Fast Hand Gesture Recognition based on Saliency Maps: An Application to Interactive Robotic Marionette Playing

M. Ajallooeian, A. Borji, B. N. Araabi , M. Nili Ahmadabadi, H. Moradi

Abstract— **In this paper, we propose a fast algorithm for gesture recognition based on the saliency maps of visual attention. A tuned saliency-based model of visual attention is used to find potential hand regions in video frames. To obtain the overall movement of the hand, saliency maps of the differences of consecutive video frames are overlaid. An improved Characteristic Loci feature extraction method is introduced and used to code obtained hand movement. Finally, the extracted feature vector is used for training SVMs to classify the gestures. The proposed method along a hand-eye coordination model is used to play a robotic marionette and an approval/rejection phase is used to interactively correct the robotic marionette's behavior.**

I. Introduction

HUMAN ROBOT INTERACTION (HRI) considers the interactions between people and robots. The HRI problem can be considered from several points of view such as verbal communication [1], remote operation [2], and programming by demonstrations [3]. One of the main aspects of HRI is the channel of interaction which can take place through different means like voice commands, facial expressions, or hand gestures. Hand gestures are common and practical tools to convey information to a robot. Different methods for hand gesture recognition has been implemented and used in HRI [4].

Gesture recognition techniques should be fast in order to be suitable to be used in HRI applications. So it would be beneficial to use fast image processing techniques like visual attention. Visual attention processes image regions in parallel and selects some spatial regions for further serial processing. It is inspired from human visual behavior and has proved to be useful in applications like scene interpretation, object recognition, visual robot navigation and localization. In this work saliency maps of visual attention are utilized for fast recognition of hand gestures.

The proposed gesture recognition method takes advantage of the saliency map of visual attention and extracts the important parts of differences of consecutive video frames, which are potential hand regions. Then, these regions in video frames are overlaid to obtain the overall hand movement.

An appropriate feature extraction method is needed to recognize a gesture. Since we are going to develop a fast method for gesture recognition, the feature extraction method itself has to be computationally efficient. For this, we implemented an improved version of the '*Characteristic Loci*' feature for coding the obtained hand movement. Extracted features from a gesture dataset are then used to train Support Vector Machines (SVM). Trained SVMs are later used to recognize the concepts of shown gestures.

We used the proposed hand gesture recognition method in an interactive robotic marionette playing task. A user performs a gesture in front of a camera and then the concept of the performed gesture is recognized. If the recognized gesture is new for the robotic marionette, a prototype for the performed gesture is developed using a hand-eye coordination model. Otherwise, the previously learnt prototype is performed. Also an approval/rejection phase is implemented to help the robotic marionette learn the correct prototype.

The rest of this paper is organized as follows: related works are reviewed in section II. Section III gives an overview on the saliency-based model of visual attention. The hand gesture recognition method is explained in section IV. The process of implementation of the proposed hand gesture recognition on the robotic marionette and the hand-eye coordination model deployed are described in section V. Finally, the paper is concluded in section VI and the future works are discussed.

II. Previous Works

A hand gesture recognition and interaction system has two phases of detecting hands and recognition of performed actions or hand gestures. In this section we will review the previous works done in the field of hand detection and gesture recognition.

A. Hand Detection

Many different approaches for hand detection using different kinds of visual cues have been proposed. A straightforward approach is to look for skin-colored regions in the image [5-7]. Skin color classification is difficult to

Manuscript received March 15, 2009.

Mostafa Ajallooeian is with the Robolab, ECE dept. Faculty of Engineering, University of Tehran, Iran; phone: +98-912-219-29-11; e-mail: ajal@ut.ac.ir (corresponding author)

Ali Borji is with the School of Cognitive Science, IPM, Tehran, Iran; e-mail: borji@ipm.ir

Majid Nili Ahmadabadi is with the Robolab, ECE dept. Faculty of Engineering, University of Tehran, Iran, and School of Cognitive Science, IPM; e-mail: mnili@ut.ac.ir

Babak N. Araabi is with the Robolab, ECE dept. Faculty of Engineering, University of Tehran, Iran, and School of Cognitive Science, IPM; e-mail: araabi@ut.ac.ir

Hadi Moradi is with the Robolab, ECE dept. Faculty of Engineering, University of Tehran, Iran; e-mail: moradih@ut.ac.ir

978-1-4244-5081-7/09/$26.00 ©2009 IEEE

handle because color is directly influenced by the lighting conditions of the scene and is also dependent on the image acquisition hardware. Thus, a color model that works fine for a given scenario may fail when conditions change. In [8], it is shown that separating the color value from its brightness, helps to overcome some of these problems but it does not solve the general problem.

Some works [9], [10], have incorporated model adaptation techniques for hand detection where model parameters are tuned automatically to the current lighting conditions. To this purpose, some kind of knowledge of the scene or the current lighting must be inferred from the images.

Other studies use structural information of an object like shape, boundaries or general appearance for its detection. Well-known examples are the appearance-based object detector of Viola and Jones [11] or Cootes' and Taylor's active appearance models [12].

Since it is important for an HRI system to percept and act in real time, fast detection has higher weigh than detection accuracy. Optimum case is both fast and high detection rate. While above methods are tuned for hand detection like object detection methods in computer vision literature, here we employ visual attention for fast detection of hand regions. This makes our approach to sound biologically inspired since attention is a key feature in human-robot interaction.

B. *Gesture Recognition*

View-independence and user-independence are two fundamental requirements for hand posture recognition during natural human-robot interaction. In [13], a method is proposed based on Zernike moments and hierarchical classification to handle these simultaneously.

In [14], a glove-based approach is proposed for mobile robot control with hand gestures. Their system is capable of spotting and recognition of six hand gestures using Hidden Markov Models (HMM).

In [15], a gesture interface is developed for a mobile robot equipped with a manipulator. The interface uses a camera to track a person and recognize gestures involving hand motion. A fast, adaptive tracking algorithm enables the robot to track and follow a person reliably through the environments with changing lighting conditions. Results are reported in the context of an interactive clean-up task, where a person guides the robot to specific locations that need to be cleaned and instructs the robot to pick up trash.

III. Saliency-based Model of Visual Attention

Visual Attention is the process of selecting and gating a subset of image regions based on their bottom-up saliency and on the prior knowledge about the scene (top-down) [16], [17]. In brain, bottom-up visual attention is solely determined by basic physical characteristics of visual scene, like luminance contrast, color contrast, orientation and motion. On the other hand, top-down visual attention is influenced by the task demands, emotions, expectations, etc, which mainly come from higher cognitive brain areas like prefrontal cortex, LIP, etc [18].

Saliency-based model of visual attention is a biologically inspired bottom-up model proposed by Itti et al [19], [20] that is an extension and implementation of a previous model by Koch & Ullman [21]. In this model, the input image I is alternatively filtered and sub-sampled to form a Gaussian pyramid [22]. Each pyramid level is decomposed into channels for red(R), green(G), blue(B), yellow(Y), intensity(I), and local orientation(O_θ). If r, g, and b are the red, green, and blue values of the color image, normalized by the image intensity I, then:

$$R = r - (g+b)/2), G = g - (r+b)/2, \\ B = b - (r+g)/2, Y = r + g - 2(|r-g| + b) \quad (1)$$

Local orientations (O_θ) are obtained by applying steerable filters to the images in the intensity pyramid I [23]. From these channels, center-surround ‘‘feature maps’’ are constructed and normalized:

$$\mathcal{F}_{I,c,s} = \mathcal{N}(|\mathcal{M}_I(c) \ominus \mathcal{M}_I(s)|), I = \{I\} \quad (2) \\ \mathcal{F}_{C,c,s} = \mathcal{N}(|\mathcal{M}_C(c) \ominus \mathcal{M}_C(s)|), C = \{RG, BY \\ \mathcal{F}_{O,c,s} = \mathcal{N}(|\mathcal{M}_O(c) \ominus \mathcal{M}_O(s)|), O = \{0°, 45°, 90°, 135°\}$$

Here, $\ominus$ denotes the across-scale difference between two maps at the center (c) and the surround (s) levels of the respective feature pyramids. I, C and O are intensity, color and orientation features respectively. $\mathcal{N}$ is an iterative, nonlinear normalization operator [24]. The feature maps are summed over the center-surround combinations using across-scale addition, and the sums are normalized again.

$$\mathcal{F}_l = \mathcal{N}\left(\sum_{c=2}^{4} \sum_{s=c+2}^{c+4} \mathcal{F}_{l,c,s}\right), l \in \{I, C, O\} \quad (3)$$

For the general features color and orientation, contributions of the feature dimensions are linearly summed and normalized once more to yield “conspicuity” maps.

$$C_I = \mathcal{F}_I, C_C = \mathcal{N}(\textstyle\sum_{l \in C} \mathcal{F}_l), C_O = \mathcal{N}(\textstyle\sum_{l \in O} \mathcal{F}_l) \quad (4)$$

Finally, all conspicuity maps are combined into one saliency map:

$$S = \frac{1}{3} \sum_{k \in \{I,C,O\}} C_k \quad (5)$$

Maximums of the final saliency map are the focuses of attention. In the unbiased condition, saliency model of visual attention selects regions of the image which are different from their surroundings. An advantage of this model is that it could be tuned to select specific objects like hands, faces, etc by weighting feature channels. Fig. 1 shows the operation of the above model over a natural image for the purpose of hand detection. In the next section, we will show how we used this model for fast detection of hands in video frames. Saliency model detects some false positive hand regions; therefore hand motion information helps to discard these regions. It is important that the time complexity of the saliency-based model of visual attention presented in [19] is of $O(n^2)$ (derived from [19]).

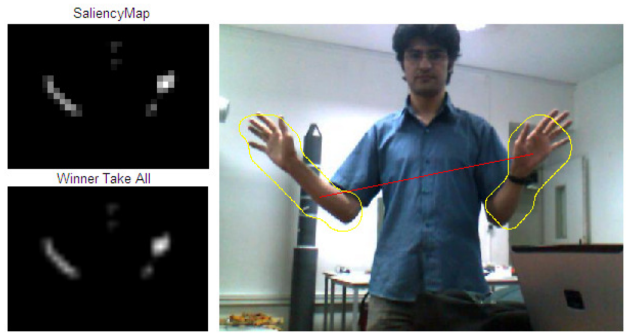

Fig. 1. Using bottom-up model of visual attention for hand detection. Upper left: resulted saliency map, Lower left: winner take all network output. Right: output hand areas.

IV. Fast Gesture Recognition

A Scenario of the problem we aim to solve is as follows:

1) User moves his/her hand and draws a conceptual sign (gesture) in front of the camera;
2) Hand movement path is then extracted from the video taken;
3) Discriminant features for classification from the obtained path are derived;
4) Finally, conceptual meaning of user's gesture is determined by classifying the feature vector.

Our proposed method for this task consists of three phases (1) hand movement extraction (2) feature extraction from hand movement path and (3) classification. In the following subsections we explain these phases in detail. Fig. 2 shows the overall process of hand gesture recognition.

A. Hand Movement Extraction

Fig. 3 shows the process of hand movement extraction. The input is a video containing an action. A series of image processing and saliency operations are performed over video frames in order to extract the movement. The processes involved are described in the following subsections.

Movement Map: This map illustrates changing regions between two successive frames. For this purpose first, two video frames (F1 and F2) are transformed to gray level images and then a softXOR operation ($\oplus$), XOR considering a threshold for activation value, is applied to them:

$$activationMap = F_1 \oplus F_2 \qquad (6)$$
$$movementMap = activationMap \odot F_2$$

where *activationMap* determines the moving regions between frames. The *movementMap* is calculated by element wise product ($\odot$) of the activation map and current frame. Also *softXOR* operator is defined in equation 7.

$$\forall\, pixel\ p:$$
$$F_1 \oplus F_2(p) = \begin{cases} 1, if\ |F_1(i,j) - F_2(i,j)| > \theta \\ \quad 0, otherwise \end{cases} \qquad (7)$$

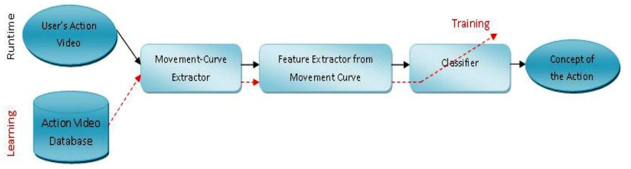

Fig. 2. Overall process of gesture recognition. System is trained over a database of action videos (red dashed arrows). Solid black arrows determine the recall process.

In above formula, θ is the threshold value. Following points must be noted when using this block. First, the threshold value should be small enough in order to reduce true-negative errors and preserve hand regions in the frames. Second, for the first frames when the movement is not actually started and the last frames when movement is already finished, this block could be inhibited.

Saliency Map: The bottom-up attention model discussed in section II is used for hand detection. In [25], the authors have tuned the saliency model for the purpose of human skin detection using a huge train set of human skins. They have used HSV color system for weighting feature channels of the model. In this study, we used the same parameter values for extracting skin regions in video frames.

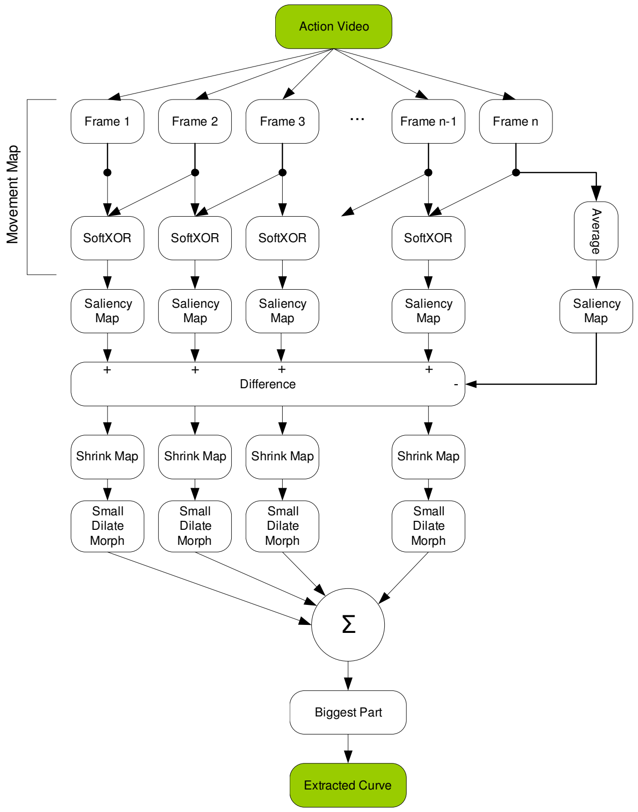

Fig. 3. Hand movement extraction. Video frames are averaged and the saliency map of the average frame is calculated. In parallel, for successive frames, movements between frames are estimated and their saliency maps are computed. Difference between these saliency maps and the average saliency map gives potential hand areas in the frames. These areas are then altered by morphological operations to show the regions of interest. Finally, summation of these small regions leads to the overall hand movement path.

The Overall Process: Following pseudocode summarizes the process of hand movement detection. The time complexity of the overall process is also analyzed, with L as the total number of the video frames, and assuming that each video frame is an $n{\times}n$ image.

```
1- Calculate the average frame of the action movie, $O(L \times n^2)$
2- Generate the saliency map for the average frame called average saliency map, $O(n^2)$
3- For each movie frame (total complexity: $O(L \times n^2)$)
    3-1 Calculate the movement map of this frame against its previous frame, $O(n^2)$
    3-2 Generate the saliency map for movement map, $O(n^2)$
    3-3 Subtract the average saliency map from 3-2 result and set the negative values to zero , $O(n^2)$
    3-4 Iteratively shrink the saliency map until it remains unchanged, $O(n^2)$ (shrink morphological operation is of $O(n^2)$, derived from [26], and it is repeated for a couple of times, $k$, that is a constant)
    3-5 Dilate the saliency map with a square structure element of width 3 or 4, $O(n^2)$ (dilate morphological operation is of $O(n^2)$)
4- overlay the outputs of step 3 for all frames, $O(n^2)$
5- Set the positive and negative values of step 4 output to one and zero respectively, $O(n^2)$
6- Calculate connected components of the generated binary image, $O(n^2)$ (derived from [26])
7- Output the largest connected component
* Total time complexity: $O(L \times n^2)$
```

Investigating the above algorithm we could come to the following considerations. Because throughout the video, hand moves a lot more than other components like face, body, etc, using the average saliency map is effective. Average saliency map contains all body components except the hand palm that have moved constantly. Therefore by subtracting it from frames, saliency maps increase the saliency probability of the hand in the frames. In order to represent a salient region with a point, we used shrink morphological operation. This filter transforms a continuous region into a single point, and a region with a hole inside it into a narrow ring around the hole.

To find the overall hand movement, we used the union of all the shrunk regions. An alternative method for this purpose might be to detect the location of hand in each frame and track these locations. Using tracked locations, hand movement pattern could be detected. However, this method leads to some difficulties. First, detection of hand in each frame is inaccurate which causes the overall path to be very spiky. Even if it would be possible to detect hand with high accuracy, much computation is needed. Furthermore, in frames where hand has fast movements, instead of hand structure we have big faded color patches. This makes some problems for methods that use the hand structure for detection purpose. Fortunately, saliency model bypasses this problem by a low order complexity operation ($O(L \times n^2)$) that only processes hand color information.

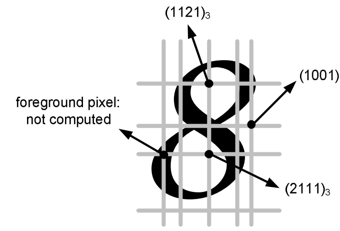

Fig 4. Characteristic Loci feature extraction for character '8'. Directions are up, right, down, left respectively and *max* is equal to 2. Characteristic Loci feature is not computed for foreground pixels.

B. Improved Characteristic Loci

After the hand movements are extracted from the action videos, the resultant movement paths should be recognized. Investigating the hand movements of different persons in different trials, we noticed that hand movement paths are just similar in overall structural pattern and are different in details and also have different scales. Therefore in order to recognize the paths, invariant features to both scaling and distortion are needed.

There are many feature extraction techniques which code the structure of the shapes [26]. As these methods are based on the structural features like identifying holes or corners, they are sensitive to structural noises. They are also expensive in terms of computational complexity. To remedy these problems we used a feature extraction technique based on a statistical-structural feature called Characteristic Loci [27]. The original Characteristic Loci feature extraction method is enhanced in this study in terms of computational complexity.

Loci Characteristic features are useful when dealing with binary images. A well known application of this feature is optical character recognition (OCR) [28]. This feature is calculated based on the position of the foreground points against each other. For each background pixel, we move in different directions and count the number of hits (intersections) with the foreground continuous blocks. For all successive foreground pixels, it is only counted once. In practice, a maximum value (*max*) is assumed for each direction and hit values above are rounded to *max*. Therefore for each pixel in each direction a value between 0 and max is derived. Four (up, down, left, right) or eight (up, up right, right ….left, up left) directions are usually considered. Moving in four directions, a number like this could be derived for a background pixel in base *max+1*:

$$(abcd)_{max+1} \tag{8}$$

where *a* is the hit value for the first direction, *b* for the next one and so on. An example of feature extraction using loci feature for character '8' is illustrated in Fig. 4.

After the values of the background pixels are computed, histogram of these values is calculated. It shows in how many background points a certain value has occurred. In above example, all points in the upper hole of '8' have the same $(1121)_3$ value. If we set *max* value to 1, a value of $(1111)_2$ is calculated. Finally, histogram values are divided by the number of background (white) pixels to become invariant to scaling.

Traditional method for Characteristic Loci feature extraction has a complexity of order $O(n^3)$, where n is the width of an image, assuming a squared image. Using this method for binary images is to some extent satisfactory. However to make our method faster, we have devised a new method for calculation of Characteristic Loci features based on dynamic programming. This new method has the $O(n^2)$ order of complexity.

The traditional algorithm for extraction of Characteristic Loci features for an image is as follows:

```
for each foreground pixel $p$
    for each direction $d$
        Trace all the pixels in direction $d$ and count number
        of intersections with the image foreground blocks.
    end for
end for
```

In above algorithm, a foreground block is a consecutive sequence of foreground pixels bounded by background pixels. Analyzing the above pseudo code, it is obvious that for each image pixel p, n-1 pixels in each direction and its opposite direction must be evaluated. For an $n \times n$ image above algorithm takes $\frac{1}{2}(n^2 - f)(n-1)k$ operations, where k is the number of directions and f is the number of foreground pixels. Therefore, it is of the computational complexity order of $O(n^3)$.

It can be seen that for each pixel p in direction d, it is not necessary to trace remaining n-1 pixels in direction d and its opposite direction. It's due to the fact that value of the pixel p is a function of its own value and value of the next pixel in that direction. So to calculate value of the pixel p, it is just enough to move in the opposite direction and use the dynamic programming to calculate of this pixel value. According to the above discussion our modified algorithm is as follows:

```
for each direction $d$
    for each row $r$
        for each pixel $p$ in row $r$
            If the previous pixel $q$ in direction orthogonal to
            direction $d$, was a foreground pixel and pixel $p$ is
            background, then set the value of pixel $p$ equal to
            the value of pixel $q$ plus one. Otherwise set the
            value of pixel p as the value of pixel $q$.
        end for
    end for
end for
```

In the above algorithm, each pixel is visited once, so it takes n^2k operations and thus algorithm is of order of $O(n^2)$. Such enhancement reduces the execution time of the feature extraction unit significantly especially for large images. For instance, for a 100×100 squared image and $k=4$ and $w=3{,}000$, original algorithm takes 1,386,000 key operations where for our improved algorithm, the number of key operations are 40,000 which leads to a 34.6 times faster computation time.

C. *Classification*

For each sign class an SVM classifier is constructed. To train each single SVM, samples in the corresponding sign class are labeled *'1'* and all data from other classes are labeled *'0'*. We experimented with linear kernel SVM classifier. To classify a test sample, it is fed to all trained SVMs and the one with the highest confidence determines the output class. Confidence of each SVM for an input is the distance of the sample from the discriminant line of the SVM into the acceptance area.

V. Experiments

A. *Hand Gesture Recognition*

Four people were asked to draw seven signs by moving their hands in the air. Signs include heart, rectangle, infinity, circle, tick, arc and the eight. Each person repeated every sign for about three to five times. Recording was in the Mobile Robot Lab at the University of Tehran and the subjects were M.Sc. and PhD students. The starting and ending points of gestures in the recorded videos are determined manually. To keep the environment as natural as possible, we did not use any artificial backgrounds or other simplifications. Subjects had to keep their moving hand in the view field of the camera. No one was allowed to be or move in the background except subject himself. Samples of the extracted hand movements are illustrated in Fig. 5. Some video frames of the movement for the infinity sign of a subject and the overall process are shown in Fig. 6.

Table 1 shows the classification results using loci features over test dataset. Results for each row are averaged over 10 different training sessions.

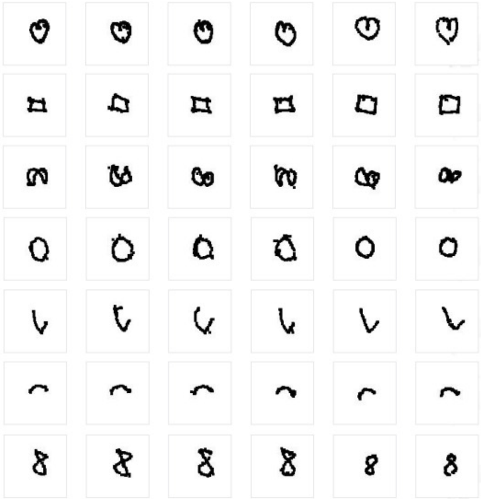

Fig. 5. Sample extracted movement paths by the method. Each column is for one subject.

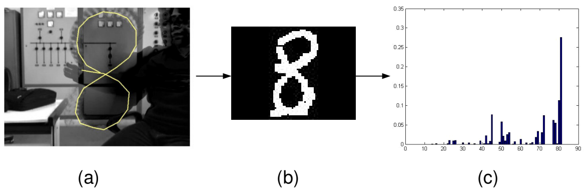

Fig. 6. The overall process. (a) user action; (b) extracted hand movement; (c) feature vector

TABLE I
GESTURE RECOGNITION RESULTS

Direction Vector	*max* counting	Correct Classification Rate% (CCR)
[1 0 1 0 1 0 1 0]	1	77.1
[0 1 0 1 0 1 0 1]	1	85.7
[1 1 1 1 1 1 1 1]	1	91.4
[1 0 1 0 1 0 1 0]	2	88.5
[0 1 0 1 0 1 0 1]	2	91.4

Direction vector is the vectors used in the Characteristic Loci feature. The directions are up, up-right, right, down-right, down, down-left, left and up-left respectively. A '1' for each of these directions means that the corresponding direction is used.

B. *Interactive Robotic Marionette Playing*

The Robotic Marionette. The proposed hand gesture recognition technique is used to play a robotic marionette. The robotic marionette used is named Hootan. Hootan is a 10-DoF robotic marionette controlled by 8 servo motors that pull the attached strings (Fig. 7). The scenario for marionette playing is as follows:

1. A user performs one of the defined gestures in front of the camera;
2. The gesture is recognized;
3. If the gesture is new for Hootan, then the extracted hand movement is used to train Hootan to build a prototype for that gesture. Otherwise, the learnt prototype for the shown gesture is performed.

In some HRI applications where the used robot structure is complex, the relation between what is done and what is seen is unknown (the inverse kinematics of the robot is unidentified). As mentioned in step 3 of the followed scenario, Hootan tries to build a prototype for the gesture shown. This means that the desired motor trajectory is needed to be identified for the corresponding gesture. For this reason, a hand-eye coordination model is used to discover the appropriate action that generates the desired gesture.

Hand-eye coordination. The hand-eye coordination model usde is presented in [30]. This hand-eye coordination model is applied to identify the appropriate action that leads to the desired visual outcome for an imitator robot. The idea of the hand-eye coordination model is to shake the imitator body to reach temporary goals set on the desired visual trajectory. The temporary goals are continually updated in order to reach and sweep the desired trajectory in the visual space (Fig 8). The gathered data during this process is used to train a nonlinear function approximation tool to learn the inverse kinematics of the imitator on the desired trajectory.

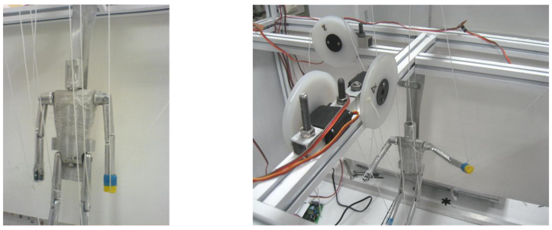

Fig. 7. Robotic marionette Hootan. Hootan is a 10-DoF marionette robot that is controlled via 8 servo motors; two for each arm and two for each foot.

This hand-eye coordination model is used for Hootan to identify the desired motor action that leads to the corresponding gesture.

Interactive Playing. An interactive playing scheme is used to play Hootan. When a gesture is performed for Hootan, there might be two different responses: the gesture is either new or not. When the shown gesture is new, the motor action that leads to that gesture is learnt through hand-eye coordination model discussed. Since no time series is defined for the movement path extracted from user performances, the learnt motor action could be executed in a number of different cases. An approval/rejection process is used to select the desired execution. The tick and wave actions are used as the approval and rejection signs respectively. So the total process of interactive playing is as follows (see Fig. 9 for an example):

1. User- perform a gesture.
2. Hootan- recognize the gesture;
3. Hootan- If the observed gesture isn't a new one, perform the learnt prototype, but if the gesture is new, identify the appropriate action using the hand-eye coordination model;
4. Hootan- perform a sequence for the learnt action;
5. User- If performed sequence is the desired one, perform the tick gesture, else perform the wave gesture;
6. Hootan- recognize the user's hand gesture;
7. Hootan- if it is an approval sign, save the performed action as the correct prototype for the observed gesture, else goto 3.

VI. CONCLUSION

In this paper we presented a fast gesture recognition method. The proposed method is built upon the saliency-based model of visual attention. The input to the proposed method is a video stream that contains a gesture. Then the movement map is calculated from successive frames. Applying the tuned saliency map for hand detection will lead to potential hand regions. These potential hand regions are then overlapped to obtain the overall hand movement. An improved version of Characteristic Loci feature is presented to code the hand movement. Finally, extracted feature is fed to SVM classifiers to recognize the shown gesture. The proposed method is used in an interactive robotic marionette playing task.

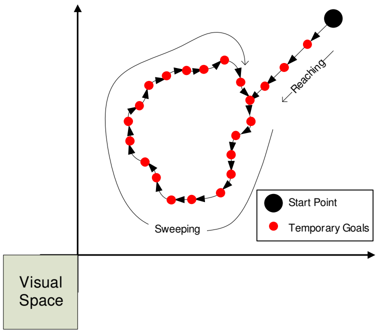

Fig. 8. Hand-eye coordination process. Starting from an initial point, the joints are randomly shaken to reach temporary goals. Temporary goals are set to reach and sweep the desired path.

A similar work is done in [29] where a mentor draws a sign and the performed action is stored by motion capture device. Then the imitator tries to recognize and reproduce the action. Comparing the results with a similar study [29], following points could be realized: 1) In [29] authors have used five gesture classes while in this study seven classes are used. Also we recognized the movements of four people while they only worked with one subject. 2) Capturing the hand movements by a camera is handy and less expensive compared with the special motion captured setup in [29]. 3) Both methods have near the same action recognition rates about 92%. Therefore, this system could be extended to be used in real world situations when a user is to interact with a robot to transfer him some concepts, commands, etc.

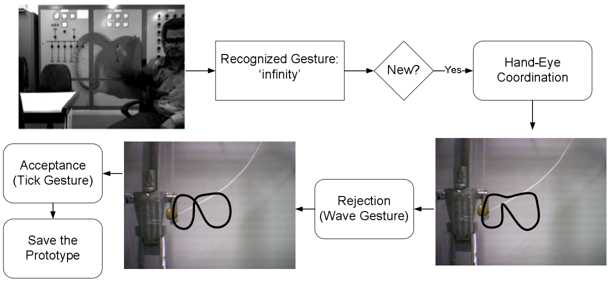

Fig. 9. An example of the interactive playing. User performs a gesture and the gesture is recognized. Then a hand-eye coordination model finds the actions that lead to the observed gesture. A sequence is executed to perform the desired gesture, but it is rejected. Finally another sequence is performed and the outcome is accepted. (Hootan's performances are smoothed so the differences could be seen)

References

[1] Prasad R., Saruwatari H., and Shikano K., "Robots that can hear, understand and talk," *Advanced Robotics*, vol. 18, Jun. 2004, pp. 533-564.

[2] S. Nishio, H. Ishiguro, and N. Hagita, "Geminoid: Teleoperated Android of an Existing Person."

[3] A. Billard and R. Dillmann, "Social mechanisms of robot programming by demonstration," *Robotics and Autonomous Systems*, vol. 54, 2006, pp. 351-352.

[4] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and autonomous systems*, vol. 42, 2003, pp. 143-166.

[5] N. Hofemann, J. Fritsch, and G. Sagerer, "Recognition of deictic gestures with context," *Lecture Notes in Computer Science*, 2004, pp. 334-341.

[6] R. Lockton and A. Fitzgibbon, "Real-time gesture recognition using deterministic boosting," 2002.

[7] J. Triesch and C. von der Malsburg, "Classification of hand postures against complex backgrounds using elastic graph matching," *Image and Vision Computing*, vol. 20, 2002, pp. 937-943.

[8] M.C. Shin, K.I. Chang, and L.V. Tsap, "Does Colorspace Transformation Make Any Difference on Skin Detection?."

[9] F. Dadgostar and A. Sarrafzadeh, "An adaptive real-time skin detector based on Hue thresholding: A comparison on two motion tracking methods," *Pattern Recognition Letters*, vol. 27, 2006, pp. 1342-1352.

[10] L. Sigal, S. Sclaroff, and V. Athitsos, "Skin color-based video segmentation under time-varying illumination.," *IEEE Trans Pattern Anal Mach Intell*, vol. 26, 2004, pp. 862-77.

[11] P. Viola, "Rapid Object Detection using a Boosted Cascade of Simple Features," *cvpr*, IEEE Computer Society, 2001, p. 511.

[12] T.F. Cootes and C.J. Taylor, "Statistical models of appearance for medical image analysis and computer vision," *Proc. SPIE Medical Imaging*, 2001, pp. 236-248.

[13] L. Gu and J. Su, "Natural hand posture recognition based on Zernike moments and hierarchical classifier," *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, 2008.

[14] S. Iba, J.M.V. Weghe, C.J.J. Paredis, and P.K. Khosla, "An architecture for gesture-based control of mobile robots," *1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999. IROS'99. Proceedings*, 1999.

[15] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, 2000, pp. 151-173.

[16] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Annual review of neuroscience*, vol. 18, 1995, pp. 193-222.

[17] C.E. Connor, H.E. Egeth, and S. Yantis, "Visual attention: bottom-up versus top-down," *Current Biology*, vol. 14, 2004, pp. 850-852.

[18] M. Corbetta and G.L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature Reviews Neuroscience*, vol. 3, 2002, pp. 201-215.

[19] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, 1998, pp. 1254-1259.

[20] L. Itti, "Models of bottom-up attention and saliency," *Neurobiology of Attention*, vol. 582, 2005.

[21] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human neurobiology*, vol. 4, 1985, pp. 219-227.

[22] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *Communications, IEEE Transactions on [legacy, pre-1988]*, vol. 31, 1983, pp. 532-540.

[23] E.P. Simoncelli and W.T. Freeman, "The steerable pyramid: a flexible architecture for multi-scalederivative computation," *Image Processing, Proceedings., International Conference on*, 1995.

[24] L. Itti and C. Koch, "Feature combination strategies for saliency-based visual attention systems," *Journal of Electronic Imaging*, vol. 10, 2001, p. 161.

[25] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, 2006, pp. 1395-1407.

[26] R.C. Gonzalez and R.E. Woods, *Digital image processing*, Prentice Hall, 2007.

[27] H.A. Glucksman, "Multicategory classification of patterns represented by high-order vectors of multilevel measurements," *IEEE Transactions on Computers*, vol. 100, 1971, pp. 1593-1598.

[28] A. Ebrahimi and E. Kabir, "A pictorial dictionary for printed Farsi subwords," *Pattern Recognition Letters*, vol. 29, 2008, pp. 656-663.

[29] S. Calinon and A.G. Billard, "What is the teacher's role in robot programming by demonstration? Toward benchmarks for improved learning," *Interaction Studies*, vol. 8, 2007, pp. 441-464.

[30] M. Ajallooeian, M. Nili A., B. N. Araabi, H. Moradi, "An imitation model based on Central Pattern Generator with application in robotic marionette behavior learning", *to appear in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, IROS2009.*