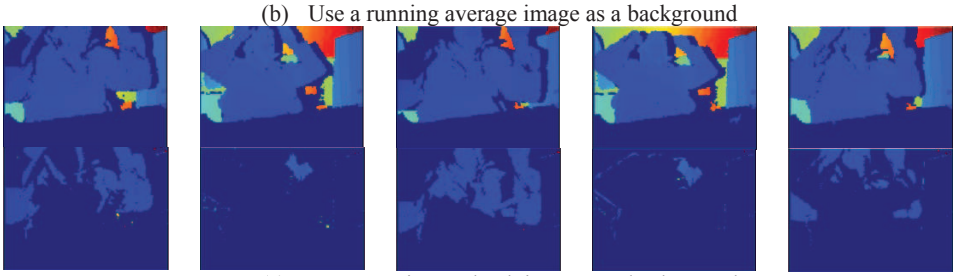
Algorithms	Method	Drawbacks
Background as a previous frame	$\lvert frame_t - frame_{t-1} \rvert > threshold$	Detected foreground contains both moving regions and some parts of a static background
Background as the maximum of *n*-previous frames	$background = \max\{I_t, I_{t-1}, \dots, I_{t-h}\}$ where I_t is the depth image at the time index t	The algorithm cannot detect backward movements
Background as an running average	$B_{t+1} = \alpha I_t + (1-\alpha) B_t$ where α is learning rate	Detected foreground contains a part of static background
Background PDF estimated by nonparametric density estimation	The PDF of each background pixel x_t is estimated from *N*-pixels $S=\{x_i\}_{i=1\dots N}$ by $\Pr(x_t) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(x_t - x_i)$ where K_σ is a kernel function. A foreground is detected by $\Pr(x_t) < threshold$	Slow processing and noisy
Background estimated by principal component analysis	In this algorithm, *n*-previous frames are re-arranged as the columns of a matrix ***A***. Later, the covariance matrix $\mathbf{C} = \boldsymbol{AA}^T$ is computed. From ***C***, the diagonal matrix of its eigenvalues, ***L*** and the eigenvector matrix, ***Φ***, are estimated accordingly. Only the first ***M*** eigenvectors (the eigenbackgrounds) are retained. Once a new image I_t is available, it is first projected in the ***M*** eigenvectors and reconstructed as I_t'. The difference I_t-I_t' between I_t and I_t' reveals the foreground image since the sub-space eigenvector represents only the static parts of the scene.	Slow processing and detected foreground contains a static background
Background estimation with a mixture of Gaussian clusters	The background PDF is presented by the mixture of K Gaussian (μ_i, σ_i, ω_i), where ω_i is the weight of each Gaussian cluster and the mean μ is updated by running average $\mu_{t+1} = \alpha F_t + (1-\alpha)\mu_t$. The cluster with the highest value of weight ω_i is chosen as a background PDF. The weight equation ω_i is updated by $\omega_{i,k}^t = (1-\alpha)\omega_{i,k}^{t-1} + \alpha I_{i,k}^t$. In order to avoid the presence of static background, the Gaussian cluster-*i* is chosen as a background which contains the largest values of $\omega_i^t \mu_i^t$.	Able to detect both forward and backward movements



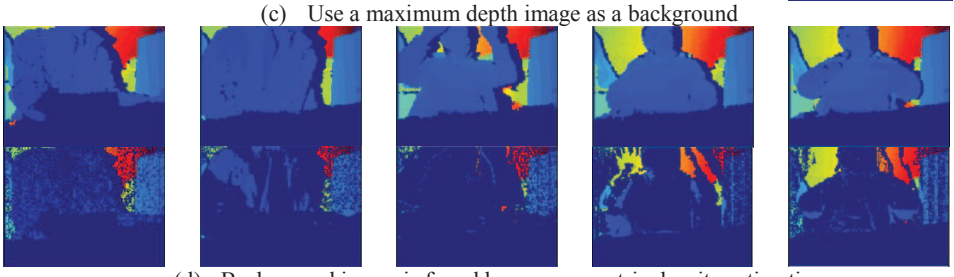
(a) Use a previous frame as a background



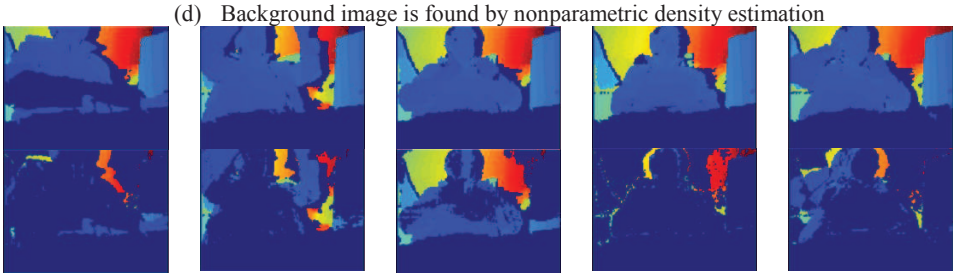
(b) Use a running average image as a background



(c) Use a maximum depth image as a background

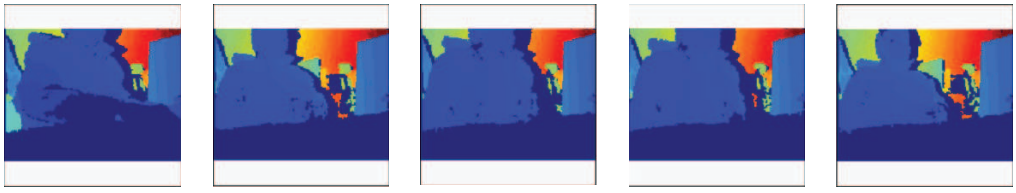


(d) Background image is found by nonparametric density estimation



(e) Background image is found by PCA

Figure 7. Different algorithm used to extract foreground images in which the top row shows the original depth images and the bottom shows the segmented foreground.



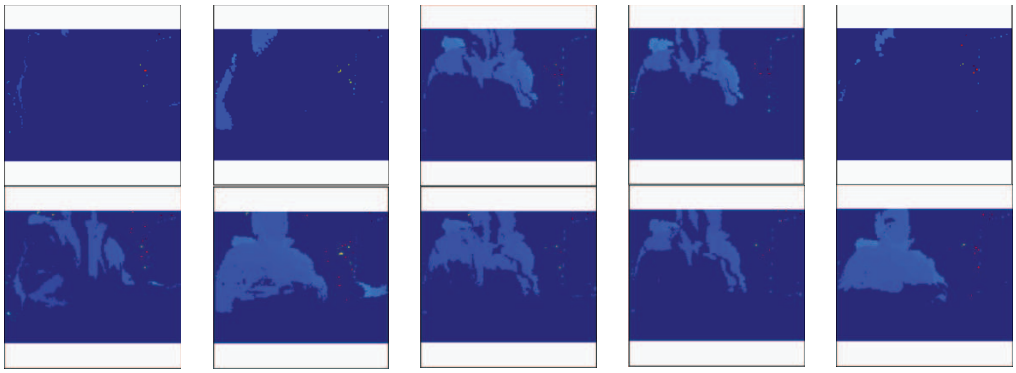
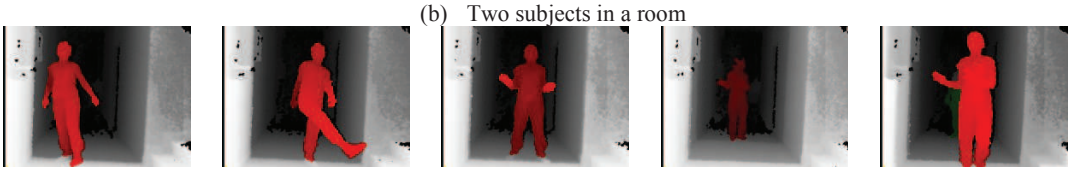


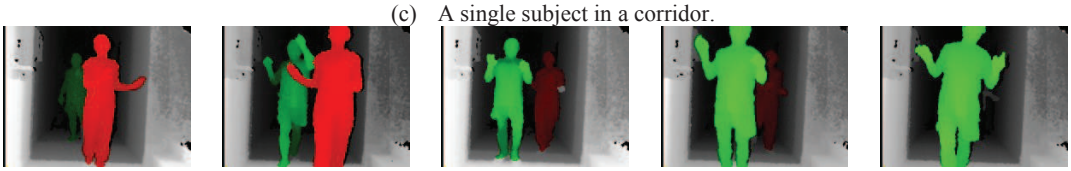
Figure. 8. Comparisons between a background estimated as a maximum depth image and as mixture of Gaussian distribution. The first row shows the original images, the second shows the foreground extracted by a maximum image and the last shows the foreground image extracted by Gaussian mixture distribution.

(a) Single object in a room

(b) Two subjects in a room



(c) A single subject in a corridor.



(d) Two subjects in a corridor

Figure 9. Tracking multiple subjects with various environments.

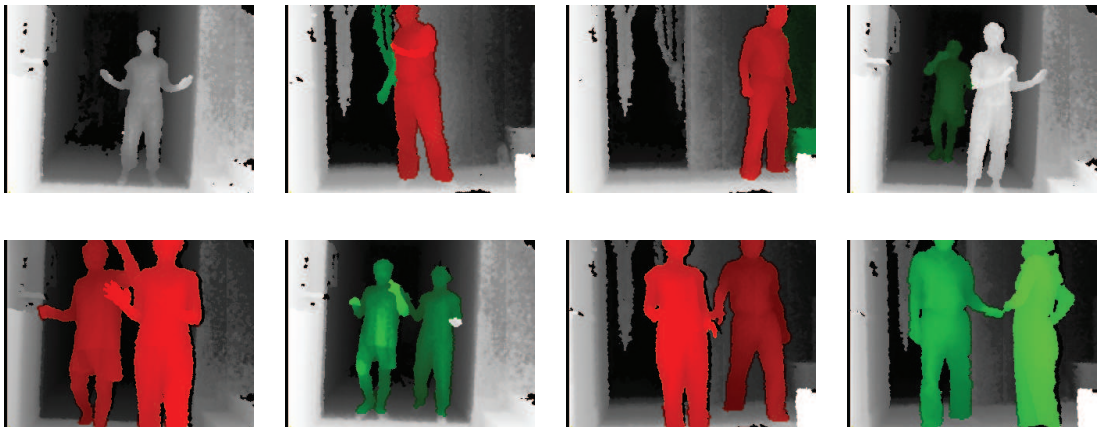


Figure 10. Failed detection examples.