

**FACULTY OF AUTOMATION AND COMPUTER SCIENCE**

**COMPUTER SCIENCE DEPARTMENT**

**HAND GESTURE RECOGNITION FOR EDUCATIONAL AND PATIENT RECOVERY APPLICATION**

Author: **Paul IANCU**

Advisor: S. l. Dr. Ing. **Razvan ITU**

1. **Project purpose and motivation**

The purpose of this project is to build a system capable of extracting high level features from a static hand gesture and integrating the system with an application. The high-level features that the system will be able to extract are: hand center position; hand orientation; fingers’ state (extended/flexion); center, fingertip and orientation for extended fingers. The application which will use the previously mentioned system will be a game in which the user will be shown a series of drawn hand gestures to which the user must respond by recreating it in a given amount of time. The gestures of the users will be extracted from a live feed using a webcam. This application can have practical usages for treating patients who are recovering from accidents or diseases which impaired their hand control. This application can also be used as an educational tool for teaching young children how to count on their fingers.

The main objectives are:

* Create a robust system for extracting high-level features of from hand gesture
* Create the application which exemplifies the usage of this system

1. **Project description**

This project aims to find, implement and evaluate solutions for the problem of real time hand gesture recognition. Interest for this technology has surged in the last couples of years, with custom pieces of hardware being developed and sold in the consumer market which aim to offer solutions to this problem, such as the Microsoft Kinect sensors and the Oculus Quest virtual reality headset.

The approach chosen in this project is a much simpler one, that offers a less complex model of the hand being tracked than the above-mentioned products. However, the approach chosen for this project has the advantage that it requires the use of only a single web-camera, which comes by default with most laptops/notebooks or can be purchased very cheaply and easily.

In the chosen solution, the user’s palm must always be facing the camera in order for him to interact with our system. This ensures that no relevant part of the hand is occluded. Although the depth information is lost by only using a single camera, the features that are extracted don’t need depth information. The user’s hand should also not occlude other objects that strongly resemble the human hand, such as the user’s face. Basic lighting conditions must also be ensured (the method can’t work in almost pitch black lighting or overly bright conditions).

For the extraction of high-level features from a gesture, only simple image processing techniques are used. Segmentation is used for extracting only the hand portion of the image from an area of interest. The most relevant feature used for segmentation is the palm’s skin color. After filtering the region of interest, we keep only the connected component that has the largest surface area. We apply noise removal techniques on the result and then binarize the image. The distance transform algorithm [4] is applied on the binary image in order to obtain the center of the palm (the point which has the greatest value in the distance transform). The contour of the hand is traced using the algorithm described at [5], and the closest point to the palm center is used to calculate the palm radius. The palm radius is used to find the set of points composing the palm (portion of hand without fingers) and the wrist points. The palm center and wrist points are used to calculate the orientation of the hand. The image is then rotated in order for the hand to have a vertical orientation. The part of the image below the wrist points is removed. The palm mask is applied over the image in order to have only the finger components left. The classification and further segmentation of the connected components from the resulted image is done by using basic anatomic proportions of the human hand.

1. **Related work**

This problem still does not have a clear solution, with many different approaches being proposed, each with their own positives and drawbacks. One of these approaches was through using special types of gloves that the user would wear. These special gloves had sensors on them that could detect the state of the fingers’ flexions. The advantage of this approach is that it can produce very accurate results. The disadvantage of this approach is the fact that users had to wear these gloves, making the interaction seem very artificial and cumbersome. Another common approach is through using sensors which also provide depth information (such as the Microsoft Kinect line of hardware). Although this approach produces very good results, the used hardware does not come by default with most computing systems and is very expensive to purchase. Approaches which use commonly found and relatively inexpensive equipment, such as web cameras, have also been proposed. The main disadvantages of single camera approaches are that depth information is lost and important features of the hand can be occluded by the hand orientation. Some methods try to restore this lost information by using deep learning models. The problem that these solutions face is that they require a large amount of data to be collected and annotated for them to produce good results. While this data gathering process can be done by large companies and institutions, with the resources at my disposal I can’t try to reproduce them. Another disadvantage of the deep learning methods is the fact that it is very hard to predict whether or not a network architecture will produce accurate results. An approach which only uses image processing methods can be seen at [2]. A prototype implementing this approach was made for this project, however a major disadvantage of this approach is that it uses many hand selected values that can’t be easily chosen automatically. The final chosen method for this project is based on [1].

1. **Necessary resources**

Software dependencies:

* Python 3.7
* OpenCV module, version 4.2.0.32
* Numpy module, version 1.18.1
* Scipy module, version 1.4.1
* Easygui module, version 0.98.1

Hardware dependencies:

* Web camera with 640x480 resolution, recording at 30 fps.

1. **Expected results**

Highly accurate classification of static hand gestures is expected, of about 95%. The method will be tested using a custom recorded data set. Each image will be extracted from frames recorded by a webcam. For each type of recognized gesture (combination of fingers extension and flexion state that are natural for a user), there will be samples recorded from at least three different orientations to test the invariance to rotation of the algorithm. The test images will also be done in environments which have different lighting variations. To test the capability of the system to perform real-time classification, custom videos will be recorded using a webcam and check the impact in frame rate that is introduced if the algorithm is run for each frame of the video. The aim for this project is to achieve a 10-15 frames per second benchmark, if not even higher. The tests for real-time performance will be done on a system with Intel Core i7-8750H Processor, having 6 cores, processor base frequency of 2.2 GHz, max turbo frequency of 4.1 GHz and 16GB of RAM.

1. **Project Timeline**

* Study of the main methods regarding hand gesture recognition (2 weeks)
* Study of a convex-hull based method for hand gesture recognition (1 week)
* Implementation of a prototype and testing for the convex-hull based method (2 weeks)
* Study of a salient hand edges and convolution operator based method (1 week)
* Implementation of a prototype and testing for the salient hand edges and convolution operator based method (3 weeks)
* Study of a distance transform and finger segmentation based method (1 week)
* Implementation of a prototype and testing for the distance transform and finger segmentation based method (3 weeks)
* Integration of the last method with the web camera (1 week)
* Implementations of techniques for the method to work properly in various lighting conditions (2 week)
* Implementation of automatic methods for determining the bounding box of the hand (2 weeks)
* Integrating an object tracking algorithm with the method (2 weeks)
* Optimization in order to achieve real time performance (2 weeks)
* Building the example application intended for patient recovery and educational purposes (3 weeks)

1. **Table of contents**
2. Introduction
   1. Context
   2. Motivation
3. Project objectives and specifications
   1. Problem specification
   2. Functional requirements
   3. Nonfunctional requirements
4. Bibliographic research
5. Analysis and theoretical foundations
6. Detailed design and implementation
7. Testing and validation
   1. Introduction
   2. Constructed data set description and results
   3. Performance evaluation
8. User’s manual
   1. System installation
   2. User’s manual
9. Conclusions
   1. Final results
   2. Future Work
10. Bibliography
11. **Bibliography**

**[1]** Chen, Zhi-Hua & Kim, Jung-Tae & Liang, Jianning & Zhang, Jing & Yuan, Yu-Bo. (2014). Real-Time Hand Gesture Recognition Using Finger Segmentation. TheScientificWorldJournal. 2014. 267872. 10.1155/2014/267872.

**[2]** Zhou, Yimin & Jiang, Guolai & Lin, Yaorong. (2015). A novel finger and hand pose estimation technique for real-time hand gesture recognition. Pattern Recognition. 49. 10.1016/j.patcog.2015.07.014.

**[3]** S. Ganapathyraju, "Hand gesture recognition using convexity hull defects to control an industrial robot," 2013 3rd International Conference on Instrumentation Control and Automation (ICA), Ungasan, 2013, pp. 63-67.

**[4]** Felzenszwalb, Pedro F. and Huttenlocher, Daniel P. Distance Transforms of Sampled Functions, TR2004-1963, TR2004-1963 (2004)

**[5]** Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985)