A MACHINE LEARNING APPROACH FOR HUMAN POSTURE DETECTION IN DOMOTICS APPLICATIONS

L.Panini, R.Cucchiara*,*
*D.S.I. University of Modena, via Vignolese 905 - 41100 Modena, Italy
email: luca.panini@libero.it ; rita.cucchiara@unimo.it

ABSTRACT

This paper describes an approach for human posture classification that has been devised for indoor surveillance in domotic applications. The approach was initially inspired to a previous works of Haritaoglou et al. [2] that uses histogram projections to classify people's posture. We modify and improve the generality of the approach by adding a machine learning phase in order to generate probability maps. A statistic classifier has then defined that compares the probability maps and the histogram profiles extracted from each moving people. The approach results to be very robust if the initial constraints are satisfied and exhibits a very low computational time so that it can be used to process live videos with standard platforms.

1. INTRODUCTION

Human posture detection is now widely explored in many application contexts, ranging from content-based retrieval, surveillance, indoor and outdoor monitoring, virtual reality until animation and entertainment. In particular, we are interested in human posture detection in the framework of home-human interface. The scope is twofold. The former is to improve the control of electronic parts in the house, in order to aid the inhabitants (especially if disabled) in daily life. The latter is to find a way to monitor people health without some invasive approaches often proposed in tele-medicine. In this case an intelligent video-based tele-assistence service, for remote control of disabled or elderly people, needs advanced computer vision techniques able to detect moving objects, classify people, localize them in the environment and detect their posture and behavior. A typical application is detecting if the person is walking, sitting or falling down and lying on the floor and eventually sending an alarm to a remote human operator.

To this aim this paper proposes an approach for people posture classification. This topic is very popular in this moment and many papers and research works address this problem. Nevertheless most of the proposal still lack of generality and are too tailored to manually set models.

In this paper we explore an approach that integrates and improves some assessed methods of human posture analysis [1,2,3] with machine learning phase to model people posture features. These features are then used in statistical classifier in order to predict the human posture. Finally a finite state chart is adopted to supervise tracking and classification process and generate alarms if required.

In the next section some related works are described. Section 3 states some Initial assumptions and working hypothesis. The defined approach is proposed in Section 4. Then the learning environment is described and eventually Section 6 shows results on different tests. Conclusions end the paper.

2. RELATED WORKS

In recent years an increasing number of computer vision projects dealing with detection and tracking of human posture have been proposed and developed. Analyzing literature's works, two basic approach to the problem stand out. From one side, systems (like Pfinder[5] or W^4[3]) use a *direct* approaches, based the analysis on a detailed human body model: an effective example is the *Cardboard Model*[4]. In many of these cases, an incremental predict-update method is used, retrieving information from every body part. Nevertheless these systems are generally too sensitive, when loosing information about features, needing often a reboot phase. For this reason, the segmentation process has to be as precise as possible using specific human cues (eg. skin detection). Often this can contributes to system instability because some of these features could not be found in every frame (caused e.g. by overlapping). In order to bypass these drawbacks, where no body parts control needed, many researchers deal with the problem in a *indirect* way using less, but more robust, information about the body. Many of these approaches are based on human body silhouette analysis. The work of Fujiyoshi et. Al. [1] uses a synthetic representation (*Star Skeleton)* composed by extremal boundary points. In [2] Haritaoglu et al. add to W^4 framework [3] some techniques for

Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP'03)
0-7695-1948-2/03 $17.00 © 2003 **IEEE**

human body analysis using only information about silhouette and its boundary. They first use hierarchical classification in main and secondary postures, processing vertical and horizontal histogram profiles from the body silhouette. Then they locate body parts on the silhouette boundary's corner. Our approach is similar to [2], as well as concerning histogram-based features, but differently from it, it is not based on a priori defined model. In fact the main strength of our approach is a machine learning phase, exploited to create feature models, further used in the classifier.

3. ASSUMPTIONS AND CONSTRAINTS

The approach aims to define a framework for posture classification over previously detected people. Thus the part of moving object detection and people classification is not the strength of the paper.

In order to be as flexible as possible, we constrain the problem dimensionality to the 2D case, reducing the analysis to monocular images. This allows our application to be applied also in those situations where a 3D reconstruction can't be issued, either for environmental or computational limits. Therefore we suppose to have a point of view where the human posture can be easily perceived without ambiguities also in the image plane.

We consider images from indoor environment with an unknown but fixed camera position. The method can be exploited in outdoor environment too but we exclude high luminance variation and other artifacts that could add difficulties in moving object detection.

Moreover, one people at time is considered: in this paper we don't address problems of people overlapping, neither problems of human parts hiding, problems that we could suppose solved by a very precise moving object detection and tracking systems or by a multi-camera system.

4. THE PROPOSED APPROACH

A high level system layout is represented in Figure 1, where gray block indicates parts referred to this paper.

As previously stated, we assume to have the capability to extract and track human bodies from a movie. In our tests we exploit a system called Sakbot (*Statistical & Knowledge-Based Object Detection*) [6] able to detect and track moving objects (called MVOs, Moving Visual Objects). The basic process is based on a adaptive background suppression, where background is modeled using statistical and knowledge-based data. Sakbot can extract foreground objects, distinguish real MVOs from shadows and other artifacts such as "ghosts" (i.e. errors in background modeling), and track MVOs during the time. Every tracked MVO is then processed by a pre-classifying phase, in order to distinguish between *people* and *non-people* tracks. For this work, we did not address this problem, using a very simple classifier, based on geometrical features.

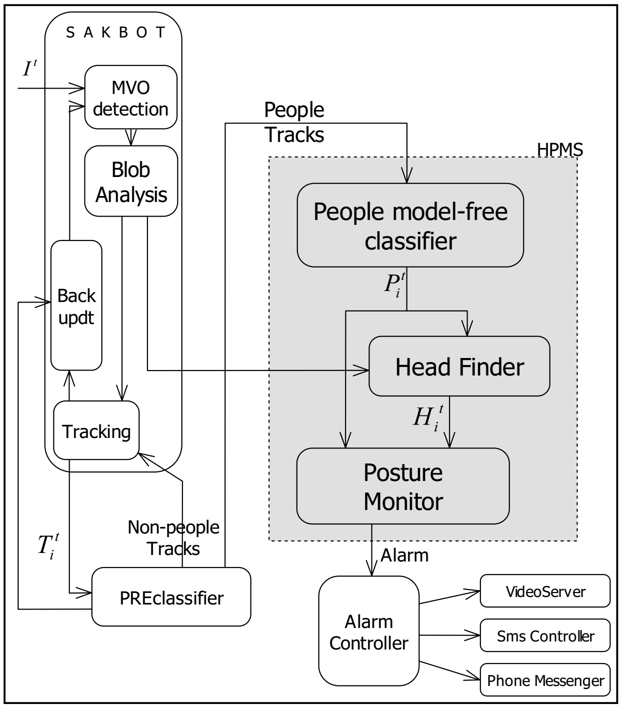


Figure 0 – Layout of the system

As in Figure 1, MVOs classified as "People tracks" are then passed to HPMS (*Human Posture Monitoring System*) which detects and analyzes the posture of the person in order to recognize if dangerous situations occurred (e.g. falling or long time inactivity). This part is basically composed by three block. The first, PMFC (*Posture Model-Free Classifier*) is a statistical example-based classifier able to distinguish between four main body postures. It uses a human-free model created by a specific training phase. When human body posture is recognized, the system pass it to the next block (*Head Finder*) which investigate the silhouette boundary, looking for the *top-of-the-head* point. This is done using a Jarvis-Convex Hull scan and light tracking phase with topological rules. The Head Finder has the main goal to make the posture detection more effective and robust, discriminating between ambiguous cases.

Finally the control switches to the third block (*Posture Monitor*) which detect dangerous situation judging the history of body posture and head position. This is done using a finite state chart which models person's behaviour. The posture monitor generates, if required, an alarm as indicated in Figure 0.

4-1. Posture model free classifier

In order to recognize human body posture, we used a similar knowledge classification, proposed by [2]. In the matter of fact, we discriminate four postures which represent our classifier's states: *standing, crawling, laying* and *sitting*. An algorithm layout is reported in Figure 1. Every MVO, classified as a person, is processed extracting information from its silhouette as in [6]. Let B a cloud of 2D points, we compute the cardinality (indicated with #) of horizontal and vertical projections respectively π and θ as follow:

$$\theta(x) = \#\left\{(x_p, y_p) \in B \mid x_p = x\right\} \qquad (1)$$

$$\pi(y) = \#\left\{(x_p, y_p) \in B \mid y_p = y\right\} \qquad (2)$$

These two integer functions represent the basic classifier features. Although other features have been explored and could be added to the classifiers, in our tests, they have been proved to be reliable enough.

The classifier works in two different modalities: *training* and *run-time*. In the first phase we construct two probability maps for every posture (state), using the respective silhouette projections.

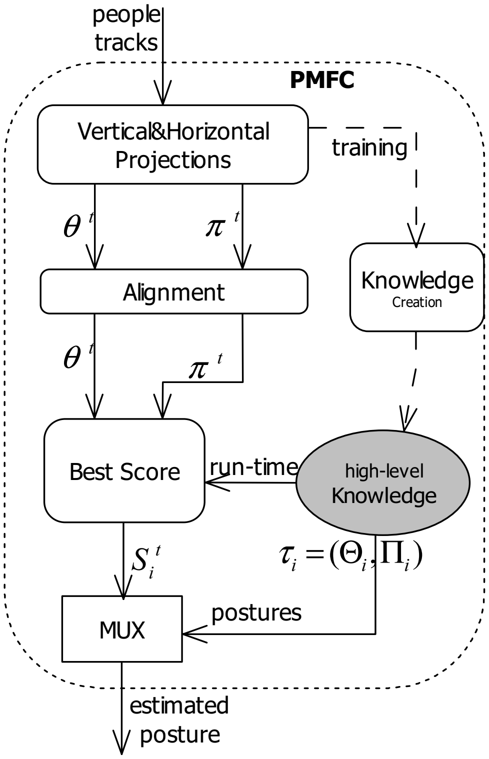


Figure 1 - Block Diagram for PMFC

Let $B_i^t(x,y)$, t=1.. T_i, a training set of T_i 2D-images referred to a the *i*-th state, and let (θ_i^t, π_i^t) it's projection silhouette couple series.

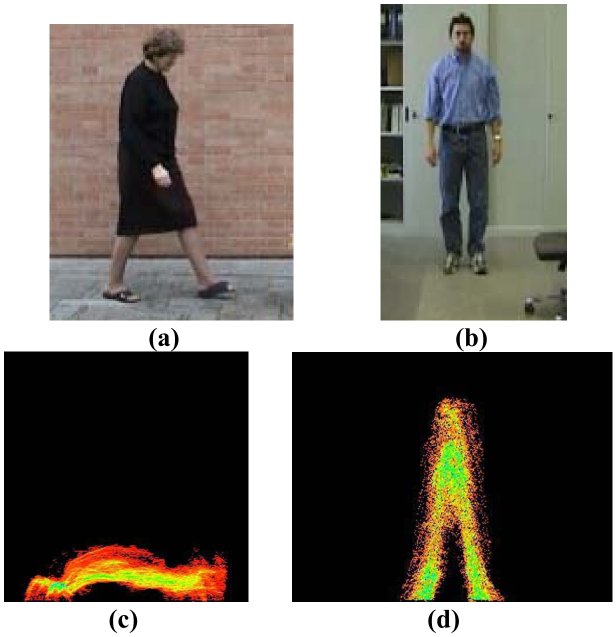


Figure 2 –Standing pose examples (a),(b). Graphical representation of the Horizontal Θ_1 (c) and Vertical Π_1 (d) probability maps for *standing* state (*i*=1).

We construct the couple of 2D *probability maps* of the state (Θ_i, Π_i) as follow,

$$\Theta_i(x,y) = \frac{1}{T}\sum_T g(\theta_i^t) \qquad (3)$$

where $g(f)$ is

$$g(f)(x,y) = \begin{cases} 1 & if\ y = f^t(x) \\ 0 & elsewhere \end{cases} \qquad (4)$$

The construction for Π_i is analogous, using π instead of θ. In run-time modality, the PMFC module compares the current projection couple with the correspondent probability maps for every state. First, the *Alignment* step shifts the π and θ histograms in the central part to the probability map and the *Best score* module (see Figure 2) computes the S score quantity.

To define the best classifying rule we have tested five different functions over different test-set models. The problem is how to compare an 1D histogram with a 2D probability map constructed by histograms overlap.

The five methods belong to three different approaches:

- method 1 and 2 provide a 2D comparison by matching the feature histogram directly over the map and combining the with a product and sum operator, respectively mean of the matched probabilities.
- method 3 synthesizes the map in two 1D histograms, considering for every histogram's entry the mean and variance values. Then a Mahalanobis distance is computed between these histograms and the π and θ histograms.
- Finally, the 4 and 5 approaches consider a comparison between the feature and the mean histogram, using respectively intersection and difference between histograms.

measure function	efficacy rate	time (msec)
1	96,63%	0,1759
2	96,34%	0,0646
3	82,57%	0,2416
4	96,34%	0,0736
5	60,30%	0,0009

Table 1: Result on different measure functions

The comparison results between classifying rules are reported in Table 1. This shows that the first devised classifier outperforms other methods. We believe that in this framework, as frequently happens, a multi-classifier could improve performance considerably.

With this a comparable execution time has been achieved with it. Below a brief explanation of the method is exposed.

Let $\tau_i = (\Theta_i, \Pi_i)$ a probability map couple for the state *i*. We consider the quantity S_i^θ obtained as:

$$S_i^\theta = \frac{1}{width(\theta)} \sum_{x=0}^{x=width(\theta)} \Theta_i\left(x, \theta(x)\right) \qquad (0.5)$$

In the same way we define S_i^π using Π_i. The final score S_i is computed as the correlation between the two scores as :

$$S_i = S_i^\theta \cdot S_i^\pi \qquad (0.6)$$

5. A FRAMEWORK FOR MACHINE LEARNING

We set up a dedicated visual framework in order to have the complete control in the training phase. In particular the probability maps of Figure 3 have been computed without any a priori defined model but only with a machine learning phase.

The GUI (graphic user interface) depicted in Figure 4(a) allows the user to load a training video. The tools indicates the detected and tracked MVOs, frame by frame, asking the user its supposed posture. A very friendly environment prevents the necessity to ask a frame by frame user interaction: in fact he has to indicate when the people change its status from a posture to another . Also the *unknown* status is accepted. In this manner in few minutes very reliable probability maps can be set ("Not Classified" in Figure 4(a)). This training phase can be improved, iterated or changed whenever required.

6. EXPERIMENTAL RESULTS

The system has been designed to meet real-time constraints and to process a sufficient number of frame per second to be reactive and adaptive enough for possible alarm. Here we report some results on videos acquired in different contexts. In particular we describe three tests:

1. the *efficacy* in posture detection *over the same videos where the training phase has been provided*: this results could be interesting for a domotic surveillance application, supposing that an initial training is done in the specific context on the specific people, as a sort of initial calibration of the system;
2. the *efficacy* in posture detection *over other videos, but taken from the same camera system*;
3. the *efficacy* and the generality of the model in posture detection *on different videos* (different camera, different scene, different actors) w.r.t. the training set;
4. the efficiency in terms of *frame rate*.

In particular, here we present results against ground-truth on six videos with three different actors.

Examples of frames are reported in Figure 4(b)…(f).

Videos have been taken with two different cameras, in 320x240 frame size, compressed in MPEG-2 format before their processing. For processing evaluation we used a test machine with this characteristics: 1,4 GHz P4, 256 MB RAM.

Three videos are taken from the same outdoor environment (Figure 4(b)…(d)), while the last three ones, in Figure 4(d) and 4(e), from a different indoor scene. In Table 2 the efficacy rate is shown to describe the system behaviour over the three test mentioned (point 1,2,3): the columns of the table report in order: name of the video (took from its subject), the name of the video used for model training, the pose number, i.e. the number of MVOs classified as person which posture has to be classified, the efficacy rate, that is the number of correctly detected poses over the total poses' number, and the achieved frame per second.

Video	Model	Pose Example	Efficacy Rate	fps
Luca1	Luca2	1318	93,10%	13,95
Luca2	Luca2	1561	96,83%	13,04
Albertina1	Luca2	380	98,93%	10,42
Roberto2	Roberto	742	67,79%	12,46
Roberto2	Luca2	742	54,04%	11,33
Roberto3	Luca2	449	89,31%	11,18

Table 2 - Test result on different videos

The results of the second rows testify the robustness of the method (as for the point 1) when the same environment and the same actor is used both for training and for testing. The first row reports results on another video with the same actor and the same camera (but a different video instance), while in the third row another actor is used and classified with the models trained with the other person. The system exibits a substantial robustness (about 90%) in every test where the model comes from the same environment. This will be a very common case in our application, where we can quietly suppose to have correspondence between training and run-time environment (and subject). Good matches came even where subject was different from training: see the last row of Table 2. We report also the errors that can occur when the camera is bad aligned. In fact the fourth and the fifth row report results on a video (Roberto 2, see for instance Figure 4(d)) where a person falls down in the same direction of the field of view. In these cases the shape in the 2D image plane is more similar to a crawling model than to a lying model. Thus the very low efficacy rate (54% and 67% only) is due to this high number of poses that cannot be classified from that point of view. Therefore this method will be enhanced in a multi-camera environment. In Table 3 we report the confusion matrix that is the number of wrong classification between the four states.

gt\class	0	1	2	3	4
0 - Not Classified	0	572	145	176	27
1 - Standing	8	2849	41	82	0
2 - Crawling	0	65	631	93	18
3 - Sitting	0	16	62	838	0
4 - Laying	0	63	56	23	217

Table 3: Confusion matrix from analyzed videos

.

As a final remark we would like to underline the speed for the process, since we are able to process always more than ten frames per second with a standard PC.

7. CONCLUSION AND ACKNOWLEDGES

The paper discusses initial results of detecting human posture for surveillance and behavior monitoring in domotic applications. The discussed approach proved to be reliable and robust if the working constraints are satisfied. Thus it can be considered a first step of a complete systems that will exploit a multi-camera system in order to provide the best point of view in a complex and cluttered environment as the domestic one. We would like to thanks the “Domotica per disabili” project that supported his research and the people of the Imagelab Staff for their valuable help.

REFERENCES

[1] H.Fujiyoshi, A.J.Lipton. *Real-Time Human Motion Analysis by Image Skeletonization* in Fourth IEEE Workshop on Applications of Computer Vision, 1998. WACV '98.

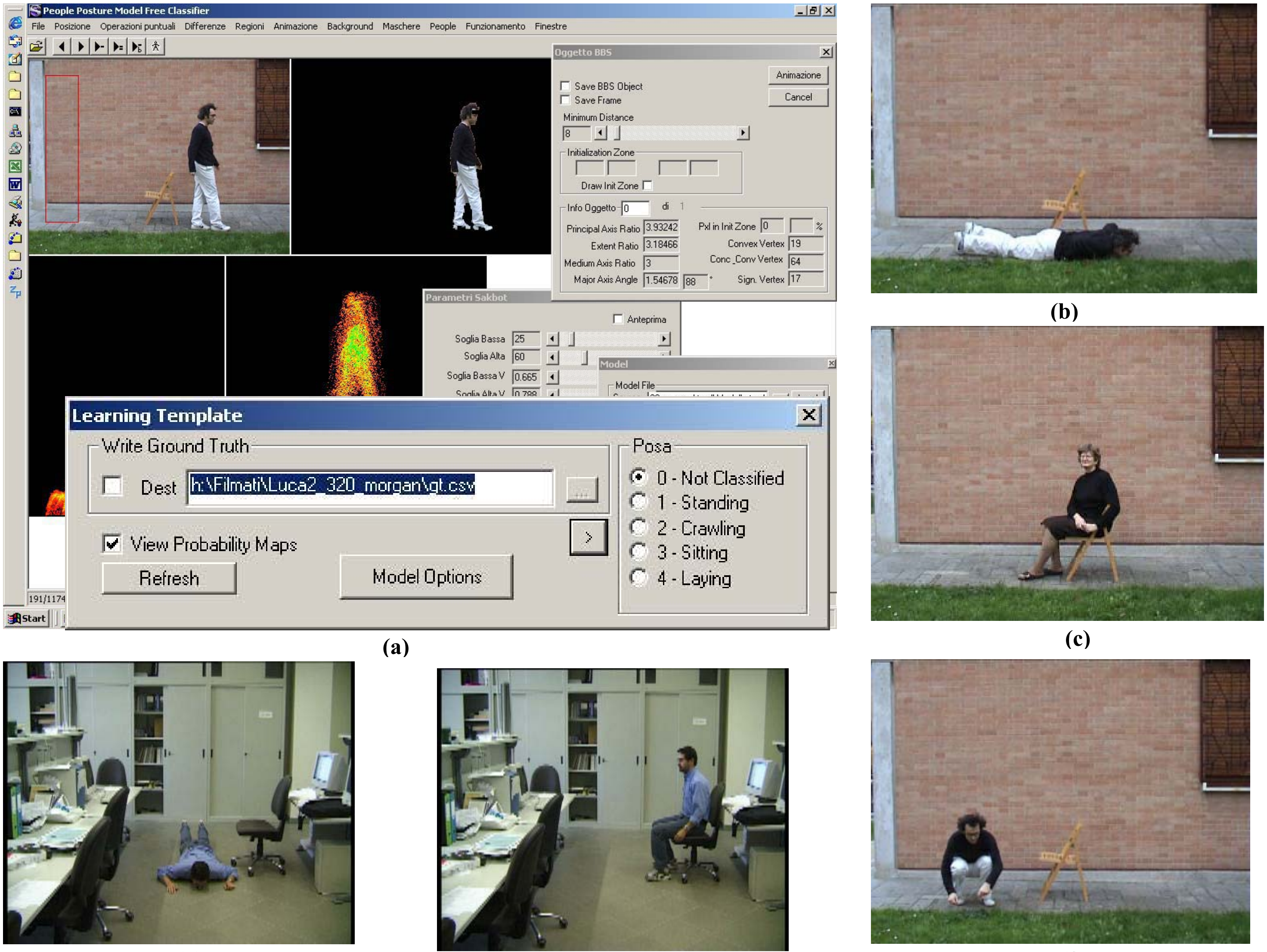
[2] I.Haritaoglu, D.Harwood, L.S.Davis. Ghost: *A Human Body Part Labeling System Using Silhouettes* in 14th Int. Conf. on Pattern Recognition, Brisbane, 1998

[3] I.Haritaoglu, D.Harwood, L.S.Davis *W^4: Real-Time Survelliance of People and Their Activities* IEEE Trans. On Pattern Analysis and Machine Intelligence,22(8),pp 809-830, Aug 2000

[4] S.X.Ju, M.J.Black, Y.Yacob *Cardoboard People: A Parameterized Model of Articulated Image Motion* in 2° International Conf. on Automatic Face & Gesture Recognition, 1996

[5] C.R.Wren, A.Azarbayejani, T.Darrel, A.P.Pentland *Pfinder: Real-Time Tracking of the Human Body* IEEE Trans. On Pattern Analysis and Machine Intelligence,19(7),pp 780-785, Jul 1997

[6] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, *Detecting Moving Objects, Ghosts and Shadows in Video Streams* in press on IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003


People Posture Model Free Classifier
File Posizione Operazioni puntuali Differenze Regioni Animazione Background Maschere People Funzionamento Finestre
Oggetto BBS
Animazione
Cancel
Parametri Sakbot
Model
Model File
Learning Template
Refresh
Model Options
Start

Oggetto BBS

□ Save BBS Object
□ Save Frame

Minimum Distance 8

Initialization Zone

Draw Init Zone □

Info Oggetto 0 di 1

Principal Axis Ratio 3.93242
Extent Ratio 3.18466
Medium Axis Ratio 3
Major Axis Angle 1.54678 88 °
Pxl in Init Zone 0 %
Convex Vertex 19
Conc_Conv Vertex 64
Sign. Vertex 17

Parametri Sakbot

□ Anteprima

Soglia Bassa 25
Soglia Alta 60
Soglia Bassa V 0.665
Soglia Alta V 0.788

Learning Template

Write Ground Truth

□ Dest h:\Filmati\Luca2_320_morgan\gt.csv

☑ View Probability Maps

Posa

● 0 - Not Classified
○ 1 - Standing
○ 2 - Crawling
○ 3 - Sitting
○ 4 - Laying

191/1174

(a)

(b)

(c)

(d)

(e)

(f)

Figure 3:
Some snapshot from the Framework:
(a) Machine Learning environment
(b,c,d,e,f) Frames from video test set