Interpret Human Gestures with a Time of Flight Camera Using Standard Image Processing Algorithms on a Distributed System

Bjoern Froemmer, Nils Roeder, and Elke Hergenroether

Hochschule Darmstadt, University of Applied Sciences,
Faculty of Computer Science
{bjoern.froemmer,nils.roeder,elke.hergenroether}@h-da.de

Abstract. The development of Human Computer Interfaces steadily moves away from peripheral devices like mouse and keyboard in certain areas, as is obvious when looking at the evolution of smart-phones, tablet-PCs and touch-enabled operating systems over the last few years. Nowadays we can even witness the transition from touch-based interfaces to touch-free interfaces. One common method to realize such interfaces is to incorporate new state-of-the art 3D cameras (often called "Time of Flight" cameras). The difficulty lies within the evaluation of the sensor-data, to achieve robust detection and tracking of people within the scene in real-time. We try to solve this task without using expensive knowledge-based approaches by employing standard image-processing algorithms because we wanted to keep the required manpower and development time, as well as costs, as low as possible.

Keywords: Natural User Interface, Human Computer Interface, Segmentation, 3D-Camera, Time-of-Flight, Region Growing, Edge Detection, Convexity Defects, Tracking, Gestural Interaction.

1 Introduction

The main research areas of this work have been as follows. What kind of image-processing methods can be used to segment and evaluate the sensor data of monocular 3D cameras in respect of tracking and evaluating human gestures? How fast (in terms of calculation time) can this be done? How reliable can we analyze and interpret human motions to achieve gestural interaction without using knowledge-based approaches?

To evaluate these questions we made use of a ToF[1] camera to record humans from the front (see figure 1). We then used the OpenCV Framework and some of it's standard image processing algorithms like edge detection, region growing and geometrical classification to segment and interpret the measured 3D data. We intentionally wanted to refrain from utilizing knowledge-based approaches like human-pose-databases in combination with classifiers that need to be trained (see [16] for example).

[1] ToF stands for Time-of-Flight and will be used as an abbreviation in this paper from here on.

C. Stephanidis (Ed.): Posters, Part I, HCII 2013, CCIS 373, pp. 317–321, 2013.

2 System and Algorithm

The implemented system consists of three main modules. These involve a camera-data server, a processing module and the client application. The camera server polls the data of the ToF camera and distributes it to the local network. The distributed data can then be accessed and processed by any number of so-called tracker-modules. This modular structure allows to disassemble complex tasks into partial solutions and distribute the computational load of the system onto multiple machines if needed. For example, it is conceivable that several trackers work cooperatively at the same or at completely different tasks within the same scene. The processing-module itself is divided into three distinct areas: segmentation, classification and tracking of objects.

Our goal is the segmentation of objects solely based on their depth information. The first step of the algorithm is to perform a pre-segmentation of the recorded depth-data to extract relevant areas for the subsequent main segmentation. This is achieved by the removal of static (unmoving) areas from the actual recorded depth-data. The next step is to remove depth-values with poor signal quality (values with low amplitudes). This ensures a solid basis for the subsequently described main-segmentation. After the aforementioned steps are done, only depth-values relevant to our goal remain and a region growing algorithm is utilized to clusterise spatially connected 3D points depending on their depth data. Based on the size of these clusters, areas of interest are identified which can then be processed further. Clusters that contain only few pixels, for example, will be eliminated using a threshold function. Depending on the application, it may be advantageous to select this threshold relative to the distance of an object. One of the possible reasons to do this is that the ratio of the areas between corresponding hands and torsos of people stays roughly the same, no matter how great the distance is between the user and the camera.

To realize hand-based interaction it is important to classify the identified objects, which remain after the segmentation, as certain body-parts. The goal here is to distinguish hands from other segmented body parts. For this task (the recognition of hands) we use geometrical classifiers based on convex hull polygons and so-called convexity defects as seen in figure 1 (image 9). The system later on basically tracks all segmented areas, but the intended human-computer-interaction should be made possible exclusively with the users hands. If a segmented area is identified as a hand, this classification will remain in the system as long as this body-part is visible to the camera, even though the hands pose might be changing. Following that preamble, a segmented area can either belong to the super-class *body-part* or to the derived class *hand*.

After this step is done, we can now track the classified body-parts (hands in this example) over time and send this data to multiple client applications, where the tracked data (movement paths of the detected hands, number of visible fingers on each hand and detected hand-pose) is processed.

3 Results and Discussion

The nature of monocular Time-of-Flight cameras allows the use of simple but efficient algorithms like region growing and edge detection to segment the measured data. Objects which are spatially connected in the 3D point-cloud will be clustered together and can then be classified as described in the foregone chapter. The Results of our implementations are visualized below:

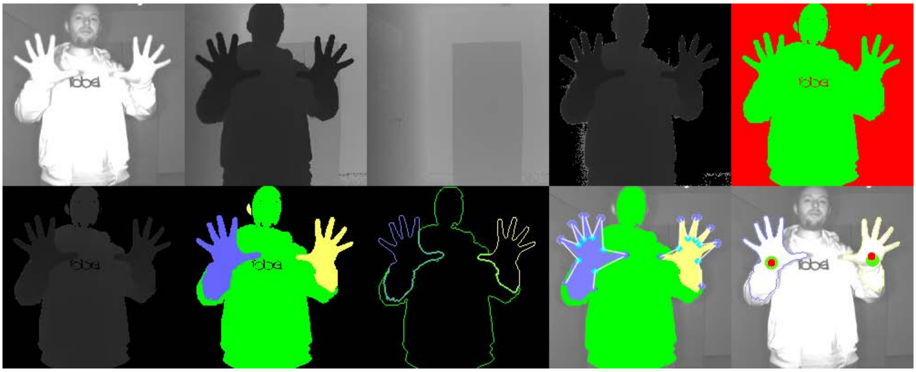

Fig. 1. Image processing chain of the segmentation algorithm

Images in Figure 1 (from left to right, top to bottom):

1. Intensity values (2D greyscale image)
2. Actual depth data (3D floating point image)
3. Initial depth data for background subtraction
4. Background subtraction result
5. Signal quality check
6. Remaining depth data after signal quality check
7. Region growing results
8. Edge detection results
9. Body-part classification through geometrical classifiers
10. Augmented intensity image with tracking results

The ToF camera we used[2] is capable of 3D-data-capturing with 25 frames per second. ToF cameras still have a relatively small resolution compared to state of the art 2D sensors (the camera we used has a sensor-resolution of 204 x 204 pixels) and limited operating range (the model we incorporated is capable of measuring 3D data in a range from 0.3 to 7 meters). As we can see in figure 2, the calculation-time progression of our algorithm directly depends on the number of segmented objects and its time-incrementation is nearly linear. With 9 segmented objects visible in the frame, a number common for our task as

[2] PMD CamCube 2.0, `http://www.pmdtec.com`

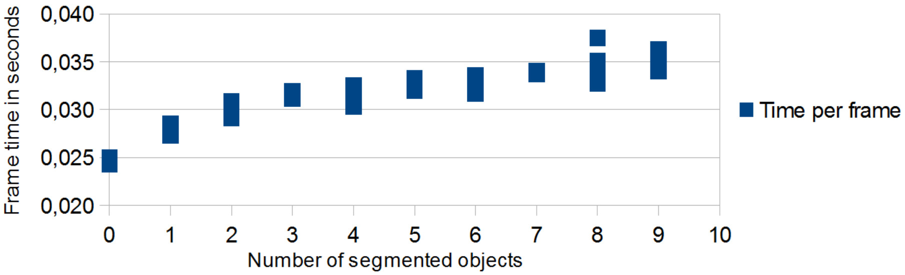

Fig. 2. Statistical analysis of calculation time in correspondence to segmented objects

empirical data shows, our yet un-optimized approach is capable of processing at least 28 frames per second. In a realistic situation we should not have more than 2 users inside the field of view of the camera because of it's aforementioned limited range and resolution. This shows that the implemented system is fast enough for real-time interaction under the described circumstances.

The main problem we encountered was based on the occlusion of objects within the 3D scene, for example when two persons visible in the scene stand or walk behind each other. Without incorporating knowledge-based approaches we were not able to eliminate these occlusion problems for now.

References

1. Parvizi, E., Wu, Q.M.J.: Real-Time 3D Head Tracking Based on Time-of-Flight Depth Sensor. In: 19th IEEE International Conference on Tools with Artificial Intelligence, pp. 517–521 (2007)
2. Parvizi, E., Wu, Q.M.J.: Multiple Object Tracking Based on Adaptive Depth Segmentation. In: Canadian Conference on Computer and Robot Vision, pp. 273–277 (2008)
3. Parvizi, E., Wu, Q.M.J.: Real-Time Approach for Adaptive Object Segmentation in Time-of-Flight Sensors. In: 20th IEEE International Conference on Tools with Artificial Intelligence, pp. 236–240 (2008)
4. Bing, L., Jesorsky, O., Kompe, R.: Robust Real-Time Multiple Object Tracking in Traffic Scenes using an Optical Matrix Range Sensor. In: Intelligent Transportation Systems Conference, pp. 742–747. IEEE (2007)
5. Fan, T.-J., Medioni, G., Nevatia, R.: Recognizing 3-D Objects using Surface Descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1140–1157 (1989)
6. Bab-Hadiashar, A., Gheissari, N.: Range Image Segmentation using Surface Selection Criterion. IEEE Transactions on Image Processing, 2006–2018 (2006)
7. Powell, M.W., Bowyer, K.W., Xiaoyi, J., Bunke, H.: Comparing Curved-Surface Range Image Segmenters. Sixth International Conference on Computer Vision, 286–291 (1998)
8. Satoshi, S., Keiichi, A.: Topological Structural Analysis of Digitized Binary Images by Border Following. Computer Vision, Graphics, and Image Processing, 32–46 (1985)

9. Heckbert, P.: A Seed Fill Algorithm. Graphics Gems I, 275–277 (1990) ISBN-13: 978-0122861666
10. Shaw, J.R.: QuickFill: An efficient Flood Fill Algorithm, `http://www.codeproject.com/gdi/QuickFill.asp`
11. Wobbrock, J.O., Wilson, A.D., Li, Y.: Gestures without Libraries, Toolkits or Training: A 1-Dollar Recognizer for User Interface Prototypes. In: UIST 2007 Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, pp. 159–168 (2007)
12. Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E.: TUIO: A Protocol for Table-Top Tangible User Interfaces. In: Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005), Vannes, France (2005)
13. Welch, G., Bishop, G.: An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, NC (1995)
14. Ringbeck, T.: A 3D Time of Flight Camera for Object Detection. Optical 3-D Measurement Techniques, ETH Zürich (2007)
15. Sklansky, J.: Measuring Concavity on a Rectangular Mosaic. IEEE Trans. Comput., 1355–1364 (1972)
16. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-Time Human Pose Recognition in Parts from a Single Depth Image, Microsoft Research. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1297–1304 (2011)