Research Paper Engineering

Multipoint Hand Gesture Recognition Using Robotic Arm Control

* Nishant Madhukar Labhane ** Prashant Harsh
*** Meghan Kulkarni

* 6,Abhineha, Right Bhusari Colony Near Bhimsen Joshi Garden,Kothrud Depot.,Pune

ABSTRACT

We propose a fast algorithm for automatically recognizing a limited set of gestures from hand images for a robot control application. Hand gesture recognition is a challenging problem in its general form. We consider a fixed set of manual commands and a reasonably structured environment, and develop a simple, yet effective, procedure for gesture recognition. Our approach contains steps for segmenting the hand region, locating the fingers, and finally classifying the gesture. The algorithm is
N variant to translation, rotation, and scale of the hand. We demonstrate the effectiveness of the technique on real imagery.

Keywords : HCI,Gesture,Segmentation

Vision-based automatic hand gesture recognition has beena very active research topic in recent years with motivating applications such as human computer interaction (HCI), robotcontrol, and sign language interpretation. The general problem is quite challenging due a number of issues including the complicated nature of static and dynamic hand gestures, complex backgrounds, and occlusions. Attacking the problem in its generality requires elaborate algorithms requiring intensive computer resources. What motivates us for this work is a robot navigation problem, in which we are interested in controlling a robot by hand pose signs given by a human. Due to real-time operational requirements, we are interested in a computationally efficient algorithm.

Early approaches to the hand gesture recognition problem in a robot control context involved the use of markers on the finger tips [1]. An associated algorithm is used to detect the presence and color of the markers, through which one can identify which fingers are active in the gesture. The inconvenience of placing markers on the user's hand makes this an infeasible approach in practice. Recent methods use more advanced computer vision techniques and do not require markers. Hand gesture recognition is performed through a curvature space method in [2], which involves finding the boundary contours of the hand. This is a robust approach that is scale, translation and rotation invariant on the hand pose, yet it is computationally demanding. In [3], a vision-based hand pose recognition technique using skeleton images is proposed, in which a multi-system camera is used to pick the center of gravity of the hand and points with farthest distances from the center, providing the locations of the finger tips, which are then used to obtain a skeleton image, and finally for gesture recognition. A technique for gesture recognition for sign language interpretation has been proposed in [4]. Other computer vision tools used for 2D and 3D hand gesture recognition include specialized mappings architecture [5], principal component analysis [6], Fourier descriptors, neural networks, orientation histograms [7], and particle filters [8].

Our focus is the recognition of a fixed set of manual commands by a robot, in a reasonably structured environment in real time. Therefore the speed, hence simplicity of the algorithm is important. We develop and implement such a procedure in this work. Our approach involves segmenting the hand based on skin color statistics, as well as size constraints. We then find the center of gravity (COG) of the hand region as well the farthest point from the COG. Based on these preprocessing steps, we derive a signal that carries information on the activity of the fingers in the sign. Finally we identify the sign based on that signal. Our algorithm is invariant to rotations, translations and scale of the hand. Furthermore, the technique does not require the storage of a hand gesture database in the robot's memory. We demonstrate the effectiveness of our approach on real images of hand gestures.

2. Hand Gesture Recognition

Consider a robot navigation problem, in which a robot responds to the hand pose signs given by a human, visually observed by the robot through a camera. We are interested in an algorithm that enables the robot to identify a hand pose sign in the input image, as one of five possible commands (or counts). The identified command will then be used as a control input for the robot to perform a certain action or execute a certain task. For examples of the signs to be used in our algorithm, see Figure 1. The signs could be associated with various meanings depending on the function of the robot. For example, a "one" count could mean "move forward", a "five" count could mean "stop". Furthermore, "two", "three", and "four" counts could be interpreted as "reverse", "turn right," and "turn left."

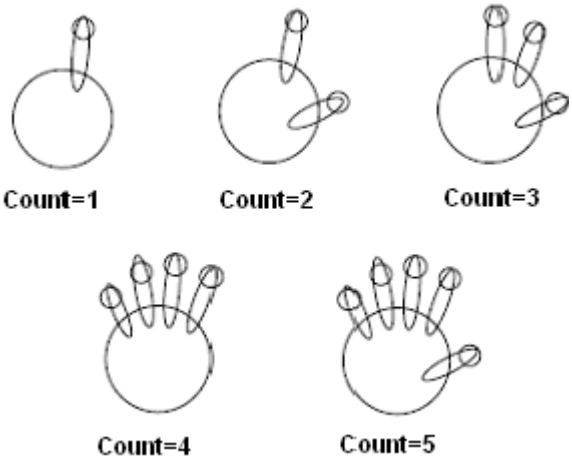


Figure 1: Set of hand gestures, or "counts" considered in our work. Adapted from [8].

Our proposed method of hand gesture recognition consists of the following stages:

- Webcam Interfacing : JMyron library is used to interface web cam and obtain continuous web cam video feed.
- Image Retreival : Images are retreived from video feed at regular intervals. (Couple of times in a second depending on CPU speed)
- Image Preprocessing : Either Sharpen Filter or Guassian Blur filter is applied to an image that is either too blurred or too sharp for further processing.
- Grayscale Conversion : Color or RGB image is converted to grayscale.
- Thresholding : The image is thresholded based on colour threshold value provided by user. The output is a pure binary image.
- Trigonometric Circular Scan : A complete scan of image is done in circular direction so that we can detect the black and white pixels.
- Finger Detection : We detect fingers and neglect wrist and other noise in scan.
- OS Interface : Keyboard or Mouse events are generated based on gestured recognized. Java Robot API is used to generate these events.

2.1 Webcam Interfacing
A webcam is a video camera that feeds its images in real time to a computer. JMyron library is used to interface web cam and obtain continuous web cam video feed.

2.2 Image Retreival
In image retrieval our system is browsing, searching and retrieving image captured by web camera i.e shots from video sequence.

2.3 Image Preprocessing
A Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales see scale-space representation and scale-space implementation.

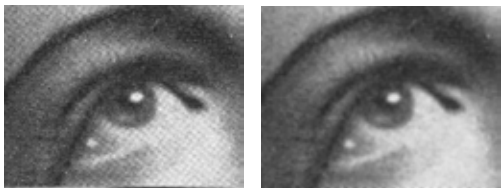


Fig: Gaussian blur can be used in order to obtain a smooth grayscale digital image of a halftone print

The Gaussian blur is a type of image-blurring filter that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image. The equation of a Gaussian function in one dimension is

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{x^2}{2\sigma^2}}$$

in two dimensions, it is the product of two such Gaussians, one in each dimension:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{x^2+y^2}{2\sigma^2}}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. When applied in two dimensions, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point

2.4 Grayscale Conversion
Color or RGB image is converted to grayscale. A grayscale or greyscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. Grayscale images are distinct from one-bit bi-tonal black-and-white images, which in the context of computer imaging are images with only the two colors, black, and white (also called bilevel or binary images). Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation The intensity of a pixel is expressed within a given range between a minimum and a maximum, inclusive. This range is represented in an abstract way as a range from 0 (total absence, black) and 1 (total presence, white), with any fractional values in between. This notation is used in academic papers, but it must be noted that this does not define what “black” or “white” is in terms of colorimetry.

Fig: RGB to Grayscale conversion

To convert any color to a grayscale representation of its luminance, first one must obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding, by gamma expansion. Then, add together 30% of the red value, 59% of the green value, and 11% of the blue value(these weights depend on the exact choice of the RGB primaries, but are typical). Regardless of the scale employed (0.0 to 1.0, 0 to 255, 0% to 100%, etc.), the resultant number is the desired linear luminance value; it typically needs to be gamma compressed to get back to a conventional grayscale representation.

2.5 Thresholding
The image is thresholded based on colour threshold value provided by user. The output is a pure binary image.

It is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary image.During the thresholding process, individual pixels in an image are marked as “object” pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as “background” pixels otherwise. This convention is known as threshold above. Variants include threshold below, which is opposite of threshold above;

threshold inside, where a pixel is labeled "object" if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside. Typically, an object pixel is given a value of "1" while a background pixel is given a value of "0." Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's labels.

Fig: Original image

Fig:Example of a threshold effect used on an image

2.6 Trigonometric Circular Scan

In trigonometric circular scan we Calculate COG of hand so that it can recognize how many fingers are there in gesture.

The segmented hand region, we calculate its centroid , or center of gravity (COG), (x, y) , as follows:
x=(x1+x2+…….xN)/N y=(y1+y2+……..yN)/N

where xi and yi are x and y coordinates of the ith pixel in the hand region, and k denotes the number of pixels in the region.

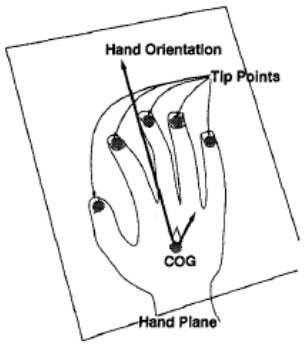
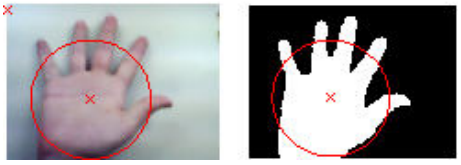


Figure : Center of gravity and extreme points of the hand.[3]



We draw a circle whose radius is 0.7 of the farthest distance from the COG. Such a circle is likely to intersect all the fingers active in a particular gesture or "count." Just to provide a visual flavor, Figure 4 demonstrates the execution of the steps described so far on a sample image.

2.7 Extracting a 1D Signal and Classification

We now extract a 1D binary signal by tracking the circle constructed in the previous step. Ideally the uninterrupted "white" portions of this signal correspond to the fingers or the wrist. By counting the number of zero-to-one (black-to-white) transitions in this 1D signal, and subtracting one (for the wrist) leads to the estimated number of fingers active in the gesture. Estimating the number of fingers leads to the recognition of the gesture.

2.8 Scale, Rotation, and Translation Invariance

Our proposed algorithm is scale invariant. Meaning that the actual size of the hand size and its distance from the camera do not affect interpretation. It is rotation invariant, since the orientation of the hand does not hinder the algorithm from recognizing the gesture. In addition, the position of hand is also not a problem.

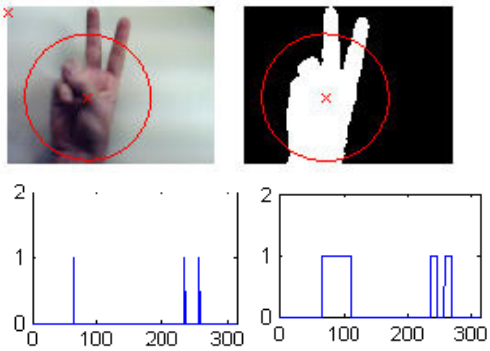


Figure: Sample result for a "two" count. Top left: Original image and the circle constructed. Top right: Segmented hand and the circle constructed. Bottom left: Zero-to-one transitions in the 1D signal extracted. Bottom right- The 1D signal q

3. Conclusion

We proposed a fast and simple algorithm for a hand gesture recognition problem. Given observed images of the hand, the algorithm segments the hand region, and then makes an inference on the activity of the fingers involved in the gesture. We have demonstrated the effectiveness of this computationally efficient algorithm on real images we have acquired. Based on our motivating robot control application, we have only considered a limited number of gestures. Our algorithm can be extended in a number of ways to recognize a broader set of gestures. The segmentation portion of our algorithm is too simple, and would need to be improved if this technique would need to be used in challenging operatingconditions.

4. Acknowledgment

We would like to thank all the researchers working on this field who in one way or another guided us on achieving our goals.

REFERENCES

[1] J. Davis and M. Shah "Visual Gesture Recognition", IEE Proc.-Vis. Image Signal Process., Vol. 141, No.2, April 1994. [2] C.-C. Chang, I.-Y Chen, and Y.-S. Huang, "Hand Pose Recognition Using Curvature Scale Space", IEEE International Conference on Pattern Recognition, 2002. [3] A. Utsumi, T. Miyasato and F. Kishino, "Multi-Camera Hand Pose Recognition System Using Skeleton Image",IEEE International Workshop on Robot and Human Communication, pp. 219-224, 1995. [4] Y. Aoki, S. Tanahashi, and J. Xu, "Sign Language Image Processing for Intelligent Communication by Communication Satellite", IEEE International Conf. On Acoustics, Speech, and Signal Processing, 1994. [5] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff, "3D Hand Pose Reconstruction Using Specialized Mappings",IEEE International Con. on Computer Vision, pp. 378-385, 2001. [6] C. Tomasi, S. Petrov, and A. Sastry, "3D = Classification+ Interpolation", IEEE International Conf. on Computer Vision, 2003.[7] W. T. Freeman and M. Roth, "Orientation Histograms for Hand Gesture Recognition", IEEE International Conf. on Automatic Face and Gesture Recognition, 1995. [8] L. Bretzner, I. Laptev, and T. Lindberg, "Hand Gesture Recognition using Multi-Scale Color Features,Hierarchical Models and Particle Filtering", IEEE International Conf. on Automatic Face and Gesture Recognition, 2002. [9] J. Brand and J. Mason, "A Comparative Assessment of Three Approaches to Pixel-level Human Skin Detection",IEEE International Conference on Pattern Recognition,2000. [10] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Prentice-Hall, 2nd edition, 2002. [11] R. M. Haralick, and L. G. Shapiro, Computer and Robot Vision, Volume I, Addison-Wesley, 1992, pp. 28-48 Vision, Volume I, Addison-Wesley, 1992, pp. 28-48.