Preliminary Experimental Study of Marker-based Hand Gesture Recognition System

C. Diaz and S. Payandeh
Experimental Robotics and Imaging Laboratory, Engineering Science Department, Simon Fraser University, Canada
Email:carlosd@sfu.ca; payandeh@sfu.ca

Abstract—**We present the initial studies of a user computer interface based on hand gestures. A set of reflective markers on a glove are used as a simplified basis for hand pose detection. Using coaxial infrared illumination we can maximize the contrast of the markers with respect to the background. The objective is to recognize the shape of various markers with their corresponding position and orientation in the image plane. The system is able to associate the shape of the markers and their corresponding location and orientation angles to various user interface commands. Marker Gestures can be static and dynamic. Whereas static gestures refer to particular hand poses, dynamic gestures correspond to trajectories described with some particular poses. The experiments show that the combined use of the proposed lighting technique, Hu moments and convex hull analysis greatly simplifies the segmentation and identification of the markers allowing the system to obtain reliable information about the pose and orientation of the hand of the user.**

Index Terms—**blobanalysis,hand gestures, hu moments, shape recognition.**

I. Introduction

In recent years, there have been considerable efforts in the area of computing games platforms to develop innovative computer interfaces. Traditional input devices like joysticks and keyboards are being replaced by new elements capable of supporting gesture-based interactions with games using a combination of action recognition and motion tracking algorithms. In the particular cases of Nintendo Wii and Sony Playstation, a manual controller is still required. A different paradigm was developed by Microsoft with its Kinect device. The Kinect system is able to locate a user with a range sensor allowing the player to input commands to the console with the use of body gestures and poses.

In this paper we propose a computer interface system based on hand gestures. A user wearing a glove with reflective markers is able to code simple commands to a computer by changing the pose, position and orientation of the glove in front of a camera. Images captured from the camera are processed in a computer to extract information related to the configuration of the hand (hand pose) and the position of the hand on the image plane.

Manuscript received August 2, 2013; revised November 5, 2013

In our proposed implementation we used a camera to receive information of the hand pose. We require the gesture recognition system to be reliable under changing backgrounds and environmental lighting conditions. Fig. 1 shows the conceptual diagram.

Images were captured with a camera using an infrared ring light to provide controlled illumination conditions. The glove was designed using two different patterns corresponding to the palm and the back of the hand and smaller markers the fingers (seen from the back side when fingers are extended). Combinations of poses are used as events to trigger the capture of a trajectory using a finite state machine. Both static and dynamic gestures are used to represent the alphabet of commands. Basic hand poses, like open and close hand, and the number of extended fingers were tested for detection performance.

The images were analyzed in a PC using OpenCV 2.0 and Microsoft Visual Studio 6. The image processing and pattern identification tasks were performed by using contour analysis (blob analysis) and comparing values for the Hu moments and hull convexity of the contours. Invariant moments (Hu moments) and blob analysis provided statistical properties related to the shape of the primitive patterns markers (patterns present on the glove) allowing the identification of the shape of the markers in any rotation on the X-Y plane.

The paper is organized as follows: Section II presents the review of the related literature;Section III explains the image acquisition setup. Section IV presents the gesture recognition paradigm. Section V and VI present the algorithmic tools used and the software architecture. Results and Conclusions are in sections VII and VIII.

II. Previous Work

In the last few decades, hand gesture recognition has been a popular field of research. Depending on the design premises, environmental conditions, available hardware and the requirement of wearable devices, authors have tested different paradigms.

Considering the acquisition devices, it can be mentioned that an important number of these developments are based on color video in controlled environments where the users are able to use their hands without any particular marker device (bare hand) [1]-[3]. Gesture recognition systems working with this approach usually give an important weight to the color of the skin. This parameter, though easily trainable, should be

doi: 10.12720/joace.2.3.242-249

retrained each time a new user is testing the system, due to variations in the skin color between users. In some other cases, a combination of a skin color model, motion detection and edge detection is used to overcome changes in background and illumination [4], [5].

Using color video images and color gloves, some authors [6]-[8] have been able to impose color constraints useful to achieve adequate segmentation results to isolate the hand from the background. This technique is useful when the environment is at least partially controlled, with known lighting and the absence of objects with similar colors to the glove.

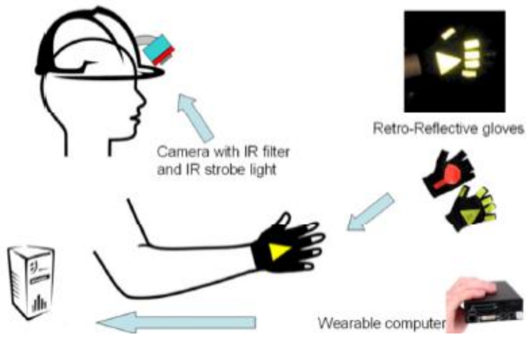

Figure 1. Initial concept scheme.

Data gloves are input devices that have been used with some success to find precise input information about the angles of every joint of the user hand. From the kinematic model of the hand and using a set of sensors located in every joint, it is possible to find the corresponding configuration of the hand. This approach [9] has not gained popularity due to the high cost of the required hardware. The use of accelerometers has been proposed to recognize gestures. In some cases, fusing the input signal from several sensors like electromyography sensors and even 2D images to achieve improved perception of the user hand [10]-[12]. All these cases have in common the use of wearable active electronics that could be inconvenient in some settings.

Different technologies allow obtaining depth information. Stereo vision systems have been used with some success in gesture detection [13]. This approach is less sensitive to color and illumination at the cost of higher hardware requirements and software complexity. On the other hand, time-of-flight cameras [14], [15]and the more recent Kinect [16] generate depth images with a minimum overhead to the host computer. In these cases, a grey scale depth image provide a simple way to get adequate segmentation of the arm and hand of the user based on how close they are to the camera.

Motion capture systems equipped withretroreflective markers have been utilized to improve the precision in modeling human motion [17]. Using synchronized coaxial lighting it is possible to increase the contrast of passive markers placed on different parts of the human body. With this technique, it is possible to work in uncontrolled environments and even outdoors and still obtain input images with high contrast on the markers. The use of reflective markers allows these systems to obtain the required location of the markers in real time due to the improved contrast of the reflective marker, requiring lower computational resources.

Regarding gesture recognition, some software techniques have been proposed for the identification task. Blob analysis and convex hull provide cues to classify hand poses [1], [18] when compared with a database.

Finite state machines [19], [20] have been used to represent different stages in a dynamic gesture. Each hand pose with a relative time duration is a distinct phase in a generic gesture.

The use of neural networks [21]is a common approach for dealing with cluttered environments and natural shapes (as the case of human hands). This approach requires training stages in case of changes in the initial data. Additionally, the reliability in the recognition results depends on the database used to train the system.

The use of Moments in pattern matching has been considered by some researchers and authors [22], [3]. Spatial and Central moments offer information about the location of the center of mass of a particular contour. Additionally, the Hu Moments offer a reliable and fast tool to make successful pattern matching due to their properties invariant to dimension, rotation and translation [23], [24].

III. Image Acquisition Setup

The use of retroreflective markers tends to saturate light in images eliminating color information. For this reason we selected in our setup, a single channel grayscale image sensor. This camera with embedded illumination source was used to obtain images of reflective markers on a glove. Fig. 2 shows the image of our experimental setup consisting of a camera, illumination source and a glove embedded with reflective markers.

A. Camera

For our study we have selected a B&W digital Ethernet camera IMPACT A10 from PPT VISION(752x480 pixels).

B. Illumination

The illumination source is a coaxial ring light array of high power Infrared LEDs (Infrared LED ring light PPT A-Series camera). IR lighting have the property of being invisible to the human eye, providing at the same time consistent illumination.

The imaging capture software (IMPACT from PPT VISION) provides digital controls of the physical properties of the camera. It is possible to adjust the gain of the camera to fixed or programmable values avoiding the auto gain function standard in most analog cameras. Additionally, higher S/N ratios can be achieved by synchronizing the trigger of the camera and shutter time with the LED light.

C. Marker Glove

A custom designed glove provided the location target. For this purpose a black glove with geometrically shaped sections of reflective tape (3M Scotchlite) was used. Retroreflectors return light back to the light source along the same light direction with a minimum scattering of light.

A light ray is reflected back along a vector that is parallel to but opposite in direction from the light source. For this project, the glove enhanced with the reflective fabric provided a simplified skeleton-like structure corresponding to the palm of the hand and the fingers.

Initial tests were performed with the imaging acquisition subsystem to determine the best parameters for the image acquisition. Fig. 3(a) shows the image obtained using the environmental lighting present in our lab. From the image it is possible to obtain the glove (and markers). But it will require a considerable amount of computing resources to achieve an adequate segmentation due to the cluttered background. As we mentioned before, a technique like background subtraction would allow a good segmentation of the markers of the glove, but re-initialization would make this approach inconvenient in our case.

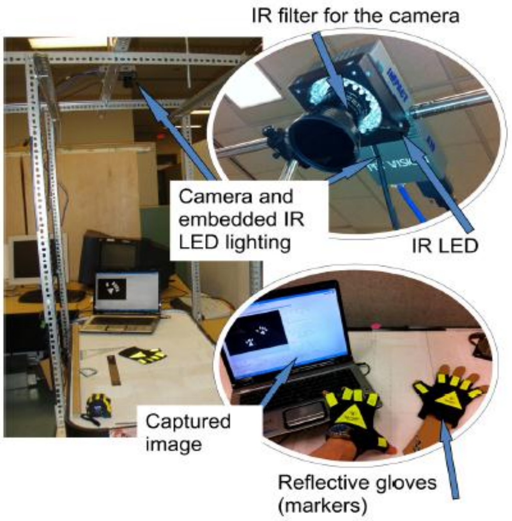

Figure 2. Image acquisition setup: Camera, infrared ringlight and infrared filter.

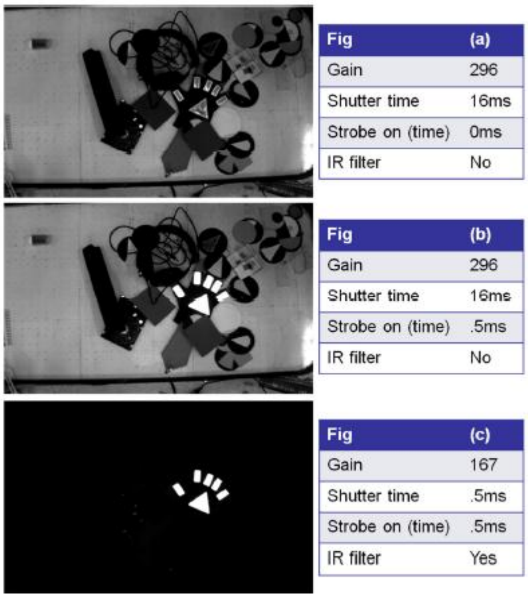

Fig	(a)
Gain	296
Shutter time	16ms
Strobe on (time)	0ms
IR filter	No

Fig	(b)
Gain	296
Shutter time	16ms
Strobe on (time)	.5ms
IR filter	No

Fig	(c)
Gain	167
Shutter time	.5ms
Strobe on (time)	.5ms
IR filter	Yes

Figure 3. Different capture and optical settings for image acquisition.

Fig. 3(b) shows the input image obtained with the use of the LED coaxial ring light. The use of coaxial illumination combined with retro reflectors increased the contrast of the reflector, simplifying the location of the markers on the image. Fig. 3(c) shows the result of reducing the effect of environmental light sources. This process is achieved by combining 2 effects. Originally, (in Fig 3(a) and 3(b)) the shutter time used in the camera was 16ms. The shutter time refers to the time that the shutter remains open when taking a new image. Along with the aperture of the lens, it determines the amount of light that reaches the sensor. While Fig. 3(a) shows the effect obtained just by using the environmental light, the Fig. 3(b) shows the effect of the environmental light + the light coming from the controlled LED ring light. Considering that the LED light can be strobed for short periods of time (Ex .5ms), we decided to limit also the shutter time of the camera to be the same of the strobe light duration (.5ms) and in synchrony with it (both camera and lighting).

An infrared filter was used in front of the camera lens to improve the contrast of the image(Hoya R-72, which passes 720nm and above). This camera is sensitive to both white light and infrared. By adding this IR band-pass filter to the optical setup we could virtually eliminate all the remaining effects coming from the surrounding light sources.

IV. Basic Gestures Definition

Static gestures correspond to the pose of the hand (or glove) at any given time. Dynamic gestures correspond to trajectories described with some particular poses or a combination of poses during a period of time. Our prototype has been designed to recognize seven static and two dynamic gestures with one hand.

For this initial study, we have used a relatively small vocabulary of gestures. Fig. 4 shows images used for one hand “static gestures vocabulary”. The use of geometrical shapes for the markers reduces the complexity of the recognition. Using this representation, it is only possible to recognize between a finger totally extended and a hidden finger due to the lack of individual markers corresponding to each joint of a finger.

To design the dynamic gestures we proposed that a particular sequence of static gestures would be used to start the gesture sequence itself. The trigger sequence is used in our system as a logical “key” indicating the computer program that the user is commanding a dynamic gesture detection. The two dynamic gestures to be recognized are illustrated in Figs.6 and 8.

The trigger sequence used to start “dynamic gesture 1” canbe seen in Fig. 5. In this case we tried to replicate a gesture related to the “double click” of a mouse. By changing the distance between the hand and the camera in the space, the images of the hand will chance in size depending on such distances. A rapid combination of motions down-up-down-up will generate a sequence of frames where the apparent size of the hand (markers) is increased or decreased (Fig. 5(b)). A state machine (Fig. 5(a)) was built to run with every new frame. A timer consisting in a frame counter keeps track of the maximum number of frames allowed to complete the gesture trigger. Once the transition to a “small hand” is detected, the timer (frame counter) keeps the number of frames. Further steps in the state machine can be achieved if new transitions are detected and the counter is not over. If the whole sequence is achieved, a trigger flag is set to be used as dynamic gesture trigger.

Fig. 6 shows several frames used to build a rectangular region of interest. Once the "trigger signal" coming from the state machine is received, we start the acquisition of the dynamic gesture. The gesture proposed in this set of pictures corresponds to the definition of a new region of interest. It can be used also to draw geometrical shapes in the air through the gesture system. In this example the sequence of frames from (a) to (d) is used to limit the corners of a rectangle. A new counter keeping track of the duration of the gesture (in frames) limits the number of frames to be considered as part of the gesture (Gesture 1).

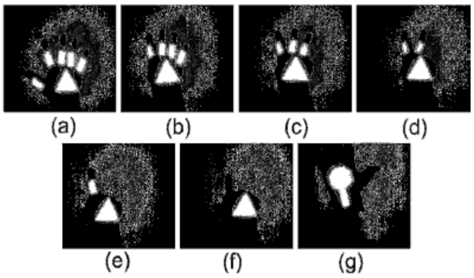

Figure 4. Static Gestures (one hand): From (a) to (e) different numbers of fingers shown. (f) Back of the hand. (g) Palm of the hand.

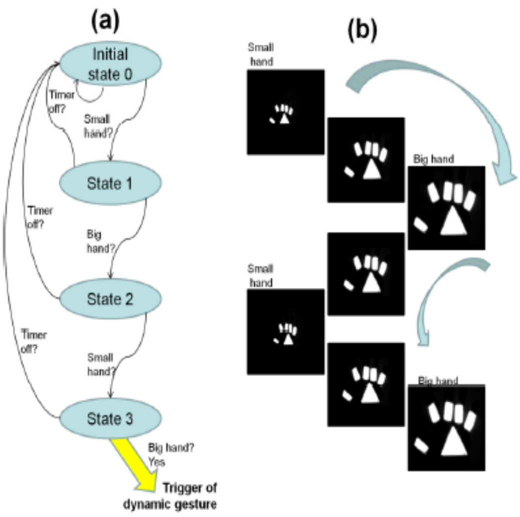

Figure 5. Dynamic gesture trigger 1: State machine (a). Gesture sequence (b).

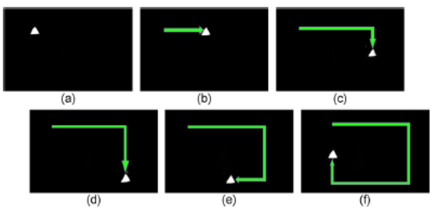

Figure 6. Dynamic gesture 1

A second gesture trigger was implemented. This time replacing the "small-big-small-big hand" sequence with a new sequence formed with "open-close-open-close hand". The same state machine approach used with the former dynamic gesture trigger was used. The correct sequence executed before the timer ends, serves as a gesture trigger for the second dynamic gesture detector. Fig. 7 illustrates the second gesture trigger logic. A sequence of static one-handed gesture is decoded using a state machine (a). In this first event trigger, it is required to detect a sequence composed by "open hands" and "close hands" (b). This sequence once detected will trigger the start of the desired gesture. "Open-close-open-close hand" transition is our representation of a mouse double "click" done in the air with the gesture system.

Once this second type of gesture trigger is on, the event is assembled with the information of the next few frames. A counter is used to keep track of the maximum duration of the gesture. In this second dynamic gesture, the motion of the hand during a number of frames is used to establish the size of a region of interest (ROI) just by pointing 2 diagonal corners (see Fig. 8). Once the "trigger signal" coming from the state machine is received (trigger 2, Fig. 7), we start the acquisition of the dynamic gesture. The gesture proposed in this set of images corresponds to the definition of a new region of interest. It can be used also to define geometric patterns using this set-up. In this example the sequence of frames from (a) to (d) are used to limit the corners just by using 2 diagonal corners.

V. Image Processing and Pattern Identification

A. Preprocessing

Due to the quality of the images obtained we used simple and fast morphology operations applied to the grayscale input image. The first step is to "binarize" the grayscale image to obtain pure black and white values to operate (0 and 255). Additionally, steps of erosion and dilation are applied to obtain clean binary images with just the required elements (markers)

B. Pattern Recognition Algorithm

By observing the simplified set of gestures used in this project (see Fig. 4) is it possible to find common shapes in several of these gestures. In all the gestures where the hand is showing the back side, there is a triangular shape representing the center of the hand(back side of the hand). Configurations from Fig. 4(a) to 4(f) only change in the number of fingers displayed. On the other hand, the palm (Fig. 4(g)) presents a very characteristic shape similar to an octagon.

Our pattern search algorithm is based on this subdivision of patterns that form the hand. A valid hand is uniquely defined as a base pattern surrounded by a number of segments (in the case of fingers extended). Two "base" patterns were used as templates in the detection algorithm.

1) Base template

The two patterns to be used as templates for hand recognition are used to train the system. These two images are filtered with the erosion and binarization process mentioned before. After that step, connected components analysis (blob analysis) is used to find the properties of the contours defined by the shapes in the template images. From these templates we obtained some descriptors to be used to identify the gestures present in the input images.

The algorithm for connected components provides a large list of points, called contours, which enclose each of the blobs. Using this list to compare patterns can be a difficult and inefficient task. We used the polygon approximations algorithm available in OpenCV to deal with this complexity. To simplify the shape analysis process we apply the Polygon Approximation function (cvAproxPoly). With this function it is possible to approximate a contour representing a polygon with another contour having fewer vertices. The implementation of this algorithm in OpenCV acts on a sequence of contours with a given "precision" parameter. The resulting contour (or list of contours) is a reduced list of points that will represent a polygon related to the original blob.

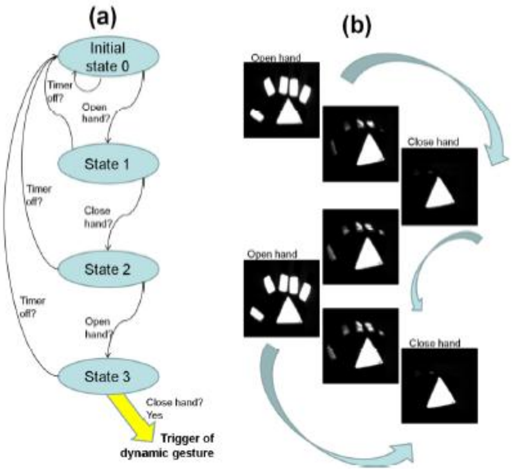

Figure 7. Dynamic gesture trigger 2.

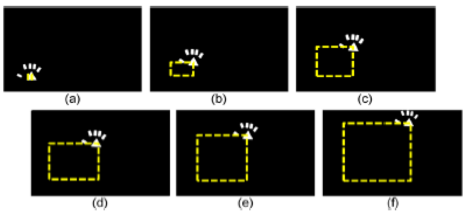

Figure 8. Dynamic gesture 2

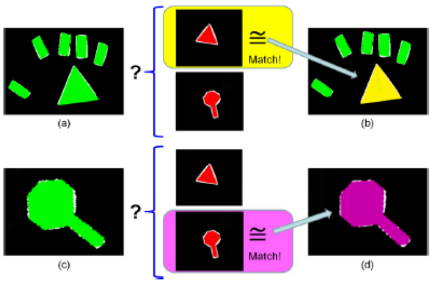

Figure 9. Pattern matching approach

2) New images treatment

The 2 patterns used as training templates represent the 2 classes used to identify any hand. Each hand is built by one of the base patterns and in some cases a number of regions (blobs) around the root pattern representing the extended fingers.

The image is decomposed in separated regions (blobs). These blobs are classified according to the similarity between them and the trained patterns. Fig. 9illustrate the classification process of the blobs in every image. According to this Fig. input images (a) and (c) display in all the independent regions (blobs) obtained after the polygonal approximation. Each of the blobs is compared to the blobs of the template images (template blobs) using Hu moments, convexity (convex hull) and some properties of the blob analysis as perimeter and area of the blob. If there is a match related to the base element of the hand (palm or dorsal view of the hand) the corresponding element is labeled with that identification (seen in yellow in image (b) and purple in image (d)).

We based our matching routine in the use of Hu moments, convex hull and the dimensions of the blobs (just to limit too big or too small regions). The goal of this blob classification process is to identify the similarity of the trained regions and the core elements forming a hand. It is critical for this step to identify the patterns by methods that are invariant respect to rotation, and size. Invariant methods are useful in our case to find a similar patterns with any degree of rotation (hand rotated in any angle around the plane of the image) using a very limited set of training images.

Hu Invariant Moments: Invariant Moments have been frequently used as descriptors for image processing, remote sensing, shape recognition and classification [23], [24]. Moments can provide characteristics of an object that uniquely represent its shape. Invariant shape recognition is performed by classification in the multidimensional moment invariant feature space. Several techniques have been developed to compute invariant features from moments for object recognition. These techniques are distinguished by their moment definition, type of data used and the method for deriving invariant values from the image moments. Hu [24] set out the mathematical foundation for two-dimensional moment invariants and demonstrated their applications to shape recognition. These moments are invariant with respect to translation, scale and rotation of the shape.

In Hu moments, Translation invariance is achieved by computing moments that are normalized with respect to the center of gravity so that the center of mass (central moments). Size invariant moments are derived from algebraic invariants but these can be shown to be the result of size normalizations. From the second and third order values of the normalized central moments a set of seven invariant moments can be computed which are independent of rotation.

There are two functions to compute Hu moments in OpenCV: cvGetHuMoments and cvMatchShapes. We made use of the latter, taking advantage of its unidimensional and compact format. CvMatchShapesfunction can report a measure of the distance in the "Hu invariant moment space" comparing two blobs directly. For the distance measurement we used the contours of the two patterns used for training and the contours obtained from every new image. With this simple procedure we could apply a powerful descriptor to compare and classify every new image captured and to measure the similarity of the blobs of that image and the trained templates.

The Convex Hull: The second criteria we used as a descriptor is the convex hull. We used this property take advantage of the two preselected templates. For the first one, the convex hull is coincident with the contour. The second template on the other hand, presents a big convex deficiency (Fig. 10). OpenCVoffers a function (cvCheckContourConvexity) that verifies the convexity of a contour. In our program we used both Hu moments and hull convexity to determine which blob corresponds to each class (palm or dorsal side of the hand).

3) *Finger detection*

The finger detection required to complete the hand and gesture is based on the criteria of area of the blob. Additionally, a finger to be valid should be located in a region around the base pattern of the hand. Once a base pattern is detected, a circular searching area of interest is established around the hand. If a blob not classified as palm or dorsal side lies inside the limits of blob size and it is located inside the searching area, it will be considered a finger attached to the closer palm. The number of fingers found related to a base pattern (close to the corresponding pattern) determines the configuration of the gesture.

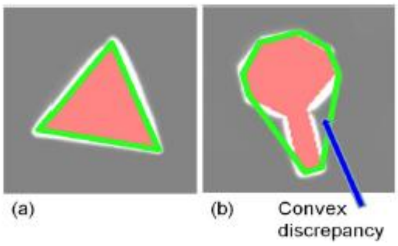

Figure 10. Convex hull discrepancy computation.

4) *Dynamic gesture detection*

At any time the system decodes static poses and records a recent history of gestures (using a state machine) while waiting for the proper combination required to build dynamic commands. The steps mentioned before related to the static gesture detection are fundamental part in the dynamic gesture decoding. Once the input images are analyzed and hand poses are identified it is just required to keep track of the more recent reports of position, pose, size and direction to feed the state machine.

VI. Software Implementation

Fig. 11 shows in pseudo code the main steps of the implementation. The descriptors used to identify each blob were Hu moment comparison, convex hull test and the dimension of the blob.

Custom functions were built:

HANDBUILDER: from a list of disjointed (unrelated) blobs, hand builder will construct a hand from a base element (palm or dorsal maker) and fingers in the vicinity.

POSEANALIZER: Every hand found with the former method is analyzed to compute the information required from every pose, number of fingers, direction of the hand (angle), size, and position on the image plane.

GESTUREANALIZER1: Keeps track of the state machine saving recent history of poses of the hand. In particular this function is related to the gesture trigger based on the change in size of the hand (perceived as a small-big-small-big hand sequence).

GESTUREANALIZER2: Keeps track of the history of poses. It is related to the gesture trigger based on change of hand pose (open-close-open-close hand sequence) to command the definition of a new ROI.

```
int main() {
    LoadImage( "hand_patterns.png" );
    cvThreshold("hand_patterns.png " );
    cvErode("hand_patterns.png");
    cvFindContours("hand_patterns.png");
    cvApproxPoly("handdown_pattern.png");
//FOR PRINCIPAL CYCLE
   for (int image = 0; image < 10000 ; image++){
      cvLoadImage( new_image );
      cvThreshold(new_image );
      cvErode(new_image);
      cvFindContours(new_image);
      cvApproxPoly(new_image );
 // to label the identity of every blob found
      for(  c=first_contour; c!=NULL; c=c->h_next ) {
          if Hu_match(new_image, handup_pattern) and convexHull{
          blob = dorsal_marker
          }else{
     if Hu_match(new_image, handdown_pattern) and  not(convexHull){
        blob = palm_marker
     }else{
        If area<max and area>min {
           blob=finger
        }
     }
// to build hands based of the identity of the blobs and the distance to the hand
Call function HANDBUILDER(blobs)
Call function POSEANALIZER(hands)
Call function GESTUREANALIZER1
Call function GESTUREANALIZER2
// if a dynamic gesture is received, we build a new region of interest.
If dynamic_event_trigger received {
    Draw new ROI
    }
}
```

Figure 11. Pseudocode.

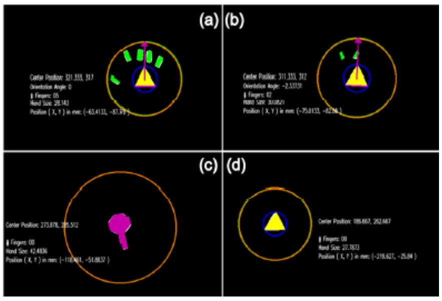

Figure 12. Static gestures detected.

VII. Results

Fig. 12 reports results of the static gesture identification. Every blob in the new image is compared with the training images and classified. After the classification process the hand is assembled. The method used to find the direction of the hand is defined by finding the base of the triangle. The base of the triangle is defined as the 2 vertices farther from the fingers. Dynamic gestures can be started at any time by just following the required trigger sequence. Fig. 13 illustrates the first dynamic gesture proposed. Fig. 13(a) and 13(b) correspond to two of the four states required for the state machine as a key to trigger the gesture sequence. From Fig. 13(c) to 13(h) the motion of the hand around the screen is used to define the four corners of a new region of interest. “Double click” gestures are intuitive due to their daily use to control computers. This version of the “double click” has the goal to test the response of the platform to new symbols for our gesture alphabet. The successful detection of the proposed sequence offers tremendous space to many new “key” combinations with additional command meanings.

By measuring the dimensions of the area the camera observes at ground level we computed the scale conversion factors to transform pixel dimension in the image space in mm in the real world. These scale factors are 1.16mm/pixel in X axis and 1.14mm/pixel in Y axis.

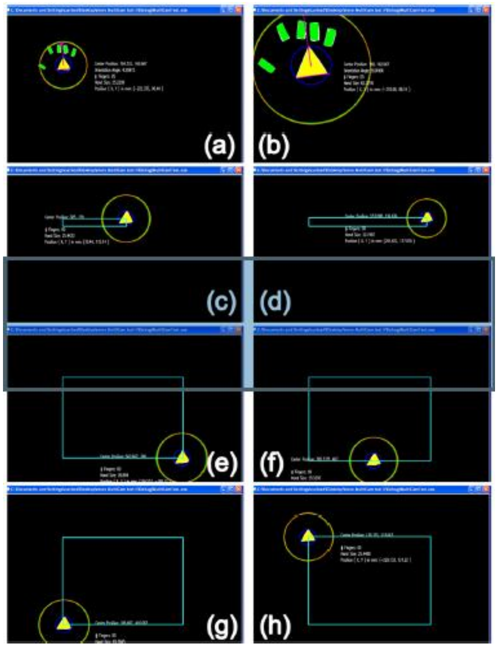

Figure 13. Dynamic gestures in action.

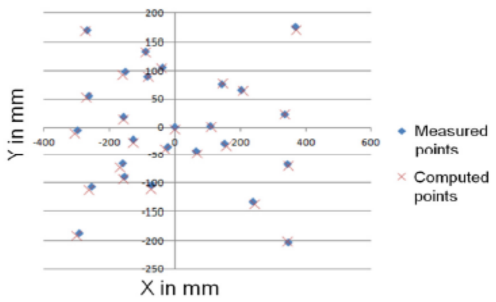

Figure 14. Measured values (mm) vs. computed values.

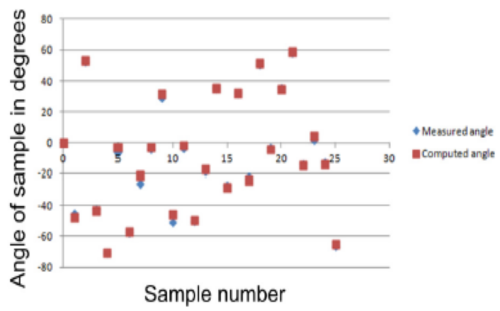

Figure 15. Angle measured vs. angle computed.

The plot in Fig. 14 was obtained by representing the (x,y) coordinates measured with 25 positions of the glove on the table vs. the computed results. It can be seen in the plot a close correspondence between the real world coordinates vs. the computed coordinates. After the conversion scale between the camera and the testing table was found, a set of 25 measures represented in this table were also computed converting the pixel location into mm. The median value of the error (error distance) is 5.99mm, while the maximum error distance is 11.16mm. Measurements were taken at the ground level of the table to simplify the process.

Finally, Fig. 15 presents the contrast between the measured rotation angles of the glove vs. the computed values. Once more it can be seen that the computed angles are consistent with the measured ones and the error is small. The median value of the error (square root of the squared difference) is 1.42 degrees while the maximum error is 4.93 degrees.

VIII. Discussions and Conclusions

The proposed glove/marker approach shows promising results to be used as a user computer interface and suggests the feasibility of using simple markers as a part of an accurate control input modality.

Through our experimental study, it was observed that the selected alphabet of gestures shows a high recognition rate. It provides a simple and intuitive set of basic gestures that can be combined in any order to generate new gestures, increasing the number of commands to be sent to the computer system. One issue of the current marker/glove interface is that the poses of the hand are oversimplified. For example a finger is represented as a single marker segment. There is no representation of the full possible configurations of the human hand. We took advantage of that oversimplification to build simple gestures easy to remember for any new user with little training.

Due to the nature of the triangular shape of the dorsal pattern it is not possible to compute the angle of the hand when no fingers are shown in the image. A similar problem would occur when just the thumb is displayed, forcing the system to compute the angle of the hand with undesired vertices. This situation could improve with the modification of that pattern by adding an additional mark or hole on the current marker.

Another issue is that the plane of the markers should be parallel to the plane of the camera up to some degree. The reflectivity properties of the markers will change with extreme angles, as also the projected shape of the pattern, adding new difficulties to the pro-posed pattern matching approach that should be considered.

One of the strengths of our system is the quality of the image we obtained from the beginning. From the conceptual design it considered that our system should be robust even in the presence of poor lighting or uneven background. As such, the task of segmentation can be considerably reduced due to the proposed illumination/imaging set-up.

References

[1] V. Ayala-Ramirez, S. A. Mota-Gutierrez, U. H. Hernandez-Belmonte, *et al.*, “A hand gesture recognition system

based on geometric features and color information for human computer interaction tasks," *Xalapa, Ver.*, Mexico, 2011.
[2] M. Bhuyan, D. R. Neog, and M. K. Kar, "Fingertip detection for hand pose recognition," *International Journal of Computer Science and Engineering*, vol. 4, no. 3, 2012.
[3] X. Wang, M. Xia, H. Cai, Y. Gao, and C. Cattani, "Hidden-markov-models-based dynamic hand gesture recognition," *Mathematical Problems in Engineering*, 2012.
[4] F. Chen, C. Fu, and C. Huang, "Hand gesture recognition using a real-time tracking method and hidden markov models," *Image Vision Comput*, vol. 21, no. 8, pp. 745-758. 2003.
[5] M. Alsheakhali, A. Skaik, M. Aldahdouh, and M. Alhelou, "Hand gesture recognition system," *Computer Engineering Department, the Islamic University of Gaza*, Gaza Strip, Palestine, 2011.
[6] C. Keskin and L. Akarun, "STARS: Sign tracking and recognition system using input–output HMMs," *Pattern Recog. Lett*, vol. 30, no. 12, pp. 1086-1095, 2009.
[7] H. Lo and S. Payandeh, "Vision-based techniques for real-time human-robot interaction," *Intelligent Autonomous Systems*, 2004, pp. 935 - 942.
[8] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," Presented at ACM Transactions on Graphics (TOG), 2009.
[9] S. Cobos, M. Ferre, M. A. Sanchez-Uran, J. Ortego, and R. Aracil, "Human hand descriptions and gesture recognition for object manipulation," *Comput. Methods Biomech. Biomed. Engin*, vol. 13, no. 3, pp. 305-317, 2010.
[10] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064-1076, 2011.
[11] V. A. Prisacariu and I. Reid, "3D hand tracking for human computer interaction," *Image Vision Comput*, vol. 30, no. 3, pp. 236-250, 2012.
[12] F. Wei, C. Xiang, W. Wen-Hui, Z. Xu, Y. Ji-Hai, V. Lantz, and W. Kong-Qiao, "A method of hand gesture recognition based on multiple sensors," Presented at 4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE), 2010.
[13] N. Jojic, B. Brumitt, B. Meyers, S. Harris, and T. Huang, "Detection and estimation of pointing gestures in dense disparity maps," Presented at Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000.
[14] H. Lahamy and D. Lichti, "Real-time hand gesture recognition using range cameras," Presented at Proceedings of the Canadian Geomatics Conference, Calgary, Canada. 2010.
[15] X. Liu and K. Fujimura, "Hand gesture recognition using depth data," Presented at Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004.
[16] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," Presented at Ro-Man, 2012.
[17] P. P. Bruyant, M. A. Gennert, G. C. Speckert, R. D. Beach, J. D. Morgenstern, *et al.*, "A robust visual tracking system for patient motion detection in SPECT: Hardware solutions," *IEEE Transactions on Nuclear Science*, vol. 52, no. 5, pp. 1288-1294. 2005.
[18] V. Frati and D. Prattichizzo, "Using kinect for hand tracking and rendering in wearable haptics," Presented at World Haptics Conference, IEEE, 2011.
[19] J. Davis and M. Shah, "Recognizing hand gestures," in *Computer Vision—ECCV'94*, Anonymous, 1994.
[20] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler, "A real-time hand gesture interface for medical visualization applications," in *Applications of Soft Computing*, Anonymous, 2006.
[21] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 8, pp. 1141-1158, 2009.
[22] P. Chakraborty, P. Sarawgi, A. Mehrotra, G. Agarwal, and R. Pradhan, "Hand gesture recognition: A comparative study," Presented at International MultiConference of Engineers and Computer Scientists, 2008,
[23] M. R. Teague, "Image analysis via the general theory of moments*,"*Josa*, vol. 70, no. 8, pp. 920-930, 1980.
[24] M. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory,* vol. 8, no. 2, pp. 179-187, 1962.

C. Diaz received his B.Sc. in Electronics Engineering from Universidad Simon Bolivar, Caracas, Venezuela in 1996. He is currently a M.A.Sc candidate in the Engineering Science program in Simon Fraser University in Burnaby, Canada.
He co-founded Modular Vision Systems in 1996 (Caracas, Venezuela), a companydedicated to develop quality control inspection machines for the packaging industry based on machine vision technologies. He was responsible for both the Research and Development (R&D) and the Project Engineering Departments for 15 years.
He is in the Experimental Robotics Laboratory (ERL) in Simon Fraser University in Burnaby, Canada under the supervision of Dr. Payandeh. His research interests include Robot vision, Image processing and Haptic computer interfaces.

S. Payandeh received his B.Sc. from the University of Akron, United States, 1981 and hisM.A.Sc. and Ph.D. from University of Toronto, Canada, (1986 and 1990). He is an IEEE member. He has been professor at the School of Engineering Science at Simon Fraser University in British Columbia, Canada since 1991. His main area of research is in robotics and in particular in interaction modeling and coordination of networked, cooperative dynamical agents. He has more than 250 technical publications in journals and conferences. He holds 6 US patents in the field of haptic user interfaces, robotics devices and haptic rendering. He also coauthored the one of the first books in the area of medical robotics and holds one of the first patents in this area.
Dr. Payandehis the group leader of the Experimental Robotics Laboratory in Simon Fraser University in Burnaby, Canada. His research interests are on Robotics, Multi-view tracking, Haptic devices and haptic rendering.

