Pose Classification Using Support Vector Machines

E. Ardizzone, A. Chella, R. Pirrone

DIAI - Dipartimento di Ingegneria Automatica e Informatica
University of Palermo
CERE - CEntro studi sulle Reti di Elaboratori
National Research Council

Viale delle Scienze - 90128 Palermo (Italy)
ardizzon@unipa.it
chella@unipa.it
pirrone@unipa.it

Abstract

The field of human-computer interaction has been widely investigated in the last years, resulting in a variety of systems used in different application fields like virtual reality simulation environments, software user interfaces, and digital library systems.

A very crucial part of all these systems is the input module which is devoted to recognize the human operator in terms of tracking and/or recognition of human face, arms position, hand gestures, and so on.

In this work a software architecture is presented, for the automatic recognition of human arms poses. Our research has been carried on in the robotics framework. A mobile robot that has to find its path to the goal in a partially structured environment can be trained by a human operator to follow particular routes in order to perform its task quickly. The system is able to recognize and classify some different poses of the operator’s arms as direction commands like “turn-left”, “turn-right”, “go-straight”, and so on.

A binary image of the operator silhouette is obtained from the gray-level input. Next, a slice centered on the silhouette itself is processed in order to compute the eigenvalues vector of the pixels co-variance matrix. This kind of information is strictly related to the shape of the contour of the operator figure, and can be usefully employed in order to assess the arms’ position. Finally, a support vector machine (SVM) is trained in order to classify different poses, using the eigenvalues array.

A detailed description of the system is presented along with some remarks on the statistical analysis we used, and on SVM. The experimental results, and an outline of the usability of the system as a generic shape classification tool are also reported.

1 Introduction and Related Work

In the last decade, the field of human-computer interaction (HCI) gained increasing importance, due to the impressive evolution of multimedia techniques. A lot of work has been spent in all the scientific and technological areas related to HCI, like virtual reality, software user interfaces, and digital library systems.

The work presented in this paper, is mainly focused on the field of human pose recognition. The correct interpretation of human body and arm pose, hand gesture, and facial expressions is a crucial issue in HCI because it represents the main information path in the interface between man and machine. Several approaches to this topic have been proposed in the last years. Some of them are mainly focused on 3D motion tracking of the whole body or of the arm only. These systems are very useful in virtual reality applications. Solina et al. [4] propose a model-based system that at first tracks the 2D silhouette of the human arm by means of iterative second-order curves fitting, and then reconstructs the 3D pose using the knowledge related to the kinematic constraints of the arm itself. Pentland [1] uses 3D blobs features to build a model of the human body: 3D geometry is inferred in real-time using an uncalibrated stereo system. Goncalves et al. [5] propose a monocular tracking system of the human arm using the extended Kalman filter: the 3D

Many other researchers are focused on the problem of recognizing hand gestures or facial expressions in order to provide their interpretation in terms of command sequences in visual interfaces [9, 12]. Graf et al. [6] propose a system based on the multiple cues of motion, shape and color in order to localize and segment faces from the rest of the input image. Kumar and Segen [7] recognize some pre-defined hand gestures as the mouse points and clicks. The system, first extracts from the background the connected region corresponding to the hand; then peaks and valleys are detected, that correspond to fingertips, and spaces between fingers. Hand gestures are recognized on the basis of the mutual position of these features. The proposed system is used in place of the mouse driver in order to manage commercial software applications.

The approach we propose, is aimed to the recognition of human arms pose as a way of direct interaction with a mobile robot, in order to obtain immediate execution of the commands bound to different gestures.

Robots acting in industrial environments, or the ones used for scientific purposes need to be repeatedly trained on the path to follow to reach the goal site or on the movements to accomplish to handle a tool or to perform welding, and so on. In the industrial framework, the human operator has to slowly move the robot, using manual commands, in order to make the robot to learn every single step of the movement. From this perspective, a visual command interface can result in a more natural way for the operator to teach the robot, thus reducing learning time, and allowing the robot to learn very complex movements. In scientific robotics, AI techniques are used to learn the path to the goal, and to cope with sensory data in order to avoid obstacles or to perceive harmful situations. In this environment, direct binding between visual perception and action can be used to induce purely reactive behaviors in the robot, and provide it with new information, not perceived before, in order to perform re-planning.

The proposed architecture is arranged as follows. We use Support Vector Machines (SVMs) in order to classify different operator poses acquired by the robot’s camera. The input image is first compared with the background in order to obtain the human silhouette using a difference operator. The resulting image is then thresholded, and the binary connected blob corresponding to the human figure is derived. Background images are continuously updated to take in account the current position of the robot. In order to classify poses, we describe them in terms of the eigenvalues of the co-variance matrix computed on the pixel rows of a vertical slice in the binary image. The slice is wide enough to surround the entire blob. The eigenvalues vector has proved to be a very good estimator of the blob contour configuration, that is representative of the pose itself. SVM have already been used in dynamic events recognition in Verri et al. [10]. The main differences between this work and our approach is that we use the eigenvalues of the pixel rows co-variance matrix to perform static pose classification in a single frame, while Verri performs inter-frame classification in terms of the evolution of the position of the box surrounding the moving object in the scene, and of the number of pixels belonging to the moving object at each frame.

The rest of the paper is arranged as follows. In section 2 the presented architecture is described in detail. Section 3 explains the experimental set-up. Finally, in section 4 conclusions are drawn and future work is discussed.

2 Pose Classification

In our approach, pose recognition and classification is achieved by means of statistical analysis of the human body contour shape. An original descriptor for this visual feature has been derived, and SVMs have been used to perform classification. The rest of this section is arranged in two parts: first, pre-processing and feature extraction from the input image will be explained. Next, all the implementation details of SVMs for the pose classification will be described.

In our framework, we are not interested in the recognition of hand or facial motions, but we’ve focused our attention on the pose of the whole human body. This kind of gestures can be described very well through the analysis of the body contour. Classical 2D shape descriptors like central moments, perimeter and area measurements, and Fourier descriptors are, in general, very sensitive to noise and heavy to compute. Moreover, to provide a detailed description of a shape, one has to take into account many different descriptors at once. A very compact descriptor for the body contour can be the vector of the eigenvalues computed from the co-variance matrix of the pixel rows belonging to a suitable slice of the binary image of the body itself: we called it the Eigenvalues Vector (EV). In what follows we’ll provide an explanation of this concept.

Suppose to have a 2D closed curve, and to sample it along the vertical coordinate, drawing a chord between the two contour points that we detect at each step. We can describe the contour, using the pattern of the chords’ length. Different shapes will have different patterns, while shapes that differ only for a scale factor will have similar patterns.
with the same relative displacements of the chords. Different shapes can be discerned through the analysis of the chords’ length configuration. The input images from the robot camera have been processed in a way similar to above example. The background is subtracted from the original 256x256 image, and the output is thresholded in order to obtain a binary blob of the body (see figure 1). Due to the robot moving, a new background image has been acquired every second. A suitable slice of the binary image is automatically selected (in our implementation we used 100 pixel wide slices) and the pixel rows $r_i (i = 1, \ldots, N = 256)$ are considered for the co-variance matrix computation. The mean value is computed as:

$$r = \frac{1}{N} \sum_i r_i,$$

while the co-variance matrix is:

$$C = \frac{1}{N} \sum_i (r_i - r) \cdot (r_i - r)^T \approx \frac{1}{N} \sum_i (r_i \cdot r_i^T) - (r \cdot r^T).$$

We can now define the EV as:

$$\text{EV} = \text{eigenval}(C)$$

(1)

Figure 1: From left to right, the input image, the difference with the background, and the thresholded image with the selected slice; at the bottom the eigenvalues histogram.

The EV takes into account the variance pattern of the vectors $r_i$: they’re like the chords. Each row vector has null components for the background points, while the silhouette points are all ones. EV describes in a compact way the mutual displacement of the silhouette contour points that describe the gesture. Different poses of the arms, legs, or other parts of the body, correspond to different vector patterns.

Classification has been performed on 7 different arm gestures, that correspond to different motion commands for the robot. In particular, both arms up indicate a stop command, both arms down mean continue previous operation (don’t care). Left arm up means go left (after starting), right arm up means go right. Left arm half up means turn left, while the right one stands for turn right. Finally, both arms half up indicate go back. Gesture classification task, is a multi-class one. The original SVMs set-up is useful to perform binary classification, while in multi-class problems it’s necessary to find a strategy to assess if a point belongs to a particular class. In our experimental paradigm, each gesture is mapped in a vector, and we have an experimental evidence that geometrical closeness between two vectors corresponds to similarity in gestures: close vectors have almost the same variance pattern, and they represent a similar silhouette configuration. Starting from this consideration we’ve trained 7 different SVMs in order to classify gestures in a one-to-many fashion. Each SVM has been tuned on a particular gesture. For each gesture a training set of Membership of a test point to a particular class has been computed on the basis of the positive distance from all the derived hyperplanes. In particular, if $d_i$ is the positive distance of a test point $x$ to the $i$-th hyperplane (i.e.: $i$-th SVM states that $x$ belongs to the $i$-th class), then membership degree percentage $m$ is computed as follows:

$$m = \frac{100 \sum_{i \mid d_i > 0} d_i}{\sum_{i \mid d_i > 0} d_i}.$$

(2)
If all \( d_i \leq 0 \) then a set of complementary (positive) distances is computed as follows:

\[
 d'_i = 100 - 100 \frac{d_i}{\sum_i d_i},
\]

then the membership degree is:

\[
 m = 100 \frac{d'_i}{\sum_id'_i}.
\]

In this way, we obtain positive distance values. Moreover, when \( |d_i| \) is higher (the point is very much out of the \( i \)-th class) then \( d'_i \) is very small.

### 3 Experimental Results

The SVMs have been trained using more than 100 thresholded images for every pose. Images have been taken from a doll figure, which has been progressively rotated 90° CW and CCW around its vertical axis (see figure 2). In table 1 are reported the classification results for the training set regarding each pose.

![Figure 2: Some examples of training thresholded images.](image)

In experiments with real images, a RWI B12 mobile robot has been employed, wandering in our lab. When in operation mode, the system detects automatically the slices of interest using a simple focus-of-attention strategy. A moving slice passes on the thresholded image, from left to right, and the EV is computed. When the \( \| \|_2 \) between the EV of the actual slice, and one of the support vectors is below a suitable threshold the process is stopped. The selected slice is then classified by SVMs. Using this approach, several images have been processed, with satisfactory results: the system classifies correctly all the proposed gestures, allowing the robot to perform the corresponding motion command. Figure 3 shows some good classifications.

### 4 Conclusions

The EV computation is a robust and compact way to describe shapes. The statistical nature of this feature guarantees low sensitivity to noisy data. The same approach is currently under investigation at our laboratory in order to provide
<table>
<thead>
<tr>
<th>Class</th>
<th>Pattern #</th>
<th>Support Vector #</th>
</tr>
</thead>
<tbody>
<tr>
<td>both up</td>
<td>185</td>
<td>42</td>
</tr>
<tr>
<td>both down</td>
<td>185</td>
<td>38</td>
</tr>
<tr>
<td>left up</td>
<td>153</td>
<td>42</td>
</tr>
<tr>
<td>right up</td>
<td>153</td>
<td>39</td>
</tr>
<tr>
<td>left half up</td>
<td>185</td>
<td>37</td>
</tr>
<tr>
<td>right half up</td>
<td>185</td>
<td>37</td>
</tr>
<tr>
<td>both half up</td>
<td>185</td>
<td>34</td>
</tr>
</tbody>
</table>

Table 1: Training statistics

Figure 3: Four test images, together with the selected slice; the eigenvalues histogram of the first image is also depicted.

descriptions for simple geometrical primitives depicted in gray-level images. In this case, under the assumption of a diffuse light source, and matte surfaces, the gray-level pixel rows are directly used in the computation. The key idea is that the one-dimensional gray-level pattern of each row has a particular configuration, depending on the shape represented in the slice under investigation. Thus, statistical analysis can be used to describe pattern arrangements.

The presented system needs some further investigation in the immediate future. In particular, we’re working to a more complex strategy for the scene investigation. In this new framework the system processes in parallel many regions of the image using growing slices. Growth is controlled by the classification score of the objects inside the slice: when the slice exhibits a score above some threshold for a particular class, its growth stops. If the slice is suitably wide, but doesn’t belong to any class, it is discarded. In this way, multiple recognition is possible, along with immediate classification.

Finally, we are currently working to the integration of this system in our three-level robotic architecture [3]. In this framework, an enhanced version of the conceptual spaces by Gärdenfors is used, that takes into account even dynamic perceptual events [2]. The conceptual level is used to map sensory level information onto the linguistic one, thus providing a possible solution to the symbol grounding problem. The conceptual model-based representation of situations and actions provides the robot with a planning capability, and can gives raise to both deliberative and reactive behaviors. In particular, direct command execution in response to a visual stimulus, can be regarded as a reactive behavior. The conceptual representation of this kind of percepts can be used to teach the robot in order to recover from unpredictable situations (i.e.: moving obstacles) and to add new knowledge about the subsequent changes in the environment.
Table 2: Test statistics for the images in figure 3

<table>
<thead>
<tr>
<th>Pose</th>
<th>Membership degree percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b. dw</td>
</tr>
<tr>
<td>both up</td>
<td>0</td>
</tr>
<tr>
<td>left up</td>
<td>0</td>
</tr>
<tr>
<td>right up</td>
<td>0</td>
</tr>
<tr>
<td>both down</td>
<td>60.60</td>
</tr>
</tbody>
</table>

References


