Hand Gesture Recognition using Image Processing

Akshay Thite (65), Abhay Tiwari(66), Manish Shinde(56)

CMPN, D17C

1. Abstract:

Gesture recognition has been a very interesting problem in the Computer Vision community for a long time. This is particularly due to the fact that segmentation of foreground objects from a cluttered background is a challenging problem in real-time. The most obvious reason is because of the semantic gap involved when a human looks at an image and a computer looking at the same image. Humans can easily figure out what's in an image but for a computer, images are just 3-dimensional matrices. It is because of this, computer vision problems remains a challenge.

2. Problem statement

We are going to recognize hand gestures from a video sequence. To recognize these gestures from a live video sequence, we first need to take out the hand region alone removing all the unwanted portions in the video sequence. After segmenting the hand region, we then count the fingers shown in the video sequence to instruct a robot based on the finger count. Thus, the entire problem could be solved using 2 simple steps -

1. Find and segment the hand region from the video sequence.
2. Count the number of fingers from the segmented hand region in the video sequence.

I. Segment the Hand region

The first step in hand gesture recognition is to find the hand region by eliminating all the other unwanted portions in the video sequence. Video sequence is just a collection of frames or collection of images that runs with respect to time.

Before getting into further details, let us understand how could we possibly figure out the hand region.

1. Background Subtraction

First, we need an efficient method to separate foreground from background. To do this, we use the concept of running averages. We make our system to look over a particular scene for 30

frames. During this period, we compute the running average over the current frame and the previous frames. After figuring out the background, we bring in our hand and make the system understand that our hand is a new entry into the background, which means it becomes the foreground object. But how are we going to take out this foreground alone? The answer is Background Subtraction.

2. **Motion Detection and Thresholding**

To detect the hand region from this difference image, we need to threshold the difference image, so that only our hand region becomes visible and all the other unwanted regions are painted as black. This is what Motion Detection is all about. Thresholding is the assignment of pixel intensities to 0's and 1's based on a particular threshold level so that our object of interest alone is captured from an image.

3. **Contour Extraction**

After thresholding the difference image, we find contours in the resulting image. The contour with the *largest area* is assumed to be our hand.Contour is the outline or boundary of an object located in an image. So, our first step to find the hand region from a video sequence involves three simple steps.

I. Background Subtraction
II. Motion Detection and Thresholding
III. Contour Extraction

II. Algorithm used

1. Find the convex hull of the segmented hand region (which is a contour) and compute the most extreme points in the convex hull (Extreme Top, Extreme Bottom, Extreme Left, Extreme Right).
2. Find the center of palm using these extremes points in the convex hull.
3. Using the palm's center, construct a circle with the maximum Euclidean distance (between the palm's center and the extreme points) as radius.
4. Perform bitwise AND operation between the thresholded hand image (frame) and the circular ROI (mask). This reveals the finger slices, which could further be used to calculate the number of fingers.

--Software and Hardware requirements

Pycharm,Camera ,OpenCV,Jetbrains PyCharm

III. Applications:

a) Virtual Reality: Gestures for virtual and augmented reality applications have experienced one of the greatest levels of uptake in computing. Virtual reality interactions use gestures to enable realistic manipulations of virtual objects using one's hands, for 3D display interactions or 2D displays that simulate 3D interactions

b) Games: When we look at gestures for computer games. Freeman tracked a player's hand or body position to control the movement and orientation of interactive game objects such as cars. Konrad et al. used gestures to control the movement of avatars in a virtual world, and Play Station 2 has introduced the Eye Toy, a camera that tracks hand movements for interactive games.

c)Sign Language: Sign language is an important case of communicative gestures. Since sign languages are highly structural, they are very suitable as testbeds for vision algorithms [12]. At the same time, they can also be a good way to help the disabled to interact with computers. Sign language for the deaf (e.g. American Sign Language) is an example that has received significant attention in the gesture literature

IV. Conclusion

In this project, we performed Background Subtraction, Motion Detection, Thresholding and Contour Extraction to nicely segment the hand region from a real-time video sequence using OpenCV and Python.

References

- OpenCV documentation **https://docs.opencv.org/3.0-beta/doc/...**
- Pycharm software **https://www.jetbrains.com/pycharm/dow...**
- Python **https://www.python.org/**