FIRST-ORDER HUMAN DETECTION AND EXTRACTION FROM TWO PRIMSENSE VIDEOS CAPTURED IN OFFLINE MODE

Nguyen Van Khoa[(1)], Ly Minh Hoang[(1)], Dang Thanh Tin[(1)]
(1) Ho Chi Minh City University of Technology, Viet Nam.
Email: vankhoa96pfiev@gmail.com

ABSTRACT

This paper presents a method, using tools like OpenCV, OpenNI and OpenGL for first-order human detection and extraction. First-order human is a human that is not occulted by anything from that point of view. The videos were captured by camera with infrared sensor and color sensor. First-order humans were captured and extracted by various devices, then were regrouped and displayed in one common frame. In this study, we present an althorithm for human detection using depth information taken by the PrimSense camera. We propose a extraction scheme to segment the human from his/her surroundings and extract the whole contours of the figure based on our detection point. Moreover this research aims at creating a foundation for building a program capable of linking many people from different places and establishing an online environment to for them to communicate.

KEYWORDS: *human detection, first-order human extraction, PrimeSense.*

1. INTRODUCTION

Nowadays, image processing is not a new scientific topic anymore wherein human detection is one of a major trends. Detecting human and objects in images or videos is a challenging problem due to variations in pose, clothing, lighting conditions and complexity of the background. Various methods are used in detecting and identifying human in front of the camera. These methods are then manipulated in many other domains such as robot – controlling [1], object detection [2][3].

Image processing is mostly done with images taken by visible-light cameras. The existing methods imitate the detection process that humans use. They use features based on gradient, such as histogram of oriented gradients (HOG) and Support Vector machine (SVM) [3][4], or they extract points of interest in the image, such as scale-invariant feature transform (SIFT). They also use the mathematical transforms like Wavelet transform and Fourier transform [5] to detect and extract the objects and the persons from background. Extracting persons from background is an image processing process that requires so much experiences and knowledge. This process supports human identification, and make this field widely applied in numerous topics in real life. Researchers have created a camera with infrared sensor and color sensor that makes it easy to identify objects in front of the camera and especially human, by taking advantage of the characteristics of sensors attached to the camera. These are the characteristics of depth information that make it easier to detect and classify persons by processing on each pixel and using tools like OpenCV, OpenGL and OpenNI which are built to support these types of jobs.

The problem of identifying human has received much interest and can be applied to many practical applications. One of them is to create an online environment where anyone from anywhere can talk to each other. There are already a number of applications that have been created to do the job such as Skype. Most of these applications are only for casual use such as chatting with friends or family. They are not used in professional environment because they are not able to remove the surrounding objects and people who are in front of the camera. The purpose of this study is to solve that problem by creating a more professional environment that makes communication easier by using the tools and features mentioned above.

2. OVERVIEW OF `METHOD

This section provides the overview of our method, which is summarized in Figure 1. Implementation details are presented in Sections 3.

Our method is based on the depth information provided by PrimeSense. First of all, the PrimeSense device captures video directly from the user. The information obtained from PrimeSense are color information and depth information recorded by sensors attached to the camera. This information is independently recorded so we have to synchronize the data obtained from the camera. After that, we will reduce the noise and equalize the histogram for each frame in the video. The human extraction process for each frame is done using

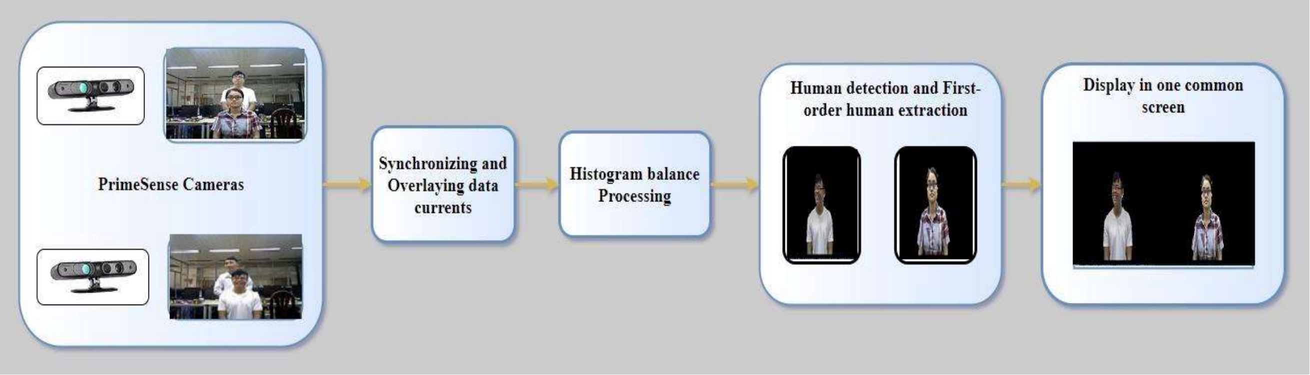

PrimeSense Cameras
Synchronizing and Overlaying data currents
Histogram balance Processing
Human detection and First-order human extraction
Display in one common screen

Fig 1 Overview of the first-order human detection method.

tools like OpenGL, OpenNI. Based on the depth information, we know the distance of each user to the camera. We use this information to extract the first-order human for each frame.

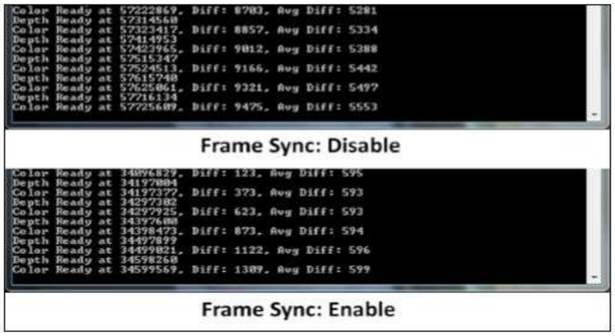
3. DETAILED IMPLEMENTATION

3.1 Recording device

The data is captured directly by PrimeSense camera. Then the information is about the depth and color. The depth-information current carries information about the distance between the human and the camera. The color-information current carries information about the color of the corresponding point in the depth space.

3.2 Synchronizing and Overlaying data currents

A device, by default, captures and sends frames from each sensor independently. This means there is no guarantee that both sensors capture a snapshot of the environment at the same time. But using frame syncing that is available for image and depth streams, we can at least decrease this difference in capture time to the lowest possible value. Figure 2 shows that there is a huge difference (almost ten times less) when depth and color frame syncing is enabled.



```
Color Ready at 57222869, Diff: 8703, Avg Diff: 5281
Depth Ready at 57314560
Color Ready at 57323417, Diff: 8857, Avg Diff: 5334
Depth Ready at 57414953
Color Ready at 57423965, Diff: 9012, Avg Diff: 5388
Depth Ready at 57515347
Color Ready at 57524513, Diff: 9166, Avg Diff: 5442
Depth Ready at 57615740
Color Ready at 57625061, Diff: 9321, Avg Diff: 5497
Depth Ready at 57716134
Color Ready at 57725609, Diff: 9475, Avg Diff: 5553
```

Frame Sync: Disable

```
Color Ready at 34096829, Diff: 123, Avg Diff: 595
Depth Ready at 34197004
Color Ready at 34197377, Diff: 373, Avg Diff: 593
Depth Ready at 34297382
Color Ready at 34297925, Diff: 623, Avg Diff: 593
Depth Ready at 34397600
Color Ready at 34398473, Diff: 873, Avg Diff: 594
Depth Ready at 34497899
Color Ready at 34499021, Diff: 1122, Avg Diff: 596
Depth Ready at 34598260
Color Ready at 34599569, Diff: 1309, Avg Diff: 599
```

Frame Sync: Enable

Fig 2 Difference between time capture of depth and color frame.

The depth and color streams are from two different sensors, and because of the difference in their positions in the device, they see objects from two different angles. This makes it hard to figure out which two pixels in these two streams are related to each other in the physical world. Fortunately, this problem is solved by OpenNI using one of its built-in methods. Using this feature, we can expect each pixel of depth to be in the same position as its color counterpart. This feature is very useful for different types of projects, including generating a color point cloud. We will overlay the depth data over the color data using this OpenNI's feature. Figure 3 shows that the left-hand side picture has both depth and image data exactly in the same place, unlike the right-hand side picture that has significant displacement.

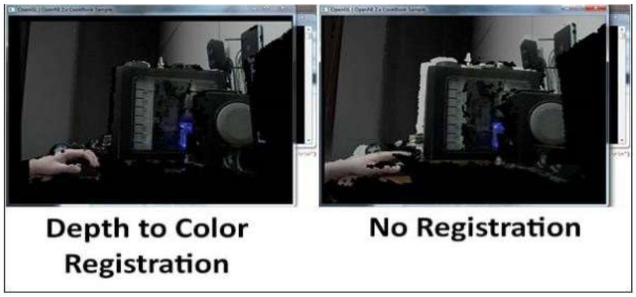


Fig 3 Difference between Depth to Color Registration image and No Registration image.

3.3 Histogram balance preprocessing for video

We have an image with different colors of grayscale color space. That is, we have different colors between white and black. Sometimes we see an image that has a lot of these colors are similar to each other and there are a lot of other colors that are rarely used. Using histogram equalization, we can use all of our color space by dynamically changing the contrast of the image to show more important data with much more detail [7]. Figure 4 is a picture before the histogram equalizing process.

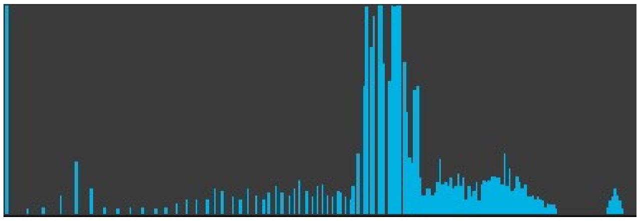


Fig 4 Histogram before Equalizing.

We can clearly see, the majority of the colors used are in a specific range and most of the color space is never really used. Using histogram equalization, we can change that somewhat, to use all parts of the color space. Figure 5 is the same histogram after correction. Histogram equalization is a good way to increase the contrast of important parts of an image that use very limited parts of a color space. But its main advantage reveals itself when we are going to fit an image with a bigger color palette into a smaller one, because in addition to showing important parts of an image more clearly, it will add more detail to the image by converting only the useful parts of color palette into a new color space. Figure 6 shows the change in grayscale line and also how the image changed.

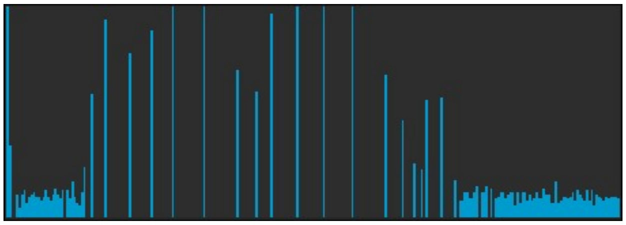


Fig 5 Histogram after equalization.

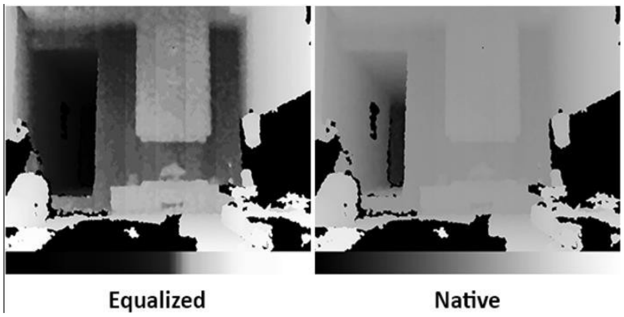


Fig 6 Changes in grayscale of Equalized and native image.

3.4 Human detection and human extraction.

We will use NiTE for detecting persons, which is a middleware based on OpenNI and more focused on natural interactions. We will use the nite::UserTracker class to get the list of all active users in the scene and get their locations and sizes in OpenNI's depth stream, along with the center of visual mass of each user. This class is responsible for getting any information relative to users and their bodies. nite::UserTracker works like openni::VideoStream and has similar ways of working. However, it is actually not a sensor or a direct output of any physical unit. nite::UserTracker gives us what NiTE recognized from a depth stream of OpenNI. We use nite::UserMap to get the user ID for each pixel in the depth stream. By using this information, we can detect the pixels belonging to one user. Each user in the scene has a distinct ID address. If a pixel belongs to nobody, that pixel's ID address will be 0. Figure 7 is the algorithm used to detect human using NiTE.

3.5 First–order human detection.

By using depth information, we can calculate the distance between the camera and the human. NiTE has a tool to determine human's Center of Mass (COM), from which we can estimate the distance between COM and the camera. Then the first-order human identification problem simply becomes the problem of identifying the human who has the smallest distance to the camera. Figure 8 is the algorithm for identifying the first-order human, we assume that the first-order human is within the radius of 10 meters from the camera, because the distance for which the camera works well is between 2 meters and 5 meters.

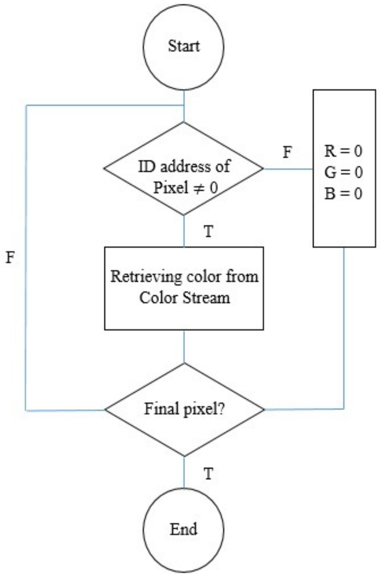


Fig 7 Algorithm to detect human using NiTE.

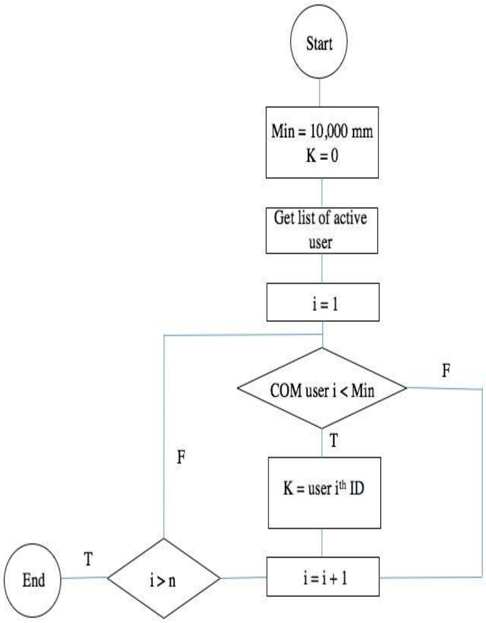


Fig 8 Algorithm to identify first-order human.

When the first-order human is detected, we can manipulate the human detection algorithm to extract the human image from the background.

3.6 Display on the screen.

The first step is calculating the resize factor of the resizing process by comparing the width and height of both the newly generated frame and the texture buffer. After that, we can see where we used a size for this buffer. Then, from this resize factor value, we can calculate the X and Y padding required for the frame to fit into our texture buffer when they don't have the same ratio.

$$resizefactor = \min\left(\frac{texturebuffer's\ width}{frame's\ width}, \frac{texturebuffer's\ height}{frame's\ height}\right)$$

$$texture_x = \frac{texturebuffer's\ width - resizefactor * frame's\ width}{2}$$

$$texture_y = \frac{texturebuffer's\ height - resizefactor * frame's\ height}{2}$$

Figure 9 shows what these variables mean. Assume we have a texture buffer with a resolution of 1280 x 1024 and a frame from the color sensor with a resolution of 640 x 480. We want to fit this frame into our texture buffer so that the user can see the entire image without cropping.

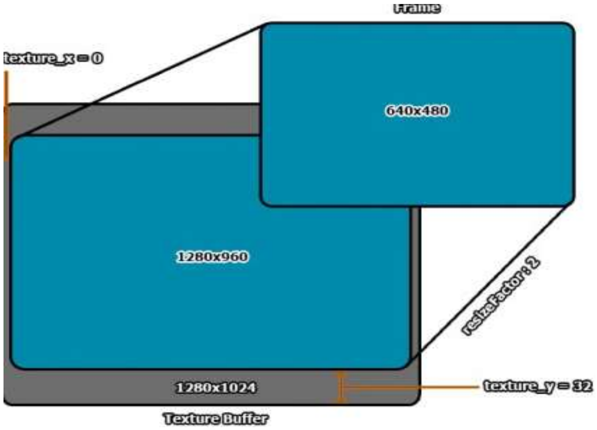


Fig 9 Size value calculation.

After doing so, we need to put each pixel into our texture buffer. For doing that, we need to loop through our texture buffer row by row and then pixel by pixel. This is done with two FOR loops, one for each row and the other for each pixel of the texture buffer. After the first FOR loop that loops through each row, we need to calculate the position of the first pixel of that row. Then, in the second FOR loop, we move forward and fill the other pixels of that row too. For each pixel, we use the first-order human detection algorithm. We need to calculate the position of the responsible pixel in the sensor's frame data, convert each pixel, and copy it to the texture buffer. From the sensor's frame data, which is a combine of Depth sensor's data and Color sensor's data, we can get the color of the first-order human.

4. EXPERIMENTAL RESULTS

We evaluate our method using a sequence of depth arrays taken by the PrimeSense in indoor environment. We took the sequence in our lab with at least two persons presented in the scene. There are also tables, chairs, shelves, computers, an overhead lamp and so on presented in the scene. The people have a variety of poses, and they interact with others or the surrounding objects.

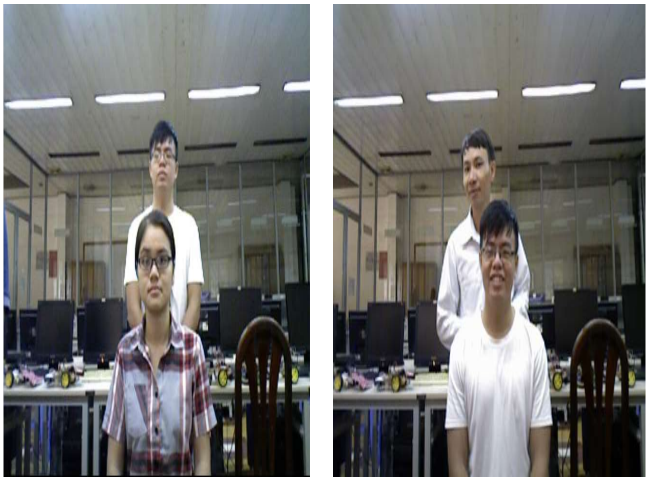


Fig 10 Input video captured from two PrimeScenes

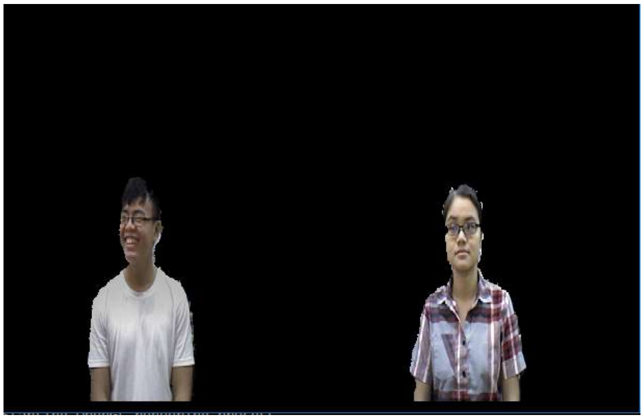


Fig 11 Extracted first-order humans and linking them in one common video.

5. CONCLUSIONS

In this paper, we propose a first-order human detection and extraction algorithm that uses the depth data and color data of each frame in video obtained from the PrimeSense camera. The experimental results show that our algorithm works well with the first-order human in all poses. The approach can be further applied in Real-time first-order human detection and extraction. The advantages of our method can be briefly described in the following. Firstly, the algorithm can be applied to multiple input videos. Moreover, there is no limit for the number of people present in each input videos. The limitation is that, there is a quick transition of color at the body contour of the first-order human because all pixels that don't belong to a human are then reset to zero value, which means that in the post-processing frame, these pixels turn into zero pixels (pixels with zero value) to make a black background. In the future, we plan to use some smoothing algorithms to smooth the human's body contour and create a more vivid background for the output frame to have a more useful software.

ACKNOWLEDGEMENT

Sponsored by Falculty of Electricals and Electronics Engineering, Ho Chi Minh University of Technology, Viet Nam National University, Ho Chi Minh City, Vietnam, this paper is completed thanks to many aids. We would like to express gratitude toward Assoc. Prof- PhD Đặng Thành Tín for sparing his occupied schedule to guide and encourage us. We as well wish to thank our classmates and friends for always metally and physically supporting us.

REFERENCES

[1] José-Juan Hernández-Lopez, Ana-Linnet Quintanilla-Olvera, José-Luis López-Ramírez, Francisco-Javier Rangel-Butanda, Mario-Alberto Ibarra-Manzano, Dora-Luz Almanza-Ojeda, "Detecting objects using color and depth segmentation with Kinect sensor," *Procedia Technology*, vol. 3, pp. 196-204, 2012.

[2] Navneet Dalal and Bill Triggs, "*Histograms of Oriented Gradients for Human Detection,*" in *2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2005.*

[3] Yunsheng Jiang and Jinwen Ma, "*Combination Features and Models for Human Detection,*" in 2015 IEEE Conf. on Computer Vision and Pattern Recognition, 2015, pp. 240-248.

[4] Qiang Zhu et al., "*Fast Human detection using a cascade of histograms of oriented gradients,*" in *2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2006.*

[5] M. Oren et al., "*Pedestrian detection using wavelet templates,*" in *Proc.* of the 1997 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 1997.

[6] Lu Xia et al., *"Human detection using depth information by Kinect,*" in 2011 *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2011.

[7] *"Image histogram,*" [Online]. Available: en.wikipedia.org/wiki/Image_histogram. [Accessed: May. 20, 2017].

[8] Soroush Falahati, 2013. *OpenNI Cookbook.* Packt Publishing, Birmingham, UK, 324 pages.