Posture Recognition in Visual Surveillance of Archaeological Sites

P. Spagnolo, M. Leo, G. Attolico, A. Distante

Istituto di Studi sui Sistemi Intelligenti per l’Automazione - C.N.R.
Via Amendola 166/5, 70126 Bari, ITALY
{spagnolo,leo,attolico,distante}@ba.issia.cnr.it

Abstract – The main aim of this work is to present a simple and reliable approach to the estimation of human body postures. The applicative context is the visual surveillance of an archaeological site. Motion detection and object recognition subsystems process image sequences coming from a still camera. Whenever a human is detected, his postures are characterized by the proposed pose estimation module. Then the results are fed to a HMM subsystem that identify the current activity of the examined subject. The proposed algorithm is based on an unsupervised clustering approach that makes the system substantially independent from any a-priori assumption about the possible output postures. The features selected for posture estimation are the horizontal and vertical histograms of binary shapes. A modified version of the Manhattan distance is used for both cluster identification and for run-time classification. After extensive experimental tests with different clustering schema, the BCLS algorithm (Basic Competitive Learning Scheme) has been selected. The proposed approach makes possible to change the number of classes, during the classification phase, without repeating the training step. Moreover it provides a measure of the reliability of its results. The proposed method has been verified on sequences acquired while typical illegal activities involved in stealing were simulated in a real archaeological site.

1. INTRODUCTION

In the last years, due to a diffuse request of safety, visual surveillance has attracted great interest from computer vision researchers. Pose recognition plays a very important role in characterizing human activities. Our application context is the visual surveillance of archaeological sites. After the detection of peoples, the system needs to be able to recover their correct posture, in order to allow a motion analysis subsystem to timely identify illegal actions. In this paper we concentrate only on the pose recognition step: details about the motion detection and the object recognition steps can be found in previous papers [22,23,24].

In literature there are three kinds of pose estimation approaches: model-free, based on indirect models and based on direct models.

Model free algorithms do not use an a-priori body model; marker points, object shapes and stick figures model based are used to recognize postures. In many works three points relative to centers of mass of head and hands have been used to model posture. In [1] these values have been calculated using color segmentation and blob segmentation techniques. In [2] boundary boxes have been used for modeling human pose; usually these are an intermediate representation during processing, while in the final representation other shapes (as ellipses) more similar to human figures, could be used. Stick-figures based approaches include information similar to human skeleton, and are very diffuse where gait studying is necessary. In [3] stick-figures are obtained through axial transformations, while in [4] distance transformation is used; in particular, in this work, it is possible to remove not interesting body parts, reducing computational time. Both axial and distance transformations give an approximation of the human skeleton. A different approach, without using a-priori model, is based on learning a direct mapping between features and posture; low level information are passed to the system as training. In [5] an unsupervised neural network based algorithm is trained with low level joints features.

Indirect model based approaches make use of an a-priori model for pose estimation. Different kinds of model can be used; usually head, hands position, or a generic description of the global human body are used. In [6] a body model given by a simple ratio between the limbs of the figure has been implemented, obtaining good results. In [7] the body model is composed by projections of a silhouette; the algorithm is considerably simple, and gives good results. It searches predefined body parts position, then it uses these information to calculate silhouette projections; these are compared with the same values referring to four principal postures. The main problem of this work is the possibility that figures assumes a posture different from four principal ones. So, authors are working to pass from a supervised posture classification to an unsupervised one. In [8] real time orientated histograms are calculated and compared with the same relative to a pre-recorded model in different posture: that with more similarity with the available histograms model is the estimated pose.

Direct model based approaches make use of an a-priori model that represents the observed subject. It is continuously updating using the observation of the figure, and gives information about posture at each time instant. About 40% of the methods proposed in literature use a similar approach. Utilized models are usually very detailed, they have an autonomous life in the system, and they are very useful in presence of occlusions. The human body model is often represented by means of connected joints; so, it can be represented in a space of states in which every axis represents a joint. To relate these information with postures an analysis by synthesis approach is used. This technique is very expensive in terms of computational time, so many constraints has been introduced to reduce dimension of the space of states. Rohr [9] considers only postures parallel to the image plane; Ong e Gong [10] create a subspace by mapping information through PCA. Other interesting works for reducing complexity have been effected by Pavlovic [11] and Moeslund and Granum [12].
For comparing image information with synthetic ones, a lot of abstraction level can be used: edges [13,14], silhouettes [15,16], contours [17], stick [18,19], blob[20], and texture [21].

In the rest of the paper, firstly motion detection and object recognition steps are presented (section 2); then, a reliable and extremely simple approach for pose estimation is proposed (section 3). Finally, the experimental results obtained on real image sequences acquired on an archaeological site, are reported (section 4).

II. SYSTEM OVERVIEW

A visual surveillance system is normally composed by a sequence of modules, each strictly depending on the results of the previous ones. To observe the area of interest the proposed system uses two cameras, configured as a master and a slave. When the master camera detects movements, the system controls a pan-tilt-zoom slave camera: the motion detection module extracts the foreground object whose nature and identity are recognized by another module. If it is classified as a human, the pose estimation subsystem is invoked to cluster its postures. Finally, an HMM module uses these results to identify the corresponding activity.

Firstly, visual surveillance has to recognize objects of interest (to be matched with given models) in a complex image. In our contest the objects of interest are moving and static objects that differ from a background model. To do it, we have implemented the background subtraction algorithm described in [22] and the update of the background model introduced in [23] to cope with problems due to light conditions. In order to reduce the undesirable noise, the algorithm for removing shadows described in [23] has been implemented. Then, all the moving blobs with an area greater than an appropriate threshold are considered as objects of potential interest (cars, people, ...). The detected objects are represented by their binary shape that is used as input for the classifier. The recognition step uses a neural network based approach described in [24]: its input features are an appropriate subset of the coefficients of the third level wavelet decomposition. At the end, regions classified as ‘human’ can be provided as input to the pose estimation subsystem.

III. POSTURE ESTIMATION

In the context of visual surveillance, performance is a fundamental requisite: so, to reduce the computational time, the proposed approach to pose estimation works on binary images. This reduces the information content but, on the other hand, makes the algorithm very fast and simple. Working on gray-level images would enable a more detailed analysis and discrimination of different postures: for example, as proposed in [7], single parts of body could be detected and tracked. Our choice reduces the complexity resulting in a very simple approach that can be easily integrated in a real time surveillance system.

The posture estimation problem has been seen as an unsupervised clustering problem in the space of the input images. This involves the selection of the features effectively representing the characteristics of each posture, the definition of a similarity measure between elements in the feature-space (used for building the prototypes during training and for classifying images at run-time), the identification of an unsupervised clustering method to be used for building the clusters on the base of the available training data.

A. Features Selection

The selected features must be able to correctly discriminate the different postures of interest. In fact, a pose is represented by a point in the multidimensional feature-space: the performance of the system is obviously strictly depending on the selected features. A wrong choice of the multidimensional space can make the data not separable, preventing any useful classification.

The selected features need to offer a wide discrimination capability at the minimum computational cost. Histograms of horizontal and vertical projections are a good trade-off between these requisites. Fig. 1 shows histograms relative to three different postures: a “standing” posture (upper image), a “bent” figure (the central one) and a “squatted” pose (lower image). In literature, these features have not been used in many applications, mainly due to their sensitiveness to scaling, translation and rotation. In our application these geometric aspects are much less relevant: using images acquired with a pan-tilt-zoom camera, there is no scaling because the control provides an observed blob of constant dimension in each frame. Even translation is not relevant, because the blob is centered in the analyzed window. Moreover, due to the natural standing position of people, rotations are practically negligible: different inclinations of the body correspond to different postures. In this context rotations are not a problem but an important cue for posture estimation.

In order to obtain more stable results, in particular in presence of scaling, histogram normalization is often used in literature. There are two common and extremely simple normalization: an area normalization (every element is divided by the total area) and a maximum normalization (each element is divided by the maximum) both providing results in the range [0,1]. Normalization changes the feature vectors, while histogram shapes are substantially unchanged; on the other hand, it is possible to obtain a smoothing of the histograms that can reduce interclass variations. The advantages obtained by normalizing, i.e. invariance to scaling, are compensated by the greater proximity of feature vectors in the multidimensional space that makes harder the correct clustering. The results obtained using normalized histograms have been compared with the same results obtained without normalization. Fig. 2 shows horizontal projections histogram both with and without a maximum normalization, plotted for the standing pose of Fig. 1. As expected, the shape of the curves is the same, but values are likely changed.
To obtain a good clustering of the static poses, it is necessary that histograms of horizontal and vertical projections would be well defined and with a low level of noise. Histograms are very sensitive to small flaws, caused, for example, by holes in the binary shape; their presence could seriously affect the histograms, making the whole process less reliable. In addition, the presence of noise degrades image quality and can affect the pose estimation. To reduce such artifacts, a smoothing operator has been implemented and applied. In particular, a median filter has been used: it is a non-linear spatial filter, ideal for reducing salt and pepper noise. It has been applied to the histograms over a window of 9 values.

B. Distance Selection

Using the selected features, images become points in a feature space and the pose recognition can be seen as a clustering problem. In a generic clustering algorithm, supervised or unsupervised, it is necessary to define a metric as a way for estimating the “proximity” of two postures, according to the selected features. The selected proximity measurement is a variation of the Manhattan distance: in the original version, it is:

\[ d_i(x, y) = \sum_{i=1}^{\text{DimX}} |x_i - y_i| \]  

where \( x \) and \( y \) are the histograms (with values \( x_i, y_i \)) and \( w_i \) are the weight coefficients. In this context there is no difference in significance of vector elements so all weights have been set to 1. This proximity measurement has the advantage of being simple and fast with respect to other more complex distances (such as the Euclidean one).

Unfortunately, it is still very sensitive to the geometrical position of the blob in the image: different position (due to translation or mirroring with respect to the vertical axial) can change its values up to preventing the detection of similarity. So, a variation of the traditional Manhattan distance has been introduced for increasing its reliability. For the vertical projections, the Manhattan distance became:

\[ d_j(Y1,Y2)= \min \left( \sum_{j=0}^{\text{DimY}} |Y1(j)-Y2(j)| ; \sum_{j=0}^{\text{DimY}} |Y2(j)-Y1(j)| \right) \]  

where the minimum is evaluated with \( i \) changing in the interval \([0, \text{DimY}-1]\). The only geometrical variation on the vertical axis is translation, because mirroring can affect only horizontal projections (mirroring with respect to the horizontal axis is very unlikely!). In fig. 3 some examples of translation and mirroring are shown:

For the horizontal projection, the modified distance is:

\[ d_i(X1,X2)= \min \left( \sum_{i=0}^{\text{DimX}} |X1(i)-X2(i)| ; \sum_{i=0}^{\text{DimX}} |X2(i)-X1(i)| \right) \]  

According to equations (2) and (3), overall similarity measurement is:

\[ d_i(Im1,Im2)=a \cdot d_i(X1,X2)+b \cdot d_i(Y1,Y2) \]  

where \( \text{Im1}, \text{Im2} \) are the two images to be compared, and \( a, b \) are the coefficients for weighting the horizontal and vertical projections. Several experimental tests have shown that the two histograms have the same relevance so the parameters \( a \) and \( b \) have been both set to 0.5.
C. Clustering

The previous selected features and the proposed distance measurement allow a clustering algorithm to group the available training images. The effectiveness of resulting clusters depends on the selected algorithm, on the characteristics of the defined distance and on the selected features. There are two main classes of clustering algorithms: supervised and unsupervised. In the supervised algorithms the training data are already labeled while in unsupervised ones this information is not available. The implemented algorithms use an unsupervised approach: in this way, no information about desired output is provided. A learning step is necessary to estimate basic characteristics of the resulting clusters.

In literature, several clustering techniques have been proposed: we have implemented and experimentally compared four different algorithms. Three of them are sequential (BSAS - Basic Sequential Algorithm Scheme, MBSAS - Modified Basic Sequential Algorithm Scheme, TTSAS - Two Threshold Sequential Algorithm Scheme) and one is based on a competitive scheme (BCLS - Basic Competitive Learning Scheme). Further information about these algorithms can be found in [25]. All these algorithms have been applied to the available data. The best results have been obtained using the BCLS. The great advantage of this algorithm is that it does not involve thresholds, being less sensitive to wrong initial setting. It requires a learning period for the selection of a “representative” for each cluster. During the run-time classification, to assign each image to the correct posture the previously defined distance is applied. The training phase is decisive to obtain good performance. The most important parameters for the BCLS algorithm are: the number of expected clusters (we have tested algorithm both requiring two or three output clusters), the quality of input images, the learning velocity during the training period (set by means of experimental tests), the number of training images (tests have been effected by using the complete data set or a part of it), the “representative” selection step and the order of presentation for the training images.

IV. EXPERIMENTAL RESULTS

The experiments have been performed on real sequences acquired with a TV camera at 30Hz. The resolution of the acquired images was 528 by 512 pixels. The windows used by the module for Posture Analysis had a resolution of 155 X 215 pixels. The test sequences have been acquired in a real archaeological, while the actions normally accomplished during illegal activities were simulated. The algorithms have been tested on 834 images (501 of a standing person, 218 of a bent one, and the remaining 115 of squatted poses) selected from the acquired sequences. The computational time obtained on a AMD Athlon XP 1600+ (with 256 Mb SDRAM and an hard-disk of 41 Gb working at 7200 rpm) shows that the proposed algorithms are fast enough to be suitable for a real time system.

The characteristics of the test images emphasize a problem for every posture recognition algorithm: sometimes it can be difficult to unequivocally assign an image to a specific cluster; even for a human. An important input parameter for the BCLS algorithm is the expected number of clusters that must be fixed apriori. We have done experimental tests looking for two or three output clusters, as it can be seen from table 1. The number of output clusters could appear weak: this is mainly due to the not excellent quality of the selected features. However, this choice is in accord with the subsystem, that will recognize the activities, based on the HMM: we expect the selected number of clusters to be adequate for this purpose. Experimental results have indicated that best results are obtained with a value of 0.2 for the learning velocity during the training period.

An important parameter for clustering algorithms is the order in which training data are given as input. So, in order to obtain reliable estimation of the performance of different algorithms and of the same algorithm when changing its parameters, a set of random sequence have been stored and used for feeding the training phase during tests. The results in the following table show the mean of the corresponding value over several runs during which the input data have been examined in different orders.

As it can be seen, the performance of the system does not significantly depend on the size of the training set. Computational time is not a critical problem: in fact with the proposed hardware architecture it is possible to process about 84 frame/sec. So, this subsystem can be integrated in a visual surveillance system working in real-time. It is important to note that the system is substantially independent from objects carried by people: if their

<table>
<thead>
<tr>
<th>% data set images used as training</th>
<th>20</th>
<th>40</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 clusters</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% correct detection (100 run)</td>
<td>97.3</td>
<td>97.4</td>
<td>94.9</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>2.4</td>
<td>1.5</td>
<td>4.9</td>
</tr>
<tr>
<td>Computational time</td>
<td>9.88</td>
<td>10.37</td>
<td>12.75</td>
</tr>
<tr>
<td>3 clusters</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% correct detection (100 run)</td>
<td>89.9</td>
<td>90.0</td>
<td>90.8</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>5</td>
<td>4.4</td>
<td>7.1</td>
</tr>
<tr>
<td>Computational time</td>
<td>15.23</td>
<td>15.87</td>
<td>16.11</td>
</tr>
</tbody>
</table>
In previous experiments the output postures were assigned to a number of different classes (two or three) defined during the training phase: this information can be not easily available for the module. Moreover, an important information would be a measure of the confidence in the classification done by the system. To cope with both these problems two extreme posture have been defined: “standing” and “squatted”. The training has been done only with features relative to these two postures. During the classification phase, the system provides a couple of complementary coefficients (in the interval [0,1]) for each image: they represent the similarity of the current pose with respect to these two extreme postures. They allow to change the number of clusters during the classification phase (changing the partition of the interval [0,1]) and at the same time provide useful hints about how trustable the classification can be considered. According with (2) and (3) the similarity coefficients are:

\[
i(Im, "standing") = \frac{d(I, "standing" r.)}{d(I, "standing" r.)+d(I, "squatted" r.)} \times 100
\]

\[
i(Im, "squatted") = \frac{d(I, "squatted")}{d(I, "standing" r.)+d(I, "squatted" r.)} \times 100
\]

where \(i(Im, "standing")\) is the similarity between current image and the pose “standing”, and “standing”\(r\). is the prototype of the pose “standing”. The same stands for the pose “squatted”. The figure 6 shows these coefficients for some typical situations. In presence of a postures similar to a representative one (first and second columns, “standing” and squatted) poses respectively) the similarity coefficients clearly identify the appropriate class; for an image in a bended posture, the similarity coefficients takes on intermediate values. Therefore, this kind of classification provides both a way for estimating the confidence of the classification and the possibility to generate a variable number of clusters (changing an appropriate set of thresholds on the output coefficients) without requiring a new and specific training phase.

<table>
<thead>
<tr>
<th>Im</th>
<th>Standing</th>
<th>Squatted</th>
<th>Bent</th>
</tr>
</thead>
<tbody>
<tr>
<td>I(Im, &quot;standing&quot;)</td>
<td>77.51</td>
<td>16.37</td>
<td>50.89</td>
</tr>
<tr>
<td>I(Im, &quot;squatted&quot;)</td>
<td>22.49</td>
<td>83.63</td>
<td>49.11</td>
</tr>
</tbody>
</table>

Fig. 6: These coefficients express how much each image in the first row is similar to the extreme postures: standing and squatted.

In this paper a simple and reliable approach to the estimation of body postures in the visual surveillance of archaeological site has been presented.

In order to reduce the a-priori knowledge required by the approach, an unsupervised clustering approach has been used. The features used for this purpose has been the horizontal and vertical histograms of the binary shapes associated to humans. A modified version of the Manhattan distance has been defined for both cluster building and run-time classification. After extensive experimental tests, the BCLS algorithm (Basic Competitive Learning Scheme) has been selected as unsupervised clustering technique for the construction of classes. The whole approach has been verified on sequences acquired while the typical illegal activities involved in stealing were simulated in an real archaeological site. The study presented in this work shows that BCLS algorithm is well suited to clustering human poses starting from the binary images of the people in the scene. The classification performance is very high although no a-priori knowledge of the scene has been used. A specific method has been designed that provides in output not only the classification of features, but also a similarity coefficient that suggests the affinity between the examined features and two main reference poses. Moreover the experiments prove that the method involves low computational load and allows the system to work in real time. The results of this work are intended to be the input data for the recognition of human behaviors that is based on a HMM based technique.

VI. REFERENCES


