Real-Time Hand Gesture Recognition for Human Robot Interaction

Mauricio Correa[1,2], Javier Ruiz-del-Solar[1,2], Rodrigo Verschae[1], Jong Lee-Ferng[1], and Nelson Castillo[1]

[1] Department of Electrical Engineering, Universidad de Chile
[2] Center for Mining Technology, Universidad de Chile
{jruizd,rverscha,macorrea,jolee}@ing.uchile.cl

Abstract. In this article a hand gesture recognition system that allows interacting with a service robot, in dynamic environments and in real-time, is proposed. The system detects hands and static gestures using cascade of boosted classifiers, and recognize dynamic gestures by computing temporal statistics of the hand's positions and velocities, and classifying these features using a Bayes classifier. The main novelty of the proposed approach is the use of context information to adapt continuously the skin model used in the detection of hand candidates, to restrict the image's regions that need to be analyzed, and to cut down the number of scales that need to be considered in the hand-searching and gesture-recognition processes. The system performance is validated in real video sequences. In average the system recognized static gestures in 70% of the cases, dynamic gestures in 75% of them, and it runs at a variable speed of 5-10 frames per second.

Keywords: dynamic hand gesture recognition, static hand gesture recognition, context, human robot interaction, RoboCup @Home.

1 Introduction

Hand gestures are extensively employed in human non-verbal communication. They allow to express orders (e.g. "stop"), mood state (e.g. "victory" gesture), or to transmit some basic cardinal information (e.g. "two"). In addition, in some special situations they can be the only way of communicating, as in the cases of deaf people (sign language) and police's traffic coordination in the absence of traffic lights.

Thus, it seems convenient that human-robot interfaces incorporate hand gesture recognition capabilities. For instance, we would like to have the possibility of transmitting simple orders to personal robots using hand gestures. The recognition of hand gestures requires both hand's detection and gesture's recognition. Both tasks are very challenging, mainly due to the variability of the possible hand gestures (signs), and because hands are complex, deformable objects (a hand has more than 25 degrees of freedom, considering fingers, wrist and elbow joints) that are very difficult to detect in dynamic environments with cluttered backgrounds and variable illumination.

Several hand detection and hand gesture recognition systems have been proposed. Early systems usually require markers or colored gloves to make the recognition easier. Second generation methods use low-level features as color (skin detection) [4][5],

J. Baltes et al. (Eds.): RoboCup 2009, LNAI 5949, pp. 46–57, 2010.

shape [8] or depth information [2] for detecting the hands. However, those systems are not robust enough for dealing with dynamic conditions; they usually require uniform background, uniform illumination, a single person in the camera view [2], and/or a single, large and centered hand in the camera view [5]. Boosted classifiers allow the robust and fast detection of hands [3][6][7]. In addition, the same kind of classifiers can be employed for detecting static gestures [7]; dynamic gestures are normally analyzed using Hidden Markov Models [4][23]. 3D hand model-based approaches allow the accurate modeling of hand movement and shapes, but they are time-consuming and computationally expensive [6][7].

In this context, we are proposing a robust and real-time hand gesture recognition system to be used in the interaction with personal robots. We are especially interested in dynamic environments such as the ones defined in the *RoboCup @Home league* [20] (our team participates in this league [21]), with the following characteristics: variable illumination, cluttered backgrounds, (near) real-time operation, large variability of hands' pose and scale, and limited number of gestures (they are used for giving the robot some basic information). It is important to mention that in the new RoboCup @Home league' rules gesture recognition is emphasized: *An aim of the competition is to foster natural interaction with the robot using speech and gesture commands* (2009's Rules book, pp. 7, available in [20]). For instance, in the new "Follow me" test, gesture recognition is required to complete adequately the test (2009's Rules book, pp. 23: *When the robot arrives at [...] it is stopped by a HRI command (speech, gesture recognition or any other 'natural' interaction), and using HRI the robot should either move backwards, move forward, turn left or turn right [...]. Then the robot is commanded using HRI to follow the walker.).*

The proposed system is able to recognize static and dynamic gestures, and their most innovative features include:

- The use of context information to achieve, at the same time, robustness and real-time operation, even when using a low-end processing unit (standard notebook), as in the case of humanoid robots. The use of context allows to adapt continuously the skin model used in the detection of hand candidates, to restrict the image's regions that need to be analyzed, and to cut down the number of scales that need to be considered in the hand-searching and gesture recognition processes.

- The employment of boosted classifiers for the detection of faces and hands, as well as the recognition of static gestures. The main novelty is in the use of innovative training techniques - active learning and bootstrap -, which allow obtaining a much better performance than similar boosting-based systems, in terms of detection rate, number of false positives and processing time.

- The use of temporal statistics about the hand's positions and velocities and a Bayes classifier to recognize dynamic gestures. This approach is different from the traditional ones, based on Hidden Markov Models, which are not able to achieve real-time operation.

This article is focused on the description of the whole system, and the use of context to assist the gesture recognition processes. In sections 2 the rationale behind the use of context information in the proposed gesture recognition system is described. In section 3 the whole gesture recognition system and its modules are described. Results of the application of this system in real video sequences are presented and analyzed in section 4. Finally, some conclusions of this work are given in section 5.

2 Context Integration in HRI: Improving Speed and Robustness

Visual perception in complex and dynamical scenes with cluttered backgrounds is a very difficult task, which humans can solve satisfactorily. However, computer and robot vision systems perform very badly in this kind of environments. One of the reasons of this large difference in performance is the use of context or contextual information by humans. Several studies in human perception have shown that the human visual system makes extensive use of the strong relationships between objects and their environment for facilitating the object detection and perception ([13]-[17], just to name a few). Context can play a useful role in visual perception in at least three forms [18]: (i) Reducing perceptual aliasing: 3D objects are projected onto a 2D sensor, and therefore in many cases there is an ambiguity in the object identity. Information about the object surround can be used for reducing or eliminating this ambiguity; (ii) Increasing perceptual abilities in hard conditions: Context can facilitate the perception when the local intrinsic information about the object structure, as for example the image resolution, is not sufficient; (iii) Speeding up perceptions: Contextual information can speed up the object discrimination by cutting down the number of object categories, scales and poses that need to be considered.

The recognition of static and dynamic gestures in dynamic environments is a difficult task that usually requires the use of image processing algorithms to improve the quality of the images under analysis and to extract the required features (color, movement and even texture information), and statistical classifiers to detect the hands and to classify the gestures. In HRI applications there exists a tradeoff between carrying out a detailed analysis of the images, using an image's resolution that allows recognizing gestures at a given distance of a few meters, which usually can take more than one second per image, and the requirement of real-time operation to allow a proper interaction with humans. Context can be used to deal with this situation, and to achieve, at the same time, robustness and real-time operation, even when using a low-end processing unit (standard notebook), as in the case of humanoid robots.

In this work, the main sources of context to be used are human faces appearing in the image, and the existence of a physical world with defined laws of movement. Main assumptions are:

- We are dealing with an HRI application in which a human is communicating with a robot using hand gestures. Therefore a frontal human face will be observed in some or even several frames of the video sequence.

- Frontal face detectors are much more robust than hand detectors, mainly due to the fact that a hand is a deformable object with more than 25 degrees, whose pose changes largely depending on the observer's viewpoint. In the literature it can be observed that frontal face detectors achieve a much higher detection rates than hand detectors, and they are much faster.

- The robot and the human have upright positions, and their bodies (hands, heads, main-body, etc.) move according with the physical rules (gravity, etc.). This allows (i) to make some basic assumptions about the relative position and scale of the objects, as well as about their orientation, and (ii) to track the position of detected objects (e.g. a face), and to actively determine their position in the next frames.

- Normally the human user is not wearing gloves and the hand-gesture is a part of a sequence, in which the hand is moved. Therefore, candidate hand regions can be detected using skin and motion information.

In the proposed gesture recognition system a face detector is incorporated, and the information about the detected face is used to: (i) adapt continuously the skin model using the pixels contained in a sub-region of the face's area, (ii) determine the image's region that need to be analyzed for detecting skin and movement, as well as new faces, (iii) cut down the number of scales that need to be considered in the hand-searching process, (iv) normalize the input to the dynamic gesture recognition module, so that it is translation's and scale's invariant. In addition, (v) hand-searching process is restricted to regions where a minimal amount of skin and movement is detected, and (vi) after detecting a hand for the first time, it is tracked until track is lost. Then, hand detection is restarted. In the next section all these processes are explained.

3 Proposed Hand Gesture Recognition System

The system consists of five main modules *Face Detection and Tracking* (FDT), *Skin Segmentation and Motion Analysis* (SMA), *Hand Detection and Tracking* (HDT), *Static Gesture Recognition*, and *Dynamic Gesture Recognition* (see figure 1).

The FDT module is in charge of detecting and tracking faces. These functionalities are implemented using boosted statistical classifiers [11], and the *meanshift* algorithm [1], respectively. The information about the detected face (DF) is used as context in the SMA and HDT modules. Internally the CF1 (Context Filter 1) module determines the image area that needs to be analyzed in the current frame for face detection, using the information about the detected faces in the past frame.

The SMA module determines candidate hand regions to be analyzed by the HDT module. The *Skin Segmentation* module uses a skin model that is adapted using information about the face-area's pixels (skin pixels) in order to achieve some illumination invariance. The module is implemented using the *skindiff* algorithm [9]. The *Motion Analysis* module is based on the well-known background subtraction technique. CF2 (Context Filter 2) uses information about the detected face and the human-body dimensions to determine the image area (HRM: *Hand Region Mask*) where a hand can be present in the image. Only this area is analyzed by the *Skin Segmentation* and *Motion Analysis* modules.

The HDT module is in charge of detecting and tracking hands. These functionalities are implemented using boosted statistical classifiers and the *meanshift* algorithm, respectively. CF3 (Context Filter 3) determines the image area where a hand can be detected in the image, using the following information sources: (i) skin mask (SM) which corresponds to a skin probability mask, (ii) motion mask (MM) that contains the motion pixels, and (iii) information about the hands detected in the last frame (DH: *Detected Hand*).

The Static Gesture Recognition module is in charge of recognizing static gestures. The module is implemented using statistical classifiers: a boosted classifier for each gesture class, and a multi-class classifier (J48 pruned tree, Weka's [19] version of C4.5) for taking the final decision. The Dynamic Gesture Recognition module recognizes dynamic gestures. The module computes temporal statistics about the hand's positions and velocities. These features feed a Bayes classifier that recognizes the gesture.

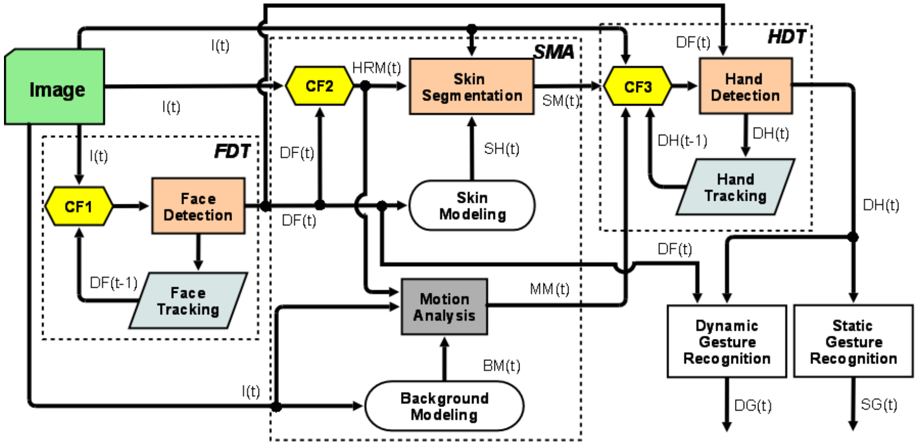


Fig. 1. Proposed hand gesture recognition system. CF*i*: *Context Filter i*. I: Image. DF: *Detected Face*. HRM: *Hand Region Mask*. SH: *Skin Histogram*. SM: *Skin Mask*: BM: *Background Model*. MM: *Motion Mask*. DH: *Detected Hand*. DG: *Dynamic Gesture*. SG: *Static Gesture*. *t*: Frame index. See main text for a detailed explanation.

3.1 Face Detection and Tracking

The FDT module is in charge of detecting and tracking faces. These functionalities are implemented using boosted statistical classifiers [11] and the *meanshift* algorithm [1], respectively. The face detector corresponds to a nested cascade of boosted classifiers, which is composed by several integrated (nested) layers, each one containing a boosted classifier. The cascade works as a single classifier that integrates the classifiers of every layer. Weak classifiers are linearly combined, obtaining a strong classifier.

The *meanshift* algorithm is used to predict the face position in the next frame. The seed of the tracking process is the detected face. We use RGB color histograms as feature vectors (model) for *meanshift*, with each channel quantized to 16 levels (4 bits) and the feature vector weighted using an Epanechnikov kernel [1]. The prediction given by *meanshift* is internally used by the CF1 (Context Filter 1) module to determine the image area that needs to be analyzed in the current frame:

$$x^v = \max\left(0, x^f - w^f\right) y^v = \max\left(0, y^f - h^f\right) w^v = \min\left(3 \cdot w^f, I_{width}\right) h^v = \min\left(3 \cdot h^f, I_{height}\right) \quad (1)$$

with x^f / y^f the *x*/*y* coordinates of the detected face (bounding box) upper-left corner, w^f / h^f the face's width/height, I_{width} / I_{height} the image's width/height, and x^v, y^v, w^v, h^v the coordinates, width and height of the image's area to be analyzed.

If a face is not detected in a frame, the prediction given by *meanshift* is used instead. The tracking module is reset after a fixed number of frames (about 200) in order to deal with cases such as faces incorrectly tracked or detected, or new persons entering the detection area.

3.2 Skin Segmentation and Motion Analysis

The SMA module determines candidate hand regions to be analyzed in the HDT module. *Skin Segmentation* is implemented using the *skindiff* algorithm [9]. S*kindiff* is a fast skin detection algorithm that uses neighborhood information (local spatial context). It has two main processing stages, pixel-wise classification and spatial diffusion. The pixel-wise classification uses a skin probability model G_t, and the spatial diffusion takes into account neighborhood information when classifying a pixel. The skin probability model is continuously adapted using information of the face-area's pixels (skin pixels). The adaptation is done by taking skin pixels of the face area, and updating a non-parametric skin model implemented using histograms:

$$G_t = G_{t-1}\alpha + \hat{G}_{face(t)}(1-\alpha), \tag{2}$$

where $\hat{G}_{face(t)}$ is estimated using the currently detected face, and G_o is the initial model, which can be initialized from a previously stored model (in our case the MoG model proposed in [22]).

The *Motion Analysis* module is based on the well-known background subtraction technique. CF2 (Context Filter 2) uses information about the detected face, the fact that in our system gestures should be made using the right hand, and the human-body dimensions to determine the image area (HRM: *Hand Region Mask*) where a hand can be present in the image:

$$\begin{array}{l} x^w = \max\left(0, x^f - 3\cdot w^f\right) y^w = \max\left(0, y^f - h^f\right) \\ w^w = \min\left(4.5\cdot w^f, I_{width}\right) h^w = \min\left(4\cdot h^f, I_{height}\right) \end{array} \tag{3}$$

with x^w, y^w, w^w, h^w the coordinates, width and height of the image's area to be analyzed. Note that just this area is analyzed by *Skin Segmentation* and *Motion Analysis* modules.

3.3 Hand Detection and Tracking

In order to detect hands within the image area defined by the HRM a cascade of boosted classifiers is used. Although this kind of classifiers allows obtaining very robust object detectors in the case of face or car objects, we could not build a reliable generic hand detector easily. This mainly because: (i) hands are complex, highly deformable objects, (ii) hand possible poses (gestures) have a large variability, and (iii) our target is a fully dynamic environment with cluttered background. Therefore we decided to switch the problem to be solved, and to define that the first time that a hand should be detected, a specific gesture must be made, the fist gesture. The fist is detected using a boosted classifier, similar to the one used for face detection, but built specifically for that gesture. The hand detector also takes as input the skin mask and the motion mask, and only analyzes regions where at least 5% of the pixels correspond to skin and movement. The integral image representation is employed to speedup this calculation (regions of different sizes can be evaluated very fast) [12].

The hand-tracking module is built using the *meanshift* algorithm [1]. The seeds of the tracking process are the detected hands (fist gesture). We use RG color and rotation invariant LBP histograms as feature vectors (model) for *meanshift*, with each channel quantized to 16 levels (4 bits). The feature vector is weighted using an Epanechnikov kernel [1]. Rotation invariant LBP features encode local gradient information, and they are needed because if only color is used, some times *meanshift* tracks the arm instead of the hand.

As already mentioned, once the tracking module is correctly following a hand, there is no need to continue applying the hand detector, i.e. the fist gesture detector, over the skin blobs. That means that the hand detector module is not longer used until the hand gets out of the input image, or until the *meanshift* algorithm loses track of the hand, case where the hand detector starts working again. At the end of this stage, one or several regions of interest (ROI) are obtained, each one indicating the location of a hand in the image. This module is explained in detail in [10].

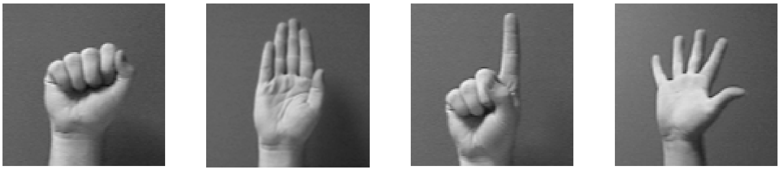
3.4 Recognition of Static Gestures

In order to determine which gesture is being expressed, a set of single gesture detectors are applied in parallel over the ROIs delivered as output of the HDT module (DH: Detected Hand). Each single gesture detector is implemented using a cascade of boosted classifiers. The learning framework employed for building and training these classifiers is described in [11]. Currently we have implemented detectors for the following gestures: *fist*, *palm*, *pointing*, and *five* (see figure 2).

Due to noise or gesture ambiguity, it could happen than more than one gesture detector will give positive results in a ROI (more than one gesture is detected). For discriminating among these gestures, a multi-gesture classifier is applied. The used multi-class classifier is a *J48 pruned tree (Weka's* [19] *version of C4.5)*, built using the following four attributes that each single gesture detector delivers:

- *conf*: sum of the cascade confidence's values of windows where the gesture was detected (a gesture is detected at different scales and positions),
- *numWindows*: number of windows where the gesture was detected,
- *meanConf*: mean confidence value given by *conf/numWindows*, and
- *normConf*: normalized mean confidence value given by *meanConf/maxConf*, with *maxConf* the maximum possible confidence that a window could get.

This module is explained in detail in [10].



Fist Palm Pointing Five

Fig. 2. Hand gestures detected by the system

3.5 Recognition of Dynamic Gestures

The Dynamic Gesture Recognition Module (DGRM) stores and analyzes sequences of detected hands (DH) online. The number of stored detections is fixed, so older detections that would exceed the predefined capacity are discarded as new detections arrive. Stored detections are discarded altogether when an inactivity condition is detected (still hand, hand out of camera range). Every time a new detection arrives, subsequences of the stored sequence that end with this new detection are analyzed. This analysis consists of computing a feature vector that comprises geometric and kinematical characteristics of the subsequence. Each subsequence's feature vector is fed to a Naïve Bayes classifier, which calculates a score for each possible dynamic gesture. This score represents the likelihood of the gesture in the given subsequence. In other words, every time a new detection (DH) arrives, a set of scores associated to each gesture is obtained (each score corresponding to a given subsequence). For each gesture, the highest of these scores is taken to be the best likelihood of that gesture having occurred, given the last frame. Finally, for each frame and each gesture, only this highest score is kept.

The highest scores alone could be used to determine the recognized gesture at the moment. However, in order to add robustness, the score should be consistently high during a interval of frames. So, for each gesture, the moving average of the last k highest scores is kept. In any given moment, the gesture with the best moving-average (*bma*) score is declared as the recognized gesture of that moment. Since not any frame is a real-end of a gesture, gesture segmentation becomes a problem. Thresholding the *bma* is a possible approach for gesture spotting. The thresholds can be learned from the training set. In addition, the current *bma* can be decremented in each round as a penalty for the subsequence from which it was extracted becoming older.

Each detected hand is represented as a vector (x, y, v_x, v_y, t), where (x, y) is the hand's position, (v_x, v_y) the hand's velocity, and t the frame's timestamp. In order to achieve translation and scale invariance, coordinates (x, y) are measured with respect to the face, and normalized by the size of the face. Using this vector, statistics (features) that characterize the sequences are evaluated. Some of the features are: mean hand's position in the x and y axis, mean hand's speed in the x and y axis, components of the covariance matrix of vector (x, y, v_x, v_y), area and perimeter of the convex hull of the (x, y) positions, average radius and angle with respect to a coordinate system placed on the mean (x, y) point, the percentage of points that fall on each cell of a 3x3 grid that exactly covers the positions of all detected hands, among others. Note that most of these features can be quickly evaluated using the same features evaluated in the previous frame (e.g. moving average).

4 Results

The whole gesture recognition system was evaluated in real video sequences obtained in office environments with dynamic conditions of illumination and background. In all these sequences the service robot interact with the human user at a variable distance of one to two meters (see example in figure 3). The size of the video frames is 320x240 pixels, and the robot main computer where the gesture recognition system runs is a standard notebook (Tablet HP 2710p, Windows Tablet SO, 1.2 GHz, 2 GB in RAM). Under these conditions, once the system detects the user's face, it is able to run at a variable speed of 5-10 frames per second, which is enough to allow an adequate interaction between the robot and the human user. The system's speed is variable because it depends on the convergence time of *meanshift* and the face and hands statistical classifiers. It should be noted that when the different context filters are deactivated and the complete image is analyzed, the system's speed is lower than 1 frame per second. This indicates that the use of context is essential to achieve the application requirements.

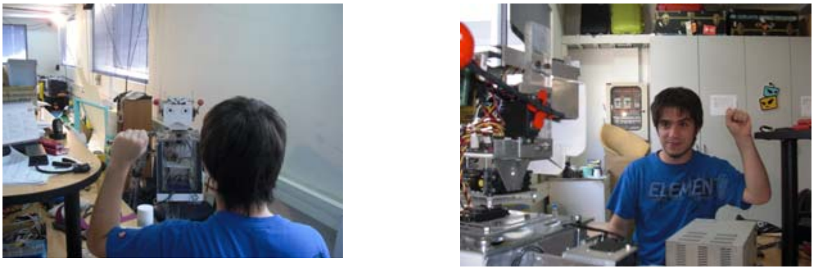


Fig. 3. Interacting with the robot in an unstructured environment

Recognition of Static Gestures. In order to evaluate this module, a database of 5 real-video sequences consisting of 8,150 frames, obtained in office environments, with variable illumination and cluttered backgrounds was built. In each sequence a single human was always interaction with our robot (altogether 4 different persons performing the 4 considered gestures). In figure 4 are shown the ROC curves of the single, static gesture detectors. Table 1 shows a confusion matrix of the multi-gesture recognition module, which consists of the four single, static gesture detectors and the multi-gesture classifier, evaluated on the same video sequences. The first thing that should be mention is that the hand detection system together with the tracking system did not produce any false negative out of the 8,150 analyzed frames, i.e. the hands were detected in all cases. From table 1 it can be observed that the gesture detection and recognition modules worked best on the *five* gesture, followed by the *pointing*, *fist* and *palm* gestures, in that order. The main problem is the confusion of the *fist* and *pointing* gestures, which is mainly due to the similarly of the gestures. In average the system correctly recognized the gestures in 70% of the cases. If the *pointing* and the *fist* gestures are considered as one gesture, the recognition rate goes up to 86%.

Recognition of Dynamic Gestures. We evaluate the proposed gesture recognition framework in the *10 Palm Graffiti Digits* database [23], where users perform gestures corresponding to the 10 digits (see example in figure 5). In the experiments the users and signers can wear short sleeved shirts, the background may be arbitrary (e.g, an office environment) and even contain other moving objects, and hand-over-face occlusions are allowed. We use the easy test set, which contains 30 short sleeve sequences, three from each of 10 users (altogether 300 sequences).

The system was able to detect and track hands in 266 of the 300 sequences (89%). In these 266 sequences, the dynamic gestures (i.e. digits) were correctly recognized in 84% of the cases. This corresponds to a 75% recognition rate (225 from 300 cases). It can be seen that this recognition rate is very similar to the one obtained in state of the art systems (e.g. [23], based on Hidden Markov Models), which are not able to operate in real-time or near real-time.

Table 2 shows a confusion matrix of the dynamic gesture recognition. It can be observed that the recognition rate of 6 digits is very high ("0"-"4", "8" and "9"). Two digits are recognized in most of the cases ("6" and "7"), and just the "5" digit has recognition problems. The "5" is confused, most of the time with the "3".

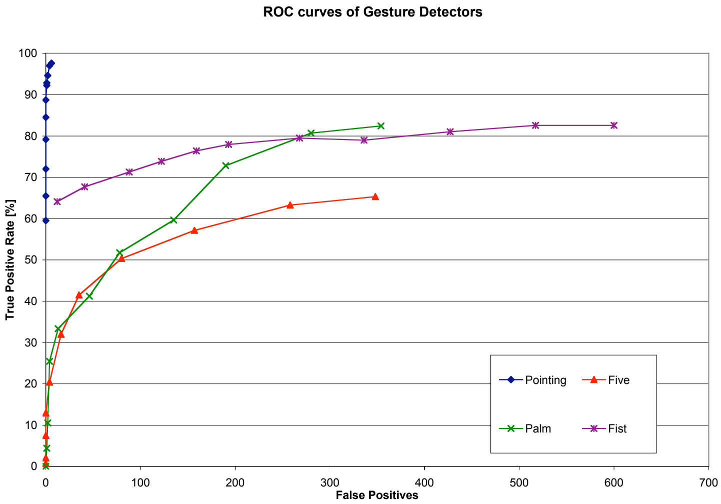


Fig. 4. ROC curves of the single, static gesture detectors

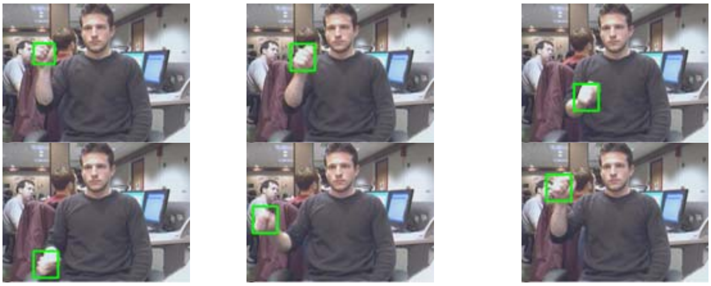


Fig. 5. Example of tracked hands in the *10 Palm Graffiti Digits* database [23]

Table 1. Confusion matrix of the final static, multi-gesture recognition module (rows: real gesture, columns: predicted gesture). RR: Recognition Rate.

	Fist	Palm	Pointing	Five	Unknown	RR (%)
Fist	1,533	2	870	9	15	63.1
Palm	39	1,196	10	659	15	62.3
Pointing	436	36	1,503	27	86	72.0
Five	103	32	6	1,446	127	84.3

5 Conclusions

In this article a hand gesture recognition system that allows interacting with a service robot, in dynamic environments and in real-time, was described. The system detect hands and static gestures using cascade of boosted classifiers, and recognize dynamic gestures by computing temporal statistics of the hand's positions and velocities, and classifying these features using a Bayes classifier. The main novelty of the proposed approach is the use of context information to adapt continuously the skin model used in the detection of hand candidates, to restrict the image's regions that need to be analyzed, and to cut down the number of scales that need to be considered in the hand-searching and gesture-recognition processes.

The system performance is validated in real video sequences. The size of the video frames is 320x240 pixels, and the robot computer where the gesture recognition system runs is a standard notebook (Tablet HP 2710p, Windows Tablet SO, 1.2 GHz, 2 GB in RAM). Under these conditions, once the system detects the user's face, it is able to run at a variable speed of 5-10 frames per second. In average the system recognized static gestures in 70% of the cases, and dynamic gestures in 75% of them.

Table 2. Confusion matrix of the dynamic gesture recognition module (rows: real gesture, columns: predicted gesture). TP: True Positives. FP: False Positives. RR: Recognition Rate.

	0	1	2	3	4	5	6	7	8	9	TP	FP	RR (%)
0	20	0	0	0	0	0	0	0	1	0	20	1	95
1	0	30	0	0	0	0	0	0	0	0	30	0	100
2	0	0	22	0	0	0	0	0	0	0	22	0	100
3	0	0	0	26	0	0	0	0	0	0	26	0	100
4	0	0	0	0	30	0	0	0	0	0	30	0	100
5	0	0	0	22	0	3	2	0	0	0	3	24	11
6	4	0	0	0	0	0	23	0	0	0	23	4	85
7	0	9	0	0	0	0	0	18	0	1	18	10	64
8	0	0	0	0	0	0	0	0	28	0	28	0	100
9	0	0	0	2	0	0	0	0	0	25	25	2	93

Acknowledgements

This research was partially funded by FONDECYT project 1090250, Chile.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-Based Object Tracking. IEEE Trans. on Pattern Anal. Machine Intell. 25(5), 564–575 (2003)
2. Liu, X., Fujimura, K.: Hand gesture recognition using depth data. In: Proc. 6th Int. Conf. on Automatic Face and Gesture Recognition, Seoul, Korea, pp. 529–534 (2004)
3. Kolsch, M., Turk, M.: Robust hand detection. In: Proc. 6th Int. Conf. on Automatic Face and Gesture Recognition, Seoul, Korea, pp. 614–619 (2004)
4. Dang Binh, N., Shuichi, E., Ejima, T.: Real-Time Hand Tracking and Gesture Recognition System. In: Proc. GVIP 2005, Cairo, Egypt, pp. 19–21 (2005)

5. Manresa, C., Varona, J., Mas, R., Perales, F.: Hand Tracking and Gesture Recognition for Human-Computer Interaction. Electronic letters on computer vision and image analysis 5(3), 96–104 (2005)
6. Fang, Y., Wang, K., Cheng, J., Lu, H.: A Real-Time Hand Gesture Recognition Method. In: Proc. 2007 IEEE Int. Conf. on Multimedia and Expo, pp. 995–998 (2007)
7. Chen, Q., Georganas, N.D., Petriu, E.M.: Real-time Vision-based Hand Gesture Recognition Using Haar-like Features. In: Proc. Instrumentation and Measurement Technology Conf. – IMTC 2007, Warsaw, Poland (2007)
8. Angelopoulou, A., García-Rodriguez, J., Psarrou, A.: Learning 2D Hand Shapes using the Topology Preserving model GNG. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 313–324. Springer, Heidelberg (2006)
9. Ruiz-del-Solar, J., Verschae, R.: Skin Detection using Neighborhood Information. In: 6th Int. Conf. on Face and Gesture Recognition – FG 2004, Seoul, Korea, May 2004, pp. 463–468 (2004)
10. Francke, H., Ruiz-del-Solar, J., Verschae, R.: Real-time Hand Gesture Detection and Recognition using Boosted Classifiers and Active Learning. In: Mery, D., Rueda, L. (eds.) PSIVT 2007. LNCS, vol. 4872, pp. 533–547. Springer, Heidelberg (2007)
11. Verschae, R., Ruiz-del-Solar, J., Correa, M.: A Unified Learning Framework for object Detection and Classification using Nested Cascades of Boosted Classifiers. Machine Vision and Applications 19(2), 85–103 (2008)
12. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 511–518 (2001)
13. Torralba, A., Sinha, P.: On Statistical Context Priming for Object Detection. In: Int. Conf. on Computer Vision – ICCV 2001, vol. 1, pp. 763–770 (2001)
14. Cameron, D., Barnes, N.: Knowledge-based autonomous dynamic color calibration. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 226–237. Springer, Heidelberg (2004)
15. Jüngel, M., Hoffmann, J., Lötzsch, M.: A real time auto adjusting vision system for robotic soccer. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 214–225. Springer, Heidelberg (2004)
16. Oliva, A.: Gist of the Scene, Neurobiology of Attention, pp. 251–256. Elsevier, San Diego (2003)
17. Strat, T.: Employing contextual information in computer vision. In: Proc. of DARPA Image Understanding Workshop, pp. 217–229 (1993)
18. Palma-Amestoy, R., Guerrero, P., Ruiz-del-Solar, J., Garretón, C.: Bayesian Spatiotemporal Context Integration Sources in Robot Vision Systems. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS (LNAI), vol. 5399, pp. 212–224. Springer, Heidelberg (2009)
19. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
20. RoboCup @Home Official website (January 2009), `http://www.robocupathome.org/`
21. UChile RoboCup Teams official website (January 2009), `http://www.robocup.cl/`
22. Jones, M.J., Rehg, J.M.: Statistical Color Models with Application to Skin Detection. Int. Journal of Computer Vision 46(1), 81–96 (2002)
23. Alon, J., Athitsos, V., Yuan, Q., Sclaroff, S.: A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation. IEEE Trans. on Pattern Anal. Machine Intell. (in press, electrically available on July 28, 2008)