Journal of Advanced Computer Science & Technology, 1 (2) (2012) 64-72
©Science Publishing Corporation
www.sciencepubco.com/index.php/JACST

Dynamic Vision Sensor Camera Based Bare Hand Gesture Recognition

Kashmera Khedkkar Safaya, Prof.Rekha Lathi

Lecturer, Information Technology, Pillai's Institute of Information Technology
Research and Studies
Kashmera.k@gmail.com
Assistant Professor, Computer Department, Pillai's Institute of Information
Technology Research and Studies
rekha_lathi@yahoo.com

Abstract

This Paper proposes a method to recognize bare hand gestures using dynamic vision sensor (DVS) camera. DVS camera only responds asynchronously to pixels that have temporal changes in intensity which different from conventional camera. This paper attempts to recognize three different hand gestures rock, paper and scissors and using those hand gestures design mouse free interface.

Keywords: *Dynamic vision sensor camera, Hand gesture recognition*

1. Introduction

Computer is used by many people either at their work or in their spare-time. Special input and output devices have been designed over the years with the purpose of easing the communication between computers and humans, the two most known are the keyboard and mouse. Every new device performs more complicated communication with the computer. This has been possible due to the result oriented efforts made by computer professionals for creating successful human computer interfaces. As the complexities of human needs have turned into many folds and continues to grow so, the need for Complex programming ability and intuitiveness are critical attributes of computer programmers to survive in a competitive environment. The computer programmers have been incredibly successful in easing the communication between man and computer. In other areas where 3D information is required, such as computer games, robotics and design, other mechanical devices such as roller balls, joysticks and data gloves are used. Humans communicate mainly by vision and sound, therefore, a man machine interface would be more intuitive if it made greater use of vision and audio recognition. Another advantage is that the user not only can communicate from a distance, but need have no physical contact with the computer. However, unlike audio commands, a visual system would be preferable in noisy environments or in situations where sound would cause a disturbance. The visual system chosen was the recognition of hand gestures. The amount of computation required to process hand gestures is much greater than that of the mechanical devices.

2. Motivation

Primary concern in hand gesture recognition is that they could make it possible for users to communicate with computerized equipment without need for external control devices. Gestures are a powerful means of communication among humans. In fact, gesturing is so deeply rooted in our communication that people often continue gesturing when speaking on the telephone. Hand gestures provide a separate complementary modality to speech for expressing ones ideas. Information associated with hand gestures in a conversation is degree, discourse structure, spatial and temporal structure. So, a natural interaction between humans and computing devices can be achieved by using hand gestures for communication between them. The key problem in gesture interaction is how to make hand gestures understood by computers. According to following graph shown in figure 1 almost 40% of all gestures are done with one hand and multiple hands at 20% and the rest are distributed between the other body parts.

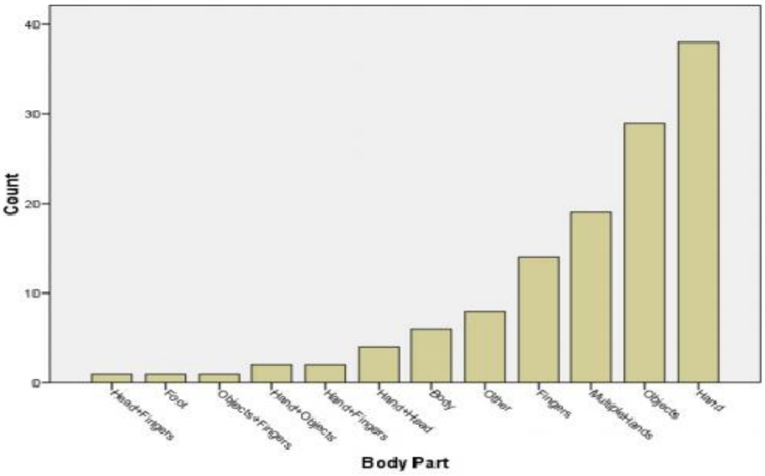


Fig.1 Gestures and Body Parts [5]

The approaches present can be mainly divided into “Data Glove based” and “Vision Based” approaches. The Data Glove based methods use sensor devices for digitizing hand and finger motions into multi-parametric data. The extra sensors make it easy to collect hand configuration and movement. However, the devices are quite expensive and bring much cumbersome experience to the users. In contrast, the Vision based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices. A review of existing literature shows that there has been a considerable amount of work done I the area of gestures based interactions, and its application in various domains. In addition, quite a bit of research exists around how gestural inputs are recognized and processed, and on the results of such gestural interactions.

3. Proposed System

The aim is to design mouse free interface and identify the hand poses or gestures. All stages of proposed systems are shown in following fig. 2

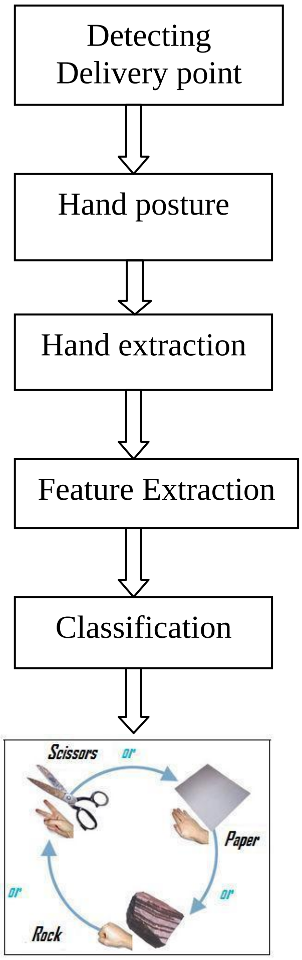
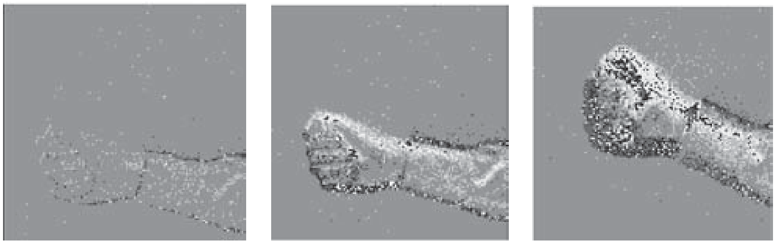


Fig.2 Block diagram of proposed system

These stages are firstly to detect the delivery point then recognize the posture of the hand and noise reduction using connected component technique later extract hand then feature extraction as to extract shape of hand and lastly classification of shape to recognize the gesture.

3.1 Detecting Delivery Point

The delivery point refers to a point where it delivers a throw. The movement of the hand is slower as the hand reaches the end of the delivery point in order to stop moving at the delivering phase. DVS cameras are only response to the pixels with luminance intensity, and send events of only those pixels. Since there is little movement of the hand at the delivery point, the number of events within a frame will be dramatically reduced. Once the delivery point is detected, system can classify different throws using a single posture captured at the time of delivery point. To find the delivery point, track the number of events for each frame, and if the number of events in the frame less than the given threshold TH, the frame is regarded as a delivery point. If TH is too large, the frame detected as delivery point can be different from actual hand throw. Even if the hand pose in the detected frame is the same as the actual throw, the frame may not be appropriate one for detecting hand pose. The reasons are that the frame is likely to contain too many events, which will increase the computational cost, and the shape of the hand can be not clear by thickening the boundary as shown in following figure 3(c). In contrast, if the size of TH is too small, as shown in figure (a) the frame detected as a delivery point may contain too few events to recognize the shape of hand. Therefore, TH needs to be carefully chosen. Figure 3(a) (b) and (c) shows the frames detected as a delivery point.



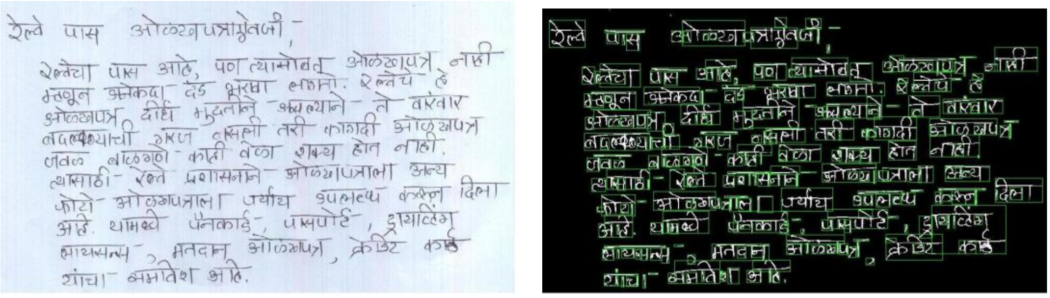
a) TH =1,000 b) TH=5,000 c) TH=10,000

Fig.3 Frames at the delivery point

However stopping the hand movement does not necessarily mean that we are delivering a throw. Therefore it track the direction of movement for each frame, and the first frame with number of events less than TH during the downward movement is considered as delivery point. To summarize, the frame is regarded as a delivery point when it has the number of events less than the given threshold (TH) and the hand is moving downward, where the moving direction of the hand referred as is determined as delivery point.

3.2 Hand posture recognition

Once the delivery point is detected, the frame at the delivery point is used to recognize hand pose. Since there are some noisy events, a noise filtering process is first conducted using connected component analysis. It is often useful to extract regions which are not separated by a boundary. Any set of pixels which is not separated by a boundary is call connected. To apply connected component analysis, the stream of events of a frame is represented by a 128 by 128 matrix. There are two types of events: on and off events. Although a single matrix can be generated by ignoring the event types each of which represents on and off events are used, and connected component analysis is applied to each matrix separately. Since events of the same type are likely to occur closely in space, we expect that the two matrix representation can filter out noisy events that are surrounded by events of a different type. Connected component analysis is shown in following in fig4.



रेल्वे पास ओळखपत्राबाबतजी,

रेल्वेचा पास आहे, पण त्यासोबत ओळखपत्र नाही म्हणून अनेकदा दंड भरावा लागतो. रेल्वेचे हे ओळखपत्र दीर्घ मुदतीने असल्याने ते वारंवार बदलण्याची गरज नसली तरी कागदी ओळखपत्र जवळ बाळगणे काही वेळा शक्य होत नाही. त्यासाठी रेल्वे प्रशासनाने ओळखपत्राला अन्य फोटो ओळखपत्राचा पर्याय उपलब्ध करून दिला आहे. यामध्ये पॅनकार्ड, पासपोर्ट, ड्रायव्हिंग लायसन्स, मतदान ओळखपत्र, क्रेडिट कार्ड यांचा समावेश आहे.

(a) (b)

Fig.4 (b) Connected component method to reduce noise

3.3 Hand Extraction

After the filtering process, the hand region is first extracted. Some portion of the forearm is presented in the same frame. Since the various lengths of forearm presented in the frame do not provide useful information to distinguish different hand postures. To do this, first estimate the width of the hand along the horizontal axis, and then record changes in width from right to left(from forearm area to hand) to locate the point with the largest width increase. Hand postures, only the hand region is used for classification. To extract the hand, first the wrist point is found. It works as follows: The frame is segmented into bH bins along the horizontal axis as shown in

Fig.6, and the segmentation point which lies in between the *ith* bin and the *i+1* bin is denoted as ∂i the size of the bin is |*max(x)-min(x)*|/bH, where *max(x)* and *min(x)* represent the largest and the smallest x-address of events respectively, and *bH* is the number of bins. Estimate the width of the hand for each bin, which is denoted by *di*,, for *ith* bin. Calculate the width change for each segmentation point ∂i by subtracting the width of the hand in the left bin of ∂i i.e di-di+1. From forearm area to hand, find the segmentation point *p* with maximum changes. All the events whose x- address is smaller than *p* are regarded as a hand region.

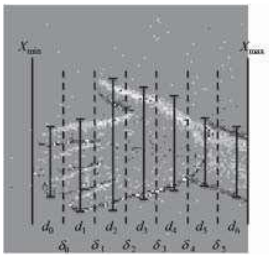


Fig.5 Hand Extraction [1]

3.4 Feature Extraction

Feature extraction is very important in terms of giving input to a classifier. The aim of this phase is to find and extract features that can be used to determine the meaning of a given gesture. One simple feature which can represent the shape of a hand might be the width distribution across the hand. Similar to the method to extract the wrist point, we segment the extracted hand into *bF* bins along the horizontal axis as shown in the fig- 7, and then estimate the hand width for each bin. A sequence of hand widths shows how the size of width changes along the horizontal axis. Since the absolute value of width can be different depending on person and the distance between the hand and the camera we use relative width, which can be obtained by dividing the absolute width of the bin by the sum of absolute widths over all bins. It is shown in equation 1,where relWidth(i) and absWidth(i) represent the relative and the absolute size of width at *i*th bin respectively.[1]

$$\text{relWidth(i)=absWidth(i)/ } bF\ absWidth(i) \qquad (1)$$

Similar to the process of hand extraction, the number of bins, b*F* should be appropriately set. If the number of bins is too small, key features cannot be captured. An extreme case is using one bin. Although computationally less expensive, the pattern of width and the hand shape cannot be inferred from only the average hand width. The opposite extreme case is using the number of column pixels of the abstracted hand data as *bF*. Although the shape of hand can be represented at the fine-grained level, it is computationally expensive. More importantly, by focusing on too specific and too local patterns of data, we may fail to extract more general data patterns. If there are more possible gestures such as stretching fingers. Each finger can have only one of two states as stretched or folded, so that most of the commonly used hand gestures are combinations of the states of all five fingers. As a first step toward the recognition of the states of fingers, we attempt to recognize the number of fingers in each bin without distinguishing whether a finger is stretched or folded or by specifically identifying the fingers in each bin. To find the number of fingers within a bin, connected component analysis is used. Instead of filtering out the component with the smallest number of events as we did in the filtering process, the number of connected components for each bin is counted.

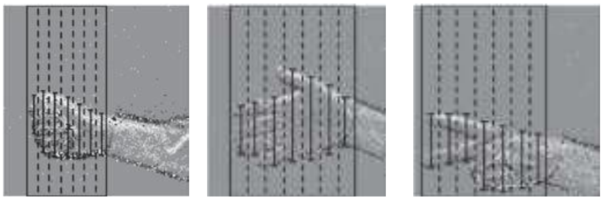


Fig.6 Frames at the Delivery Point [1]

Next, as a step toward the recognition of the state of fingers, we attempt to recognize whether any of the fingers except the thumb is stretched or not. This could be easily recognized by comparing ratio of the length from the tip of fingers to the thumb to the length from the thumb to the wrist as shown in following fig-8. This can roughly estimate the location of thumb by locating the column with the maximum width. This feature is called as Horizontal Ratio.

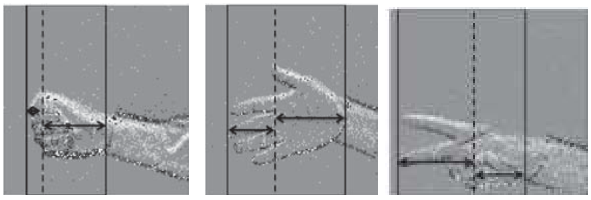


Fig.7 Horizontal ratios [1]

3.5 Classification

Once the features are extracted, a machine learning algorithm is applied to build a model for prediction. In this paper, we use naive Bayes algorithm which is a simple version of Bayesian network with an assumption that all the features are independent given the class. This method is an effective and fast method for static hand gesture recognition. This method is based on classifying the different gestures according to geometric-based invariants which are obtained from image data after segmentation; thus, unlike many other recognition methods, this method is not dependent on skin colour. Gestures are extracted from each frame of the video, with a static background. The Naïve Bayes classifier works on a simple, but comparatively intuitive concept. An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters necessary for classification. In this width of palm and width of fingers are as input to Bayes technique and output will be classifying rock, paper and scissors hand gestures. It requires small amount of training data for classification. Training data for hand gestures are class features i.e. rock, paper, and scissors. Training data for mouse free interface are right click, left click and mouse click and class variables are rock, paper, and scissors as shown in following fig-9[5]

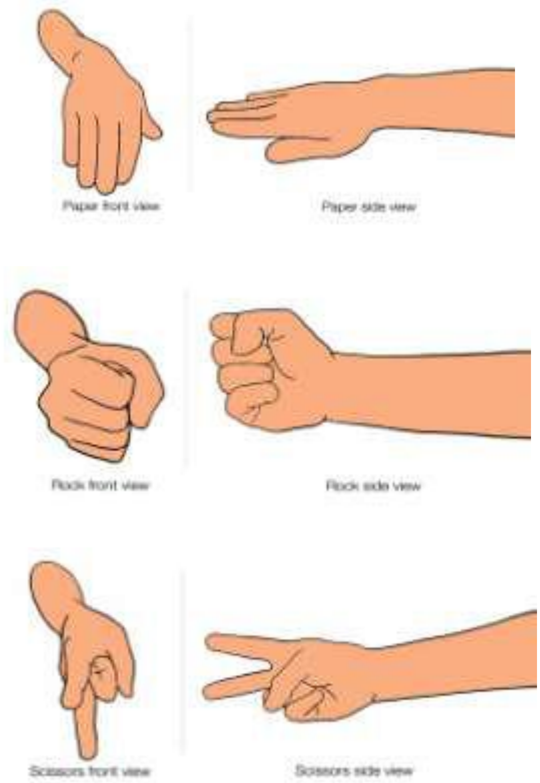


Fig.8 After classification rock, paper, scissors hand gestures

4. Conclusion

This paper proposes a method to classify bare hand gesture using dynamic vision sensor (DVS) camera. This is focused on classifying three different gestures (rock, paper, and scissors) and using those three gestures mouse free interface can be designed. In this paper there are five stages to recognize different hand gestures. To recognize a throw, the point where the player delivers a throw should be detected first; called as called as point as delivery point. Once the delivery point is detected, it can classify different throws using only a single posture captured at the delivery point. After delivery point is detected the noise will be reduced and hand extraction will be done. Now to find the shape of hand, feature recognition is carried out for further classification stage. After recognizing hand poses those hand gestures will be use in mouse free interface wherein left click, right click function will be included. Using of this mouse free interface there will ease of using mouse function by gestures.

References

[1] Eun Yeong Ann, Jun Haeng Lee, Tracy Mullen, John Yen,”*Dynamic Vision Sensor Based are Hand Gesture Recognition*”,978-1-4244-9915-1/11/$26.00,2011IEEE.

[2] Pragati Garg, Naveen Aggarwal and Sanjeev Sofat,” Vision Based Hand Gesture Recognition”, World Academy of Science, Engineering and Technology 49 2009.

[3] C.A.Bouman,”Connected Component Analysis”, Digital Image Processing-9, Jan,2012.

[4] Preeta Rajamani,” Best Practices in Gestural Design”, Bentley University

[5] http://www.worldrps.com/game-basics.

[6] Sanjay Meena,”A study on hand gesture recognition. Technique”, Departement of Electronics and Communication engineering National Institute of Technology, 2011.