Signal Processing: *Image Communication* 23 (2008) 769–787

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Learning articulated appearance models for tracking humans: A spectral graph matching approach

Nicolas Thome [a,b,*,1], Djamel Merad [b], Serge Miguet [b]

[a] *Laboratoire d'InfoRmatique en Images et Systèmes d'information, Université Lumière Lyon 2, CNRS 5, Avenue Pierre Mendès France, 69576 Bron Cedex, France*
[b] *Laboratoire d'Informatique de Paris 6, Université Pierre & Marie Curie, CNRS 104, Avenue du President Kennedy, 75016 Paris, France*

ARTICLE INFO

Article history:
Received 11 March 2008
Received in revised form
20 September 2008
Accepted 22 September 2008

keywords:
Real-time multiple people tracking
On-line articulated appearance learning
People identification
Body part labeling from silhouette
Spectral graph matching
Topological model

ABSTRACT

Tracking an unspecified number of people in real-time is one of the most challenging tasks in computer vision. In this paper, we propose an original method to achieve this goal, based on the construction of a 2D human appearance model. The general framework, which is a region-based tracking approach, is applicable to any type of object. We show how to specialize the method for taking advantage of the structural properties of the human body. We segment its visible parts by using a skeletal graph matching strategy inspired by the shock graphs. Only morphological and topological information is encoded in the model graph, making the approach independent of the pose of the person, the viewpoint, the geometry or the appearance of the limbs. The limbs labeling makes it possible to build and update an appearance model for each body part. The resulting discriminative feature, that we denote as an *articulated appearance model*, captures both color, texture and shape properties of the different limbs. It is used to identify people in complex situations (occlusion, field of view exit, etc.), and maintain the tracking. The model to image matching has proved to be much more robust and better-founded than with existing global appearance descriptors, specifically when dealing with highly deformable objects such as humans. The only assumption for the recognition is the approximate viewpoint correspondence between the different models during the matching process. The method does not make use of skin color detection, which allows us to perform tracking under any viewpoint. Occlusions can be detected by the generic part of the algorithm, and the tracking is performed in such cases by means of a particle filter. Several results in complex situations prove the capacity of the algorithm to learn people appearance in unspecified poses and viewpoints, and its efficiency for tracking multiple humans in real-time using the specific updated descriptors. Finally, the model provides an important clue for further human motion analysis process.

* Corresponding author at: Laboratoire d'Informatique de Paris 6, Université Pierre & Marie Curie, CNRS 104, Avenue du President Kennedy, 75016 Paris, France. Tel.: +33 1 44 27 87 50.
E-mail addresses: nicolas.thome@lip6.fr (N. Thome), djamel.merad@liris.cnrs.fr (D. Merad), serge.miguet@liris.cnrs.fr (S. Miguet).

[1] This work has been supported by the SAS Foxstream.

0923-5965/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2008.09.003

1. Introduction

Human motion analysis is currently one of the most active research fields in computer vision. It attempts to detect, track and identify people, and more generally, to interpret human behaviors, from image sequences involving humans. It has attracted great interest from computer vision researchers due to its promising applications in many areas such as visual surveillance, perceptual user interface, content-based image storage and retrieval, video conferencing, athletic performance analysis, virtual reality, etc.

Event detection is the highest level for applications aiming at achieving a semantic description of the scene. Basically, low-level steps correspond to object detection and/or classification. Tracking humans can thus be understood as the intermediate level of the analysis process. Its main challenge consists in facing the data association problem, i.e. temporally linking features chosen to analyze human behaviors. Tracking is thus of primary importance since it has to fill the semantic gap between the brute image descriptors and the high-level concepts.

Thus, intensive attention has been focused in the last 20 years to providing robust tracking algorithms able to manage a wide variety of scenarios. The major challenges encountered in real situations include cluttered background, noise, change in illumination, occlusion and scale/appearance change of the objects. This paper addresses the tracking problem by an original approach where human detection, body part labeling, appearance modeling and people identification are interleaved. Moreover, we concentrate on tracking approaches able to run in near real-time, in the context of monocular sequences, and with static and non-calibrated cameras.

1.1. Paper contribution

In order to track multiple people in real-time, we propose an approach that relies on learning a specific descriptor for each person detected in the sequence. This feature is then used for identifying people in complex situations, and maintaining the tracking. The main goal of this work consists in building an *articulated appearance model* (AAM) that is specifically adapted to learn the appearance of humans. To achieve this aim, our contribution is twofold.

Firstly, we propose an efficient strategy for labeling limbs from each extracted blob. This step is formulated as a graph matching problem, where we make use of a *topological model* of the human body to detect and label each visible body part. The proposed approach encodes the silhouette properties by means of the graph structure in the most compact form. Indeed, the feature extraction only relies on shape and topology, and is thus applicable for any viewpoint/human pose. Importantly, the limbs labeling approach makes it possible to classify the tracked object between human/non-human, i.e. the graph matching output is used as a people detector.

Secondly, the AAM encodes color, texture and shape properties related to each tracked person. Once learned, these features are used for identification. Importantly, the appearance model learning is performed for each rigid part of the articulated structure. Thus, the formed descriptor capitalizes on the strength of the existing features in terms of discriminability and robustness. Finally, this appearance-based part of the tracking approach includes a solution for tracking people during occlusions that is carried out by means of an adapted particle filtering technique.

To summarize, we claim that the proposed approach can manage to learn human appearance in very generic configurations, and provides a people-specific descriptor adapted to the representation and the recognition of humans. The remainder of the paper is presented as follows. Section 2 gives a state of the art of the existing approaches for detecting and tracking humans, and Section 2.3 points out the specific shortcomings of the existing human appearance descriptors that are addressed in the paper. Section 3 proposes a brief overview of the overall approach, while Section 4 details the limbs labeling approach, and Section 5 explains how the AAM is generated and used for recognition. Section 6 presents results illustrating the approach efficiency for tracking humans in various conditions, and proves that the AAM outperforms both global templates and color histograms for recognition. Finally, Section 7 concludes the paper and proposes direction for future works.

2. State of the art

Detecting, tracking humans and inferring their pose in videos is arguably one of the most challenging problems in computer vision due to large variations in body shape, appearance, clothing, illumination and background clutter.

2.1. Existing approaches for human detection

We can classify approaches for detecting humans between global methods and part-based strategies. Global methods [36,18,51,52,10,11] try to detect humans with a single template, essentially relying on a sliding window mechanism. Gavrila and Philomen [18] extract edge images and match them to a set of learned exemplars using chamfer distance. Dalal and Triggs [10] introduce the Histogram of Oriented Gradient (HOG) descriptor, learned with a linear kernel SVM, that constitutes the state of the art for pedestrian detector. In addition, many authors combine motion features to the static edge features, most of the time using optical flow [52,11]. Other motion-based approaches firstly use a foreground pixel detector, and classify the extracted blobs as human/non-human using a given shape descriptor, e.g. projection histograms [20]. Realizing the difficulty to capture pose and viewpoint variability with a single flat classifier, other approaches rely on individually detecting body part candidates, and modeling their assembly [31,14,37,35,39]. Top–down approaches estimate the whole human position and the part location by modeling the limb likelihood and their relative position jointly [14,37]. Other

approaches adopt a two-stage strategy: a bottom–up detector is applied on the image to extract candidate parts, then a top–down procedure inference the configuration and finds the best assembly [31,35,39]. Most approaches use the canonical tree model for body parts, hence efficiently solving the assembly problem with dynamic programming.

2.2. General tracking problem formulation

Roughly speaking, tracking can be regarded as the problem of inferring the configuration of an object at time t, given its previous position at time $t-1$. It generally consists in a two-step prediction/correction scheme. Firstly, a model of the dynamic is applied to explore the parameters space. Secondly, the object configuration is updated over time with respect to a given image similarity measurement. Formally, let us denote $X(t) \in \mathbb{R}^N = (\theta_1, \theta_2, \ldots, \theta_N)^{\mathrm{T}}$ the state parametric form of the object at time t, the tracking task can be formulated as an inference problem, where one aims at estimating the state $\widetilde{X}(t)$ at time t best explaining the image measurement $(Z_t, Z_{t-1}, \ldots, Z_1)$ up to time t, i.e. solving

$$\widetilde{X}(t) = \underset{(\theta_1,\ldots,\theta_N)\in\mathbb{R}^N}{arg\,\max}[X(t) = x_t|(Z_t, Z_{t-1}, \ldots, Z_1) \quad (1)$$

Numerous approaches for tracking humans in video sequences have been proposed in the last 20 years. An exhaustive review of the existing strategies is beyond the scope of the paper and the reader can refer to [17,32,55]. Regarding methodology, we can distinguish the methods with respect to the following criteria: search strategies (in relation with the data association problem and thus with the detection), the chosen feature, and the use of a model.

When tracking multiple people, a crucial decision consists in choosing which image measurement to account for a particular human. In the radar tracking literature, this problem is often referred to *data association*. It can be performed by using a brute-force top/down search, i.e. explicitly exploring the parameter space next to the previous location. Alternatively, bottom/up detectors (such as those presented in Section 2.1) can be used to guide the data association process, identifying some particular region of interest. In both cases, the data association can be achieved with two different *search strategies*. In deterministic search strategies, the correspondence problem is solved by choosing the nearest neighbor approach. For example, the Kalman filtering [25] is a recursive linear estimator that is optimal in minimizing the covariance error, provided the conditional observation distributions are Gaussian. However, the Gaussian assumption is violated in many situations when tracking objects in image sequences, due to clutter or occlusions, to name but a few. There are several statistical data association techniques which tackle this problem. Particle filters are approximate stochastic sampling strategies that can explicitly deal with multi-modal distributions. For that reason, they have intensively been used since the pioneer work of Isard and Blake [21].

The chosen image feature that defines the similarity measurement is another major aspect of the tracking algorithms. *Appearance-based methods* track connected regions that roughly correspond to the 2D visual aspect of video objects based on their dynamic models. Appearance refer to color, shape and texture, and related approaches are generally known as *visual tracking* strategies. *Motion-based approaches* depend on a robust method for grouping visual motion consistently over time. *Feature-based* methods mainly rely on matching various image descriptors from frame to frame, and clustering techniques are used for detection and tracking. Finally, many approaches perform people detection using background subtraction, particularly in surveillance applications. This motion segmentation step is then often directly used for performing data association, i.e. linking the different connected component over time [20,57,19]. These strategies are appealing since they provide a first solution to the multiple people tracking issue, and are compatible with real-time purpose.

Apart from the chosen feature, approaches for tracking humans can be classified depending on the use of a model. *Model-based approaches* try to impose high-level semantic constraints by exploiting the *a priori* knowledge about the object being tracked. 3D models are by essence viewpoints independent, and 3D tracking approaches have been investigated by directly minimizing an image to model measurement (generative approaches) [46], or by learning the features to pose mapping from exemplars (discriminative approaches) [34]. However, 3D approaches are mostly not able to achieve real-time, and mainly require manual initialization. Many methods modeling the body part assembly with 2D models have been proposed, for example cardboard models or pictorial structures [14,37]. Model-Based approaches present important benefits when tracking in complex situation. As they explicitly model the body part structure, they are better armed to deal with occlusions than model-free approaches.

Importantly, the most advanced approaches try to combine advantages of the previous classes, leading to hybrid methods. For example, Ramanan et al. [37] propose an approach using a spatio-temporal pictorial model, using motion, color and shape features for performing the inference task with a partitioned sampling strategy. Interestingly, the proposed approach is connected to ours by the fact that the algorithm is firstly dedicated to learning people appearance for providing a discriminative person specific model. However, the approach tries to take advantage of some particular configurations such as lateral walking poses for learning appearance, making the tracking efficiency limited by the availability of such configurations. Moreover, the algorithm requires between 7 and 10 s processing per frame. We now give more details on some recent works being the most related to ours, i.e. approaches which explicitly model people appearance for performing a real-time tracking in a monocular context.

2.3. Real-time, appearance-based tracking approaches

Appearance is among the most stable features when tracking humans, and therefore makes this descriptor

appealing for tracking algorithms. Basically, we can distinguish approaches modeling appearance using probability density functions (PDF) [15,4,7,8,23,22,54,28,49], and methods using templates [3,42,20,57].

Wren et al. [54] use small blob features statistics (position and color) to track people. They represent human shape as a combination of blobs representing various body parts, such as head, torso, hands and legs. Each blob is modeled by a multi-normal distribution capturing position and color in the YUV space. Tracking all the small blobs allows them to track the whole body of a single human. However, their work is only adapted to an indoor environment and is only intended to track a single human, which is too limited for our purpose. McKenna et al. [28] propose an adaptive background removal algorithm that combines gradient information and color features to deal with shadows in motion segmentation. They differentiate three levels of tracking: regions, single human and human groups. They manage to obtain good results of tracking multiple persons even in the case of occlusions by introducing an appearance model based on a combination of color histogram and Gaussian mixtures. However, the PDF modeling in both previous approaches makes the spatial information to be completely ignored during the feature extraction step. For example, a person wearing a yellow tee-shirt and blue pants is represented in the same manner as a person wearing a blue tee-shirt and yellow pants.

Alternatively to statistical representations, appearance can be modeled using templates. Thus, Haritaoglu et al. [20] develop a system named W4 for a real-time visual surveillance system operating on monocular grayscale or on infrared video sequences. W4 makes no use of color cues, but employs a combination of shape analysis and tracking to locate people and their body parts. Moreover, the method explicitly uses a template to identify people after an occlusion. The appearance model represents people globally, and the major difficulty in using such a template for matching corresponds to the way the appearance is updated over time to provide a relevant descriptor. In [20], each image is registered to the previously built template with respect to its median coordinate. This strategy is not adapted to highly deformable objects such as humans, and the update commonly fails at being properly carried out in typical situations illustrated in Fig. 1. In Fig. 1(a), we point out the fact that the model is very sensitive to segmentation errors or partial occlusions of the limb. The two people enter the scene at frame 100 and the models are initialized. At frame 200, we can notice that their appearance models have been wrongly updated. For the red-framed person, it is due to segmentation errors. The legs were indeed not detected at frame 100 and are extracted later: as the model is registered with respect to the median coordinate, the different body part appearance update is not properly performed. For the green-framed person, the legs were not detected at the beginning because hidden by the desk. This leads to the same difficulties. Fig. 1(b) illustrates problems to manage strong perspective effects. Indeed, the person is walking in the optical axis direction, making its apparent size in the image sensitively vary from frame 100 to 250. Again, this leads to badly update its corresponding appearance model: for instance, the appearance of the head detected at frame 250 is averaged with some part of previously detected torso at frame 100.

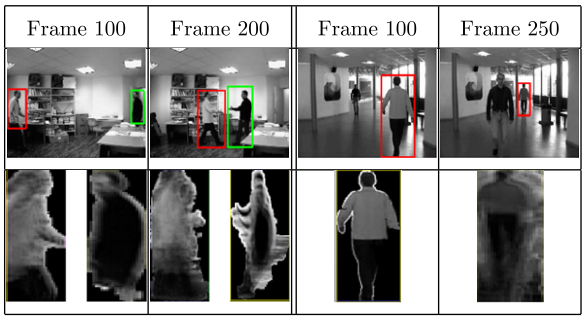

Fig. 1. Global appearance models limitations. (a) Segmentation errors and occlusions. (b) Strong perspective effects.

Zhao et al. [57] build upon the work of Haritaoglu and suggest a much finer system for tracking. They use an ellipsoid model for the modeling of the 3D human shape and track its parameters with a Kalman filter. They make use of an appearance model that integrates a color clue. The mask of the model is an ellipse instead of a rectangle but this model still suffers from the former drawbacks.

3. The proposed approach

3.1. Approach overview

We propose a hybrid approach, schemed in Fig. 2, that can manage the tracking of multiple people in complex situations. At the pixel level, people are detected by a background subtraction algorithm, and a data association step links connected moving regions over time. These low-level steps, described in Section 3.2, are applicable to any type of object. They are devoted to distinguishing 'easy situations' from other. Thus, non-ambiguous blob associations (i.e. one-to-one) are capitalized on to learn people appearance by building AAMs. The discriminative generated features are then used to disambiguate the blob tracking in complex scenario (occlusion, people leaving the field of view, etc.) by identifying people, i.e. matching appearance models.

3.2. Generic object system components

In this section, we give more details on how human detection and data association is performed.

3.2.1. People segmentation

For isolating people in the video, we firstly compute a binary map describing regions where motion occurs by computing a difference between the current frame and a reference one representing the static part of the scene. This step involves building and updating the background image. Many approaches have been proposed for estimat-

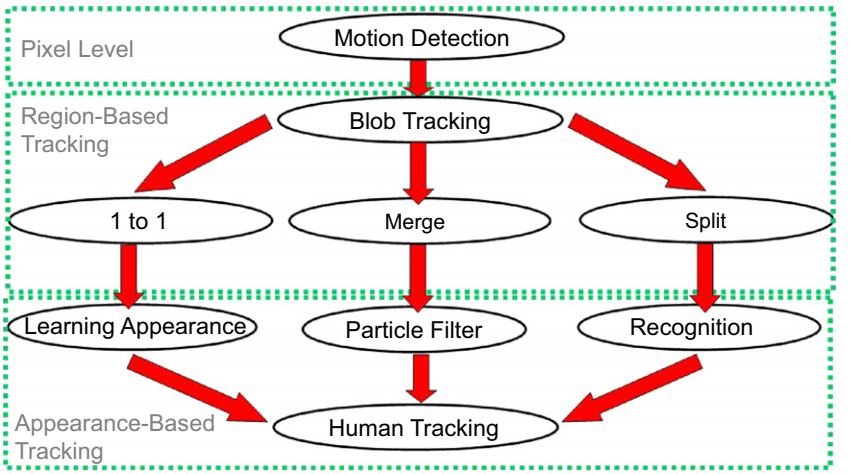

Fig. 2. General tracking framework.

ing the PDF of the gray-level (or color) value for each pixel [47,33,12]. For optimizing the trade-off between computational complexity and segmentation accuracy, we choose to model each pixel PDF with a mixture of K_G Gaussians, using an adapted version of the Stauffer and Grimson algorithm [47]:

$$P_r(X_t) = \sum_{i=1}^{K_G} w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (2)$$

where

- $w_{i,t}$ represents the weight of each distribution and represents the portion of the data accounted for by this Gaussian.
- $\eta(X_t, \mu_{i,t}, \Sigma_{i,t})$ is a normal distribution with mean $\mu_{i,t}$ and covariance matrix $\Sigma_{i,t}$. For a computational purpose, we choose to model each distribution with a diagonal covariance matrix which assumes that the red, blue and green pixel values are independent. But contrary to [47,33] we do not insist that they have identical variances.

The main adaptation compared to the original approach [47] consists in updating on-line the number of required distributions for explaining the pixel history. Following Carminati and Benois-Pineau idea [5], we choose to automatically estimate the required number of distributions using an ISODATA algorithm [2]. To ignore shadows during the segmentation stage, we detect them at the pixel level by using the color features $c_1c_2c_3$ proposed by Salvador [41], which has the property of being invariant in luminance. For more details on this part, the reader is referred to [50].

At this stage, the motion segmentation outputs a binary map where moving pixels have been isolated, and no consideration of spatial coherence has been developed. Morphological operations are applied in that sense. Finally, a connected components labeling algorithm is used to merge pixels into regions. We only keep the significant ones by thresholding their area.

3.2.2. Blob tracking

A simple and computationally efficient blob tracker is used to *dynamically link extracted regions* over time. We used a two-pass forward/backward strategy inspired from [54,20,19]. It aims at matching the M objects O_i detected at time t to the N regions R_j extracted at time $t+1$. We use a simple first-order motion model to predict object locations O_i' at time $t+1$. Then, for each O_i' and R_j, we compute a $M \times N$ similarity matrix containing the set of area overlap $A_{ij} = O_i' \cap R_j$. For each R_j we determine its predecessors number np_j and sort them with respect to their A_{ij} value (the most probable predecessor is supposed to be the one with the greatest overlap surface). We obtain a list of predecessors $P_k, k \in \{1; np_j\}$. Identically, for each projected O_i' we determine its successors number ns_i and sort them with respect to their A_{ij} value, and obtain a list of successors $S_l, l \in \{1; ns_i\}$.

The algorithm output a set of five possible links that can be exhaustively enumerated:

(i) If $np_j = 0$ a new object is created.
(ii) If $ns_i = 0$ the previously tracked object will be ignored by the blob tracker at the next times steps. In addition, it might be removed, if it is not *active* (see the following remarks regarding temporal coherence). Otherwise, the object is kept because it may be identified later on, based on its learned appearance model.
(iii) If $np_j = 1$ and $ns_i = 1$, the correspondence is one-to-one. This is the easiest case, where a single object can successfully be tracked by the region linking (see Fig. 2). Thus, we take advantage of these situations to label body parts (Section 4) and learn people appearance (Section 5.1).
(iv) If $np_j = 1$ and $ns_i > 1$ a split is detected: a single region that was previously tracked is now detected as several blobs (see Fig. 2). In that case, we use the previously learned appearance models for identifying people and maintaining a robust tracking (see Section 5.2).

(v) If $n_{pj}>1$ and $ns_i=1$ a merge is detected: several blobs that were previously individually tracked are now only detected as a single region (see Fig. 2). This suggests that an occlusion is occurring, and we use a particle filter to manage this arduous situation (Section 5.3).

It is worth mentioning that the blob tracker takes advantage of *temporal coherence* for making the region association robust to noise. Thus, we use an approach connected to the Multiple HypoThesis Tracking strategy [38], i.e. we only update links that are supported over time. In particular, a blob is denoted as *active* if is has continuously been matched one-to-one for a sufficient number of frames N_{coh}. Then, the split end merge association are only considered if they are related to *active* objects, heavily decreasing the segmentation errors impact.

4. Body part labeling

Each time a region is one-to-one linked over time by the algorithm described in Section 3.2.2, a morphological and topological analysis of the silhouette is carried out to detect and label the visible body parts. To achieve this goal, we propose the following strategy, illustrated in Fig. 3. Firstly, we extract a set of segments from the silhouette (Section 4.1). Secondly, we form an image graph from this set of segments and a model graph, generated from a topological human skeleton model (Section 4.2). Finally, we perform a node-to-node matching of the image and model graphs (Section 4.3).

4.1. Getting a set of segments

For each 'single human' detected region, the limbs are considered as parts of the silhouette skeleton.

4.1.1. Skeleton computing

The skeletonization process is an important stage in our application. Several 2D skeletonization algorithms have been reported in the literature, and they can be classified into two categories: the discrete methods [27,26], and the continuous methods, mainly based on the Voronoi diagram computation [1,13]. The latter approaches have several advantages in our context. Only the contour points are necessary, which considerably decreases the number of points to process. The obtained skeleton and the initial shape are topologically equivalent. And last, but not least, the obtained skeleton is isomorph to a graph, which represents an important advantage regarding the chosen representation for the recognition (see Section 4.2). For these reasons, we use this continuous strategy for computing the 2D skeleton of the shape. However, the Voronoi diagram computation remains relatively sensitive to the noise on the contour shape. To overcome that shortcoming, we propose a pre-processing step devoted to smoothing the initial shape, by computing the Fourier descriptors (FD) [56] of its outer and (potential) inner contours.

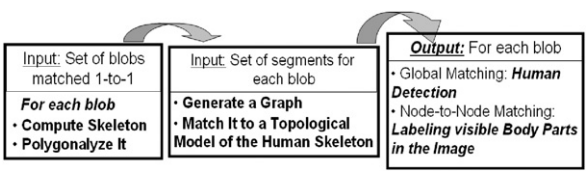

Fig. 3. Overview of the limbs labeling approach.

4.1.2. Skeleton polygonalization

The 2D skeletonization being performed, we aim at extracting a set of N segments from the silhouette, i.e. identifying $N+1$ extreme points. Each skeleton point corresponds to the center of the Delaunay triangle circumscribed circle, and we take advantage of this neighborhood information encoded in the data structure. Thus, we can classify the skeleton points into two clusters, that we denote as points of types 1 and 2. The formers contain ending or branching points, i.e. points having either a single neighbor or more than two neighbors. Points of type 2 correspond to those having exactly two neighbors. The set of segment extremities that we want to detect are thus composed of points of type 1. In addition, we polygonalyze the discrete curve formed by the skeleton points between each couple of type 1 points. Let us consider the discrete curve $\mathscr{C}$, containing a sequence of K points, and let us denote P_0 its starting point. We define a criterion $\gamma(i)$, quantifying the non-linearity for the ist point of the sequence:

$$\gamma(i)=\frac{1}{i}\sum_{k=1}^{i}\overline{P_kH(P_k)} \tag{3}$$

where

- $H(P_k)$ corresponds to the orthogonal projection of the kst point on the line (P_0,P_i) (see Fig. 4(a)).
- $\overline{P_kH(P_k)}$ is the signed distance from the point P_k to the line (P_0,P_i). The distance is thus (arbitrarily) positively counted if the point lies on a given side of the segment $[P_0,P_i]$, and negatively otherwise (see Fig. 4(a)).

The proposed polygonalization scheme consists in computing a discrete approximation of the curvilign integral of the curve $\mathscr{C}$. For each point P_i of a given skeleton branch, we match $|\gamma(i)|$ against a given threshold γ_S. If $|\gamma(i)|>\gamma_S$, we identify the skeleton point with the maximum $|\overline{P_kH(P_k)}|$ value (see Fig. 4(b)). The segment $[P_0;P_{k_{\max}}]$ is thus extracted, and the algorithm is recursively run from $P_{k_{\max}}$ (see Fig. 4(c)). The proposed polygonalization approach presents two main advantages for our purpose. Firstly, the signed distance used in Eq. (3) is robust to 'oscillating curves', that are common due to the silhouette sampling before the skeleton computing. Secondly, the algorithm is based on the points of type 1, that are robust and accurate as resulting from the Voronoi diagram computation.

Fig. 5 illustrates the segment extraction scheme, processed at each frame. Fig. 5(a) represents an extracted silhouette output by the background subtraction step

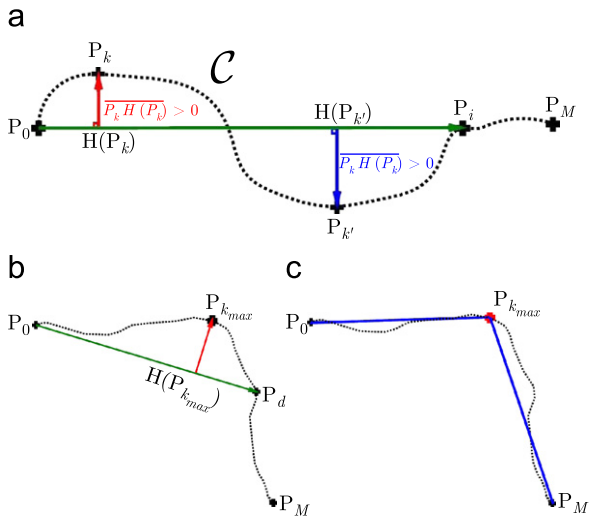

Fig. 4. Skeleton polygonalization. (a) Defining a linearity criterion. (b) Identifying breaking points. (c) Generated segments.

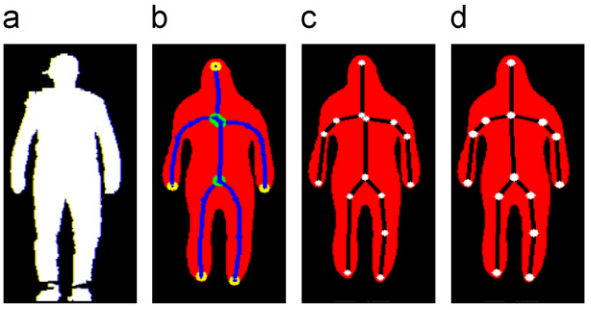

Fig. 5. Segments extraction from silhouette.

(Section 3.2.1), Fig. 5(b) illustrates the silhouette smoothing and the skeleton computation. Fig. 5(c) shows the first set of segments after the polygonalization and the Fig. 5(d) presents the final set after removing small edges.

4.2. Skeleton model and graph generation

The segments being extracted from the silhouette, as illustrated in Fig. 5(d), are supposed to be limbs that we intend to label. To perform this task, we make use of a skeleton model, illustrated in Fig. 6(a). We thus model 14 rigid parts of the body: torso (blue), head (yellow), arms and shoulders (green), legs and ankles(red). The skeletal model can be considered as a 3D model, as we consider the connections between limbs in the 3D world. Indeed, the 2D connections in a given image are often incomplete due to occlusions. However, we only consider connections between limbs, and the skeletal structure must be regarded as a *topological model*.

4.2.1. Making use of a graphical model

Graphical models are widely used for representing objects and dealing with structural information. For our limbs labeling purpose, the graph representation is particularly adapted to model the relationship between the different body parts. Thus, we generate the graphical model illustrated in Fig. 6(b) to encode the topological structure of the skeletal model presented in Fig. 6(a). The graph is actually a tree, rooted at the bottom of the torso segment (node 0 in Fig. 6). For the sake of clarity, we will herein always refer to the graphs with the vocabulary of nodes and arcs, and to the set of segments with the vocabulary of edges and vertices.

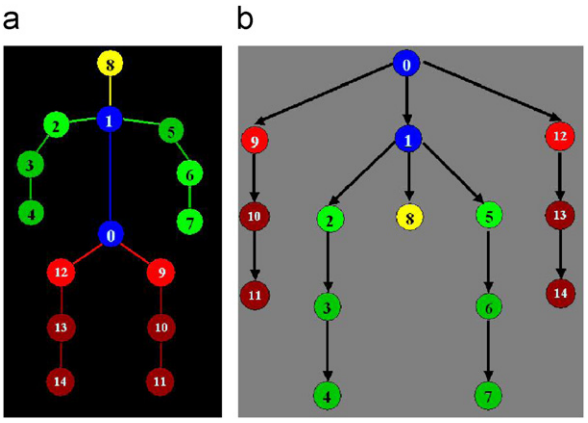

Fig. 6. Topological models used. (a) 3D skeleton model. (b) Model graph.

4.2.2. Generating the image graph

To generate a graph from the set of segments extracted from the silhouette, we have to map each segment vertex to a graph node, in order to match the model and image graphs (Section 4.3). The first issue corresponds thus to identifying the vertex of the set of segments that corresponds to the root of the graph.

4.2.2.1. Root node identification. To achieve that goal, we associate to each inserted segment its mean radius R_m:

$$R_m = \frac{1}{K}\sum_{i=1}^{K} r_i \tag{4}$$

where r_i corresponds to the radius of the circumscribed circle (of the Delaunay triangle) for each of the K skeleton points corresponding to the segment. From the R_m definition (Eq. (4)), we measure the mean distance of the segment to the silhouette boundary. Thus, R_m enables us to identify the torso as the segment with the largest mean radius. As far as we tested, this image feature is very efficient for localizing the segment corresponding to the torso, and is robust to viewpoint and human pose variations. As we want to generate a graph with nodes corresponding to segment vertices, an ambiguity between two nodes N_i remains after the torso location. To properly identify the vertex corresponding to the root, we generate the two possible trees G_i, rooted at the torso N_i ($i \in \{1;2\}$). We then compute the distance $D(G_i, G_M)$ between the two graphs and the model graph

G_M, in the following way:

$$D(G_i, G_M) = w_{T_o} * D_{Top}(G_i, G_M) + w_{T_r} * D_{Track}(G_i)$$
$$D_{Top}(G_i, G_M) = \sqrt{\sum_{j=1}^{\Delta} [\chi(N_i(j)) - \chi(G_{M0}(j))]^2}$$
$$D_{Track}(G_i) = \sqrt{[N^x_{i,t} - N^x_{r,t-1}]^2 + [N^y_{i,t} - N^y_{r,t-1}]^2} \quad (5)$$

The distance $D(G_i, G_M)$ contains two terms D_{Top} and D_{Track}, weighted by w_{T_o} and w_{T_r}. D_{Top} only uses the structural information of the graphs to estimate the closest image graph to the model. This similarity measurement is the heart of the proposed graph matching approach, and is detailed in Section 4.3.1.1. Basically, it corresponds to computing the Euclidean distance between the topological signature vector (TSV) of the two roots. Fig. 7 illustrates the root identification. The maximal mean radius segment detection, leading to find the torso, is presented in Fig. 7(a). The two possible graphs resulting from rooting the image graph at the two possible torso vertices are shown in Figs. 7(b) and (c), respectively. The root TSV computation is presented in Fig. 7(d), making it possible to compute the topological distance with respect to the model $D_{Top}(G_i, G_M)$ for each graph (Fig. 7(e)). As we can notice in this example, the chosen root for the image graph corresponds to the node 0.

The tracking distance, D_{Track}, is only evaluated if the limb tracking is activated (see Section 4.3.2.3.3). In that eventuality, we determine the geometrical Euclidean distance, in the image, between each candidate $N_{i,t}$ for being the image root at time t and the previously labeled root $N_{r,t-1}$ at time $t-1$ (see Eq. (5)).

4.2.2.2. Managing inner contours. The extracted set of segments after the skeletonization might be cyclic if inner contours exist in the silhouette. We recall that we aim at generating a graph with a tree structure, i.e. an acyclic graph. If a loop is detected during the graph generation process, we determine the arcs to remove by the following reasoning. If the cycle contains the arc corresponding to the segment detected as the torso (as computed in Eq. (4)), we remove an arc linking one of the nodes corresponding to the torso vertices. The choice between the two possibilities is performed by minimizing the topological distance D_{Top} (Eq. (5)). This makes it possible to use the *a priori* structural information contained in the model, where no loop exists. If the cycle does not contain the torso, we choose to remove the arc corresponding to the minimal mean radius in the loop. The proposed approach for generating a tree from the loopy graph is similar to the minimal spanning tree algorithm (see [40]), with the weight corresponding to the invert of the mean radius. However, our method is much faster as it is local, and requires a single minimization for each cycle.

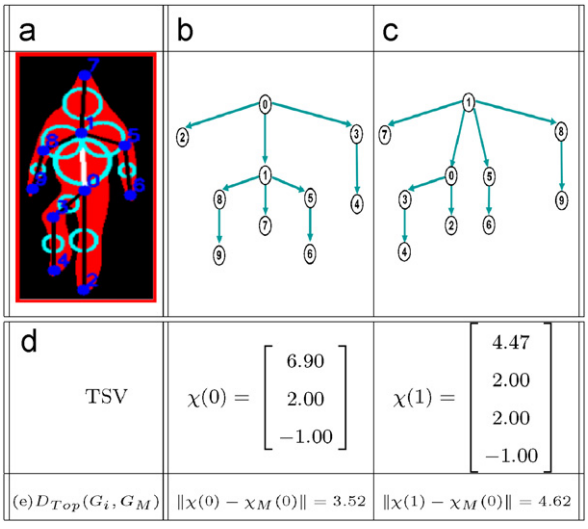

Fig. 7. Root identification in the image graph.

4.2.2.3. Performing human detection. A very interesting property of the image graph generation corresponds to its capacity to behave as a human detector. Until now, the approach is applicable to any kind of object. As we want to track humans and build an appearance model for them, an essential step consists in initializing the model, i.e. being able to classify the detected object as human/non-human. The topological distance $D_{Top}(G^r_I, G^r_M)$ between the root of the image graph G^r_I and the model graph G^r_M, as defined in Eq. (5), is used to perform that task. Indeed, computing the Euclidean distance between the root's TSV of two given graphs makes it possible to compute a global similarity measurement between their associated skeletons, and has been applied in the shape indexing context [43,44]. Therefore, thresholding the topological distance $D_{Top}(G^r_I, G^r_M)$ between the image graph and the model graph is a powerful way to build a human detector. Thus, if $D_{Top}(G^r_I, G^r_M)$ is below a predefined threshold D^T_{Top}, the detected object is recognized as a human, and we use the graph matching strategy proposed in Section 4.3 to label the different nodes of the image graph. Otherwise, the process stops at this stage. That is, there is no attempt at labeling the silhouette segments, and no AAM for the object is built at this time step. In order to take advantage of temporal coherence, and to make the parameter D^T_{Top} setting related to the tracking step, D^T_{Top} is made time dependent, i.e. we multiply it by the factor $e^{-N_T t}$ (N_T being the number of frames that the blob has been tracked).

4.3. Graph matching

Once the image graph is generated, we aim at performing a node-to-node matching with respect to the model graph in order to identify the visible body parts of the people present in the scene. There is a plethora of graph matching strategies present in the literature. An exhaustive review of the topic is far beyond the scope of the paper, and the reader can refer to [6,29,24]. For our purpose, we can classify the approaches regarding the chosen similarity measurement, and between exact/inexact matching strategies. In exact matching algorithms, the most used concept is the graph isomorphism and a many works are dedicated to searching for the best isomorphism between two graphs or sub-graphs.

However, in a number of cases, the bijective condition is too strong, and the graph isomorphism does not exist. Therefore, the problem is rather expressed as an inexact graph matching problem, leading to optimize some objective function measuring the adequacy between vertices and edges of both graphs. Approximate methods may be divided into two groups of algorithms. The first group is composed of methods which use spectral representations of adjacency matrices, while the second group is composed of algorithms which work directly with graph adjacency matrices, and typically involve a relaxation of the complex discrete optimization problem.

4.3.1. Spectral graph matching: the shock graphs

The shock graph method [45] is dedicated to indexing shape by skeletal graph matching and is thus closely connected to our purpose. Regarding methodology, the graph matching approach is an approximate method that makes use of the spectral representation of the adjacency matrices. Basically, the approach consists in encoding the graph structure in each node by a spectral characterization (Section 4.3.1.1), and in performing a node-to-node matching (Section 4.3.1.2).

4.3.1.1. Encoding graph structure. In the shock graph approach, a powerful way to encode the structure of a DAG consists in turning to the domain of spectral graph theory to compute a topological feature in each graph node. Let us consider a graph G of order L, and let us denote its nodes N_i ($i \in \{1;L\}$). The main property, established for trees in [45] and generalized to arbitrary DAGs in [44], states that it is possible to represent G by its adjacency matrix AG (of size $L \times L$), with 1's (-1's) indicating a forward (backward) edge between adjacent nodes in the graph, and 0's the absence of link. The graph spectrum $\Gamma(AG)$ is defined as the set of the N eigenvalues magnitudes of AG. A descriptor, called TSV and denoted $\chi(N_i)$, is thus extracted at each node N_i of the graph. Its computation processes as follows. Let us consider a node N_i of the graph G, having k children C_j ($j \in \{1;k\}$). For each child, we consider the sub-graph rooted at C_j, with order N_{C_j}. The adjacency matrix AG_i^j (of size $N_{C_j} \times N_{C_j}$) of this sub-graph is then diagonalized. We compute the sum of the eigenvalue magnitudes of AG_i^j. This value corresponds to the jst element of the TSV $\chi(N_i)$ for the node N_i. At the end of the computation $\chi(N_i)$ is sorted in decreasing order. Fig. 8(a) illustrates the computation of the TSV.

$\chi(N_i)$ is the feature vector that is used for performing the matching between the model and image graphs, by computing the Euclidean distance between each node TSVs. Thus, all $\chi(N_i)$'s must have the same dimension so that the distance computation is feasible. Thus, $X(N_i)$ is defined as a M-sized vector, M being the maximum between the maximum degrees of the image and model graphs. As the majority of the sub-graphs rooted at a given node N_i have a degree $<M$, we have to set a default value for elements that are not computed (i.e. from $k+1$ to M, k being the number of children of N_i). In [45,30], the TSV is thus padded with 0's. Alternatively, we prefer padding the TSV with the -1 value. We justify this choice in Section 4.3.2.3. Fig. 8(b) illustrates the result of the TSV computation for each node of the model graph presented in Fig. 6(b).

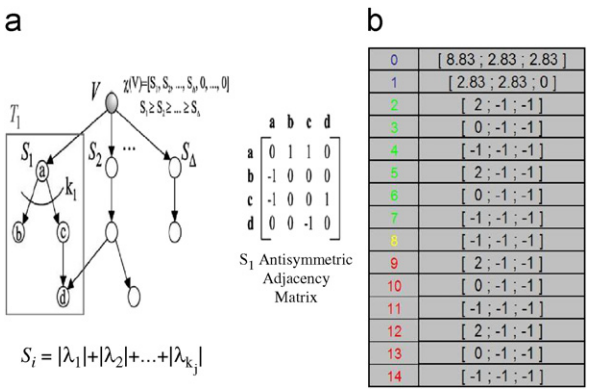

Fig. 8. Topological signature computation. (a) TSV computation, from [48]. (b) TSV model graph.

4.3.1.2. One-to-one nodes matching. Let us consider the model graph $G_M = (V_M, E_M)$, and the image graph $G_I = (V_M, E_M)$ to match, and let us denote $\delta(G_M)$ and $\delta(G_I)$ their maximal degrees. d is then defined as the maximum degree between the two graphs: $d = \max(\delta(G_M), \delta(G_I))$. For each node v_M (resp. v_I) of the graph G_M(resp. G_I), we define $\chi(v_M) \in R^d$ (resp. $\chi(v_I) \in R^d$) as the only vector which corresponds to the topological signature introduced in Section 4.3.1.1, and illustrated in Fig. 8.

Then, the matching algorithm starts by forming the matrix Π of size $n_1 \times n_2$, whose elements (v_M, v_I) are computed as follows: $\Pi(v_M, v_I) = \|\chi(v_M) - \chi(v_I)\|^2$. Then, the bipartite graph $\Gamma(V_M, V_I, E_\Gamma)$ is formed, where the arcs are weighted using the matrix $\Pi(G_M, G_I)$. Using the scaling algorithm of Goemans [16], the maximum cardinality, minimum weight matching in Γ is determined. The result is a set of node matching between G_M and G_I called M_1, which can be sorted in a descending order of similarity. From M_1, (v_M^1, v_I^1) is chosen as the pair which has the smallest weight among all the pairs in M_1. (v_M^1, v_I^1) is then removed from the list and added to the solution of the matchings. The same depth-first search strategy is recursively applied for the two sub-graphs rooted in v_M^1 and v_I^1, and it stops when one leaf of the two trees is reached. Then, a backtracking scheme starts, and the branches are dynamically recomputed: all the sub-trees of the graphs G_M and G_I whose roots have been matched are removed (reinforcing the one-to-one matching). The processing terminates when all nodes are in the solution set.

4.3.2. Shock graphs: discussion and adaptation

4.3.2.1. Spectral representation strengths. For our limbs labeling purpose, the shock graph matching strategy is a very powerful way to analyze the silhouette shape and match it against the skeleton model.

Firstly, we point out the fact that the TSV offers a hierarchical and fine description of the skeleton. For example, it makes it possible to discriminate graphs much

more accurately than using global statistical features, such as minimal/maximal/mean/standard deviation degree (see [44]). Secondly, the TSV strength corresponds to its robustness to shape variations. We recall that the TSV is computed from the adjacency matrix graph *AG*, and it is thus clear that the extracted feature only encodes the topological information of the graph. In particular, geometrical or appearance features are discarded. Moreover, some important theoretical properties have been established regarding the TSV. The interlacing theorem, established by Cvetkovi [9], states that the eigenvalues difference of the two graphs can be bounded, if one graph is a sub-graph of the other. This smooth variations of eigenvalues makes the skeleton matching robust to occlusions. This is an essential result for our purpose, because there must be missing nodes between the model graph (which encodes all connection between limbs) and the image graph (which corresponds to a viewpoint specific instance of the skeleton, with a particular joint configuration between limbs). Some other interesting properties have been established by Wilkinson [53], Stewart and Sun [48], and Shokoufandeh [44]. The reader is referred to their works for a detailed description of the study. For us, the main interesting results that must be pointed out correspond to the stability of the TSV under insertions or deletions of nodes (see [44]). In addition to the robustness to occlusions, this property makes it possible to manage insertions or suppressions of nodes that are likely to arise due to low-level step errors: background subtraction, skeleton computing, segment extraction, etc. As a result, we can state that the TSV constitutes a descriptor that captures discriminative features of the shape, without being sensitive to small perturbations.

Apart from the TSV strength, the node-to-node matching strategy proposed in the shock graph approach is very efficient in our limbs labeling context. Firstly, the depth-first search strategy, inspired by the Reyner algorithm [40], makes it possible to output a set of correspondences that respect the hierarchical order of the nodes, which is primordial for our application. The algorithm recursively finds matches between the nodes of the sub-trees, starting from the tree root and proceeding to the bottom in a downward direction. Secondly, the backtracking of the matching algorithm is much more efficient than in standard depth-first strategies, where this step is made statistically. Here, the algorithm dynamically recomputes the branches at each graph node, and chooses the following branch to go down by minimizing the TSV distance between the remaining nodes.

4.3.2.2. Approach limitations. There are, however, some important shortcomings when applying the shock graph matching strategy in our context of limbs labeling, mainly because the considered graphs have a small number of nodes. We can classify the approach limitations with respect to the three following criteria:

(i) Some nodes of the graphs are indistinguishable from their TSV while their sub-graphs have a different structure. There is thus a problem of *specificity*: when dealing with small graphs, the TSV is not able to discriminate between different structures.
(ii) The cost for matching to nodes when using the weighted matrix $\Pi(G,H)$ of the bipartite graph mainly relies on the Euclidean distance between TSVs. Indeed, the cardinality is only used to sort the correspondences with equal TSV distances. Intuitively, it would seem preferable to select large groups having small differences in terms of TSV than to select small groups with identical structure. In addition, the depth of the two matching nodes is not accounted for when forming correspondences. There is thus a problem of *hierarchical ordering*: the influence of the size of the matching nodes and their depth is marginally taken into account.
(iii) The method only provides one solution that is considered as the best at the current step of the algorithm. In case of ambiguity, the process is thus arbitrarily run from one (randomly) selected match.

4.3.2.3. Approach adaptations. To overcome the three shortcomings previously stated, we adapt our algorithm as follows.

4.3.2.3.1. TSV Padding. In the original approach [45], we recall that when the TSV is not computable, it is padded with zeros. However, this solution does not make it possible to discriminate terminal nodes *T* from nodes *BT* having an unspecified number of children that are terminals. Indeed, the 0 eigenvalue magnitude sum of a leaf node is indistinguishable from the padded 0 of the TSV. This is the origin of the lack of specificity of the descriptor for small sub-graphs. To overcome this shortcoming, the TSV is padded with the -1 value in our approach. Another negative value would also be possible (as the TSV elements are necessary positives). With our proposed encoding, the number of 0 values for a *BT* node makes it possible to determine its number of terminal nodes. Shokoufandeh et al. [44] already note this problem and suggest to add as extra dimension the eigenvalue magnitude of the root node sub-graph. In our context, we believe that this solution unnecessarily increases the dimension of the TSV. We think that our representation has the same specificity potential, and is more compact.

4.3.2.3.2. Bipartite graph matching. We propose a matching criterion that gives a stronger impact than in the initial approach to the size and the depth of the matching nodes. Thus, we compute the cost $CM(u,v)$ of a matching between two nodes u and v in the following way:

$$\begin{aligned} CM(u,v) &= \|\chi_u - \chi_v\| + P_{\text{Group}} + P_{\text{Depth}} \\ P_{\text{Group}} &= -\alpha * N_u \times N_v \\ P_{\text{Depth}} &= \beta * |D_u - D_v| \end{aligned} \tag{6}$$

Thus, matching two nodes u and v with the criterion defined in Eq. (6) depends on the topological distance between their respective TSVs $\|\chi_u - \chi_v\|$, on the number of nodes N_u and N_v of the sub-graphs rooted in u and v, and on the difference in depth $|D_u - D_v|$ between the nodes. P_{Group} is defined to overcome a shortcoming of the initial

approach when dealing with small graphs. Indeed, as the original algorithm looks for exact matching between TSVs, the algorithm tends to preferentially match small groups that do not contain much structural information. In the extreme, terminal nodes (i.e. leaves) do not contain any structural information: we can match a given terminal node of a given graph to any terminal node of another graph, so that the matching is meaningless. As we look for an inexact match between graphs (due to noise, viewpoint variations, occlusions, etc.), we claim that it is relevant to allow small variations between TSVs while supporting matchings between large groups. Regarding P_{Depth}, we can note that weighting the matching cost proportionally to the depth difference between two nodes has been proposed by the authors [45] as a direction for future works.

α and β are weights that determine the importance of each term in the similarity, that have been experimentally set up. Learning α and β from training data could be possible for more complicated applications, but we find it sufficient for our purpose. It must be pointed out that both terms P_{Group} and P_{Depth} must have a small weight compared to the topological distance between nodes' TSV. Indeed, the TSV is the feature that makes the algorithm robust to occlusions, and node insertions/deletions. Therefore, it has to remain the feature guiding the matching process. However, we claim that is preferable in our context to match nodes that have slightly diverging TSVs but a large number of children and that are at a close hierarchy level than nodes with an exact TSV match but that are far away in the hierarchy and that contain no structural information because they are close to the leaves of the tree.

4.3.2.3.3. Polygamous graph matching. The last important adaptation with respect to the initial shock graph approach corresponds to the ability to output several matching solutions instead of one. Our algorithm can be considered as a polygamous graph matching method. Thus, each time multiple matches having a similar score (as defined in Eq. (6)) exist, we create a new set of correspondence, that we denote 'branch'. We define the global cost of each branch as $\text{Cost}_{\text{branch}} = \sum_{i \in \text{branch}} CM_i(a,b)$. When the algorithm terminates, we sort all the generated branches by increasing order of cost. Matching each branch cost with a specified threshold makes it possible to output a set of *probable* correspondences.

4.3.2.3.4. Top/down verification. At this stage, the approach only uses the structural information of the graph to label the visible body parts. This strategy has been chosen because it only models the body part assembly, leading to an approach that is invariant to the viewpoint and the pose of the person. However, there are correspondences that are topologically consistent but that are not satisfying regarding geometry of temporal coherence. Thus, we propose to solve the graph matching ambiguities using a top/down verification step. In our context, there are two kinds of ambiguities that are common. Firstly, the head might be confused with an arm, in case of a single segment being extracted in the image. Then, the verification step uses the geometrical information of the model, by assuming that the head should be the segment the most parallel to the torso. Thus, for each candidate N_I for the head (which father is F_{NI}), we compute the projection P_R on the segment being identified as the torso, with image nodes denoted N_0 and N_1:

$$P_R = \overrightarrow{N_i - F_{NI}} \cdot \overrightarrow{N_1 - N_O} \tag{7}$$

The head is supposed to be the node for which P_R is maximum. This assumes that the head and torso slopes are close to each other (which is a reasonable hypothesis whatever the configuration), while arms and torso slopes are not. Finally, the head is supposed undetected in the image if the maximum P_R value is below a given threshold.

A second common topological ambiguity case corresponds to left/right configurations between legs and arms that are indistinguishable only regarding the structure of the image and model graphs. These ambiguities cannot be resolved by using simple geometrical reasonings either. Thus, we propose to track each node over time. After initialization, for each node candidate for an left/right arm/leg that have previously been detected, a tracking score is estimated by computing the distance in the image between the current candidate and the previously detected node. This computation is performed in a similar manner as defined in Eq. (5) for disambiguating the root identification. Finally, we compute a score relating the appearance of a given candidate and the appearance of its previously corresponding limb. How this appearance update and matching are performed are detailed in Section 5. Globally, the verification step processes as follows. All remaining branches are re-ranked using the geometrical, tracking and appearance feature described above. The set of correspondence with the smallest cost gives the body part labeling result at the current time step.

5. Tracking people with AAMs

In the previous section, we explain how a set of segments is extracted and labeled from the silhouette, in order to identify the visible body parts in the image. By now, we attempt at capturing an appearance model for each limb so that a robust descriptor of each human in the sequence can be learned on the fly, and used latter on for recognition. the block diagram shown in Fig. 9 illustrates the main component of this section, i.e. how learning and using the appearance model for recognition.

We choose to model each body part with an ellipse, with the parameters $(C; W; L; \alpha)$, corresponding to center position, length, width and angle between the length and the x-axis, respectively. Although we could think directly

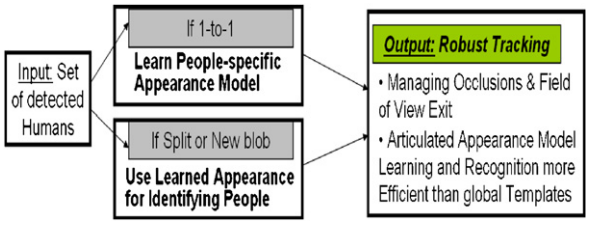

Fig. 9. Overview of the appearance learning and processing approach.

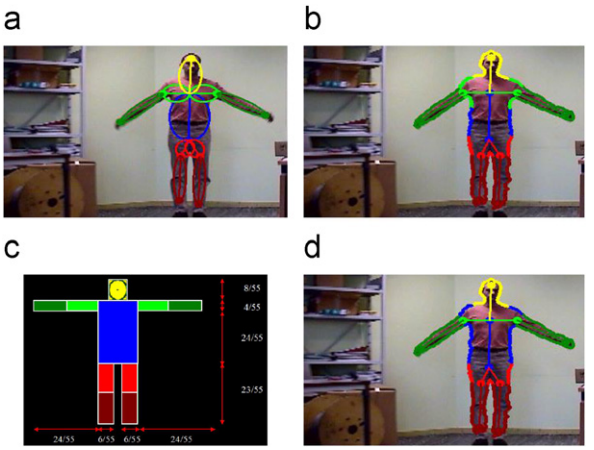

Fig. 10. Articulated appearance model generating. (a) Ellipse from segments. (b) Labeling using Eq. (8). (c) Model geometry. (d) Final labeling.

using the segment labeling results to estimate each limb ellipse parameter, this option actually presents two drawbacks. Firstly, we judge the accuracy of the joint position estimation insufficient, particularly for the neck, because the segment extraction is performed after smoothing the silhouette. This is illustrated in Fig. 10(a). Moreover, we want to generate an AAM for which the segments detected as the shoulders and the hip are part of the torso template. Thus, we use initial contour extracted after the background subtraction scheme (before smoothing), and we determine for each of its point P_i the closest previously labeled segment. Formally, we compute, for each P_i ($i \in \{1;N\}$) the distance $\delta(P_i,S_j)$ to each labeled segment S_j ($j \in \{1;M\}$) with vertices $E_1(S_j)$ and $E_2(S_j)$. Let us denote $H(P_i)$ the orthogonal projection of P_i on S_j, $\delta(P_i,S_j)$ can thus be computed in the following way:

$$\delta(P_i,S_j)=\begin{cases}\|\overrightarrow{P_iH(P_i)}\| & \text{if } HP_i \in S_j \\ & \text{and } [P_i;HP_i]\cap\mathscr{C}=\emptyset \\ \min\limits_{k\in 1,2}(\|\overrightarrow{P_iE_k(S_j)}\|) & \text{if } [P_i;E_k(S_j)]\cap\mathscr{C}=\emptyset \\ \infty & \text{otherwise}\end{cases} \tag{8}$$

Eq. (8) makes it possible to match each contour point to a given labeled segment. This is illustrated in Fig. 10(b). The AAM consists in a 10-part body model, as illustrated in Fig. 10(c). Thus, each contour point identified as being part of one of the torso, hips or shoulders segments is associated to the torso template. The final labeling using this constraint is illustrated in Fig. 10(d).

5.1. Learning people appearance: textural and shape template generation

Once the silhouette contour points are partitioned to match one human body model template, we determine the best fitting ellipse for each limb. It is thus translated, rotated and scaled to match the geometrical model shown in Fig. 10(c). Thus, this registration to a fixed articulated structure makes the appearance update invariant to affine transforms. Note that the ratio between the different limbs in Fig. 10(c) has been fixed by consulting anthropometrics data, and corresponds to a front view. However, it must be highlighted that this choice has no consequence on the articulated model update. For example, the size of the different body parts for a side view will be warped to those corresponding to a front view, without altering the quality of the templates update nor on the possible matching process (see next sections).

Once the registration is performed, we use a temporal appearance model inspired from Haritaoglu et al. [20], and that has been extended by Zhao et al. [57] for integrating color. The main strength of this appearance modeling corresponds to its ability to combine the people appearances seen over time to build a template that fully exploits the temporal aspect of the video sequence. As pointed out in Section 2.3, performing this averaging is only relevant for rigid objects. It is thus completely violated for the whole human body. However, we claim that this assumption is reasonably fulfilled for each body part being a rigid part of the articulated structure, and we then update the appearance for each of the detected limb.

Thus, each limb appearance model is constituted of two templates having the same meaning as those previously introduced in [20,57] for the whole body. The *weight template* W captures shape information. It represents the foreground probability, i.e. the probability for a given pixel of the bounding box to be part of a limb. The *textural templates* T captures color and textural features and gives a discriminative description of the objects' appearance. Updating templates proceeds as follows. Let us denote $I^t(x,y)$ to be the transformed current image patch at time t, after aligning detected and model bounding boxes, and $P^t(x,y)$ as

$$P^t(x,y)=\begin{cases}1 & \text{if } (x,y) \text{ is inside current body part} \\ 0 & \text{otherwise}\end{cases} \tag{9}$$

$P^t(x,y)$ corresponds thus to the binary map of the silhouette after the geometrical transformations. Then, the update equations for W and T are thus given by

$$\begin{aligned} W^t(x,y) &= W^{t-1}(x,y)+P^t(x,y) \\ T^t(x,y) &= \frac{T^{t-1}(x,y)W^{t-1}(x,y)+I^t(x,y)}{W^{t-1}(x,y)+1}\end{aligned} \tag{10}$$

Note that $T^t(x,y)$ is a 3D vector containing color components treated independently.

5.2. People identification: matching AAMs

So far, we learned each people appearance when it was easily visible (i.e. when a one-to-one dynamical region liking is observed). We now explain how the updated AAM can be used to *identify people* and maintain a robust tracking in complex situations. We make an attempt at recognizing people in two usual scenarios (see Fig. 2). Firstly, when a new person enters the scene, we want to decide whether the person has already been tracked in the past. This application is particularly interesting in multi-camera settings. Secondly, when a split is detected for a merged blob at the region level. This commonly

occurs in case of occlusions, when people are crossing or interacting.

In both cases, the previously learned AAMs are used to identify people by their color, textural and shape properties. Basically, the recognition is carried out by computing a similarity measurement between AAMs. Formally, let us denote $A^k(P_i) = \{T^k(P_i), W^k(P_i)\}$ the kth AAM for the kth limb of the person P_i as the concatenation of the weight and textural templates defined in Eq. (10). As in [20], we propose to compute the distance $D_a(A^k(P_i), A^k(P_j))$ between the kth body part appearance models of two people P_i and P_j in the following way:

$$D_a(T^k(P_i), T^k(P_j)) = \frac{\sum_{(x,y)\in L_k\times H_k} \|W^k(P_i)(x,y)\cdot T^k(P_i)(x,y) - W^k(P_j)(x,y)\cdot T^k(P_j)(x,y)\|}{\sum_{(x,y)\in L_k\times H_k}[W^k(P_i)(x,y) + W^k(P_j)(x,y)]} \quad (11)$$

L_k and H_k correspond to the kth template width and height, respectively (see Fig. 10(c)). Then, we define a distance $D_a(P_i, P_j)$ between the AAMs of people P_i and P_j as follows:

$$D_a(P_i, P_j) = \frac{\sum_{k=1}^{8} \omega_k \cdot D_a(A^k(P_i), A^k(P_j))}{\sum_{j=1}^{8} \omega_k} \quad (12)$$

The weights ω_k determine the relative importance of each body part in the similarity measurement. ω_k is setup proportionately to the number of times the given body part has been updated. This seems to be a astute choice, as the learned appearance confidence is directly related to the amount of data collected for performing the averaging.

In the case of a new person P_N entering the scene, we perform the recognition by minimizing the distance between the newly formed appearance model and the M previously learned ones, i.e. we determine j_0 such that

$$P_{j_0} = \arg\min_j \{D_a(P_N, P_j)\}_{j\in\{1;M\}}$$

Matching $D_a(P_N, P_{j_0})$ to a given threshold D_a^T makes it possible to identify a new person entering the scene.

When we are trying to recognize a given person P_S after a sequence of merge/split at the region level (see Fig. 2), the strategy differs slightly. Indeed, the split blob comes from a given merge region containing a known number M' of people among the total M number of people learned by the system over time. Thus, the people identification is carried out by minimizing the distance $D_c(P_S, P_j)$, $j \in \{1; M'\}$:

$$D_c(P_S, P_j) = w_a * D_a(P_S, P_j) + w_t \times D_t(P_S, P_j) \quad \text{where } D_t(P_S, P_j) = \|C(P_S) - C(P_j)\| \quad (13)$$

The distance $D_c(P_S, P_j)$ contains two terms D_a and D_t with associated weight w_a and w_t that have been set experimentally. In addition to the appearance term D_a defined in Eq. (12), the people recognition involves a tracking term D_t, that measures the distance between the centers of the two people bounding boxes $C(P_S)$ and $C(P_j)$. Indeed, as we detail in the next section, we propose an approach for tracking people during an occlusion. Thus, the tracking output for each of the M' people inside a single blob is used for specifying the people identification when the blob splits.

5.3. Tracking during occlusion

So far, the proposed algorithm is able to learn people appearance as soon as a region-based tracking can easily be achieved, and use the people-specific appearance model to identify humans. In this section, we explore the problem of tracking people during an occlusion, i.e. when a merge is detected at the region level (see Fig. 2).

Tracking object during occlusions is among the most challenging problem in computer vision. In visual tracking, the problem consists in using appearance feature for recovering the object position. The case of occlusion is particularly difficult because each object likelihood often becomes multi-modal during the occlusion. Thus, tracking estimators modeling the likelihood with Gaussian distributions (like Kalman filtering [25]) are likely to fail and make the tracking drift irreversibly. To overcome this shortcoming, particle filters have been proposed, the CONDENSATION (Conditional Density Propagation) algorithm [21] being considered as pioneer work. The principle consists in trying to approximately estimate the likelihood density instead of finding an optimal solution only valid when the Gaussian assumption is fulfilled. Basically, the approach can be separated into three steps: motion modeling for prediction position, weighting each candidate position, and re-sampling, as illustrated in Fig. 11.

Formally, the particle filtering principle consists in approximating the distribution of the considered state vector X of the object by a set of N samples $x^{(i)}$ with associated weights $\pi^{(i)}$:

$$X = \{(x^{(i)}, \pi^{(i)}), i = 1, \ldots, N\} \quad \text{with} \sum_{i=1}^{N} \pi^{(i)} = 1 \quad (14)$$

The PDF estimation is based on the *importance sampling* principle, i.e. consists in associating to each sample or particle a weight $\pi^{(i)}$ that measures its likelihood. For tracking during occlusions, we propose here to use an hybridization of the CONDENSATION algorithm, that is only applied in that situation, i.e. only when sophisticated

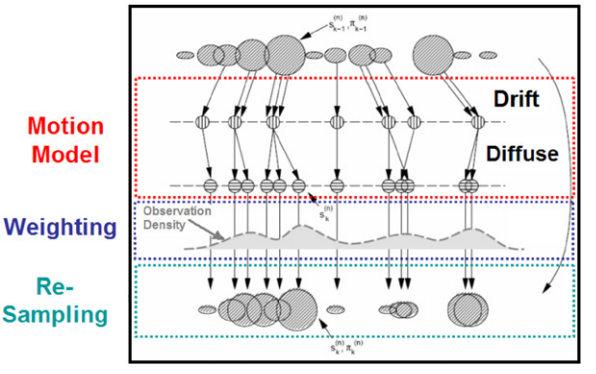

Fig. 11. Particle filtering for tracking during occlusions, adapted from [21].

statistical filtering are required for estimating people positions. We claim that this strategy is optimal when attempting to maximize the accuracy/computational time ratio. The state vector $X = \{C, S\}$ that aims at being estimated corresponds to the bounding box center position ($C = (Cx, C_y)$) of each person inside the merged blob, as well as its scale factor S. It is initialized with the parameters of each individual person before the merge occurs. We use a very simple first-order motion model, followed by a Gaussian noise diffusion to explore the search space. The weighting consists in computing a simple color correlation measurement between the propagated particles and the observed image data. The main specificity of the hybridization in our context corresponds to the possibility to bound the search space, i.e. to discard some particles that are outside of the merged region containing several people. This both improves the accuracy of the estimation and decreases the required computation time. A standard re-sampling strategy is used to support particles with large 'importance', or likelihood. Finally, the algorithm outputs the estimated state $\widehat{x}$ corresponding to the weighted average of the particle set: $\widehat{x} = E[X] = \sum_{i=1}^{N} \pi^{(i)} x^{(i)}$.

6. Results

6.1. Body part labeling results

We present here some results illustrating the efficiency of the proposed limb labeling strategy from the extracted silhouette.

Fig. 12 focuses on results corresponding to the graph matching part of the system. The coloring convention used is related to Fig. 6: head is drawn in yellow, torso in blue, arms in green and legs in red. Note that the distinction between the two potential segments in arms and legs is illustrated with a difference of intensity. These results prove the ability of the approach to manage unspecified viewpoints or human postures, illustrating the invariance of the topological features extracted from the graphs. In Figs. 12(a) and (b), the body part identification is presented for a standing posture with a back view and a side view, respectively. We can notice that the head labeling is properly performed in both cases although arms are formed by a single segment. This is a result of the geometric reasoning presented in Section 4.3.2.3.4, where the verification step makes it possible to disambiguate similar topological configurations. Figs. 12(c), (d) and (e) show the results for sitting, falling and lengthened poses, respectively. We can notice that the torso is properly identified as the largest mean radius segment in each case, illustrating the fact that the image feature proposed in Section 4.2.2 is robust to viewpoint and posture variations. In addition, we can note that the head is localized in Figs. 12(c) and (e) because of the geometrical verification step (the head being the segment whose slope is the closest to the root). At the opposite, in Fig. 12(d) the graph topology is sufficient to conclude as the two segments for the arms are detected. Finally Fig. 12(f) shows an example where someone is walking on the hands. Arms and legs are properly labeled, because the TSV encoded from the graph is totally posture invariant and because the segments connectivity is here sufficient to provide an unambiguous labeling.

In addition, we can see in these examples that padding the TSV with -1's, as described in Section 4.3.2.3.1, makes our topological matching approach much more accurate than the original shock graph method [45]. As an example, let us consider Fig. 12(b). With the original approach, the topological matching with respect to the model is quite uninformative: for instance we can match the three nodes of the arms to any of the nodes 2,3,4,5,6,7 of the model! Such ambiguities do not occur with our proposed encoding of the terminal nodes: we match the node of the elbow to nodes 3 or 6 of the model and the nodes of the hands to nodes 4 or 7 (only left/right ambiguities remain, they are resolved as explained in Section 4.3.2.3.4).

Fig. 13 presents results where a significant number of silhouette image segments is missing. In these examples, at least one branch starting from one of the two root nodes (torso) is absent. These missing visible body parts can come from various situations. There may be partial occlusions in the image: in Fig. 13(a) only a sub-part of the body is visible in the image and in Fig. 13(b) the legs are occluded by the table. In addition, we have to face limbs auto-occlusions, making the corresponding segment extraction from the silhouette impossible. For example in Fig. 13(c), the two legs merge in the silhouette and a single segment is extracted. In Figs. 13(c)–(e), the arms and the torso are not separated in the image. Finally, in Fig. 13(f),

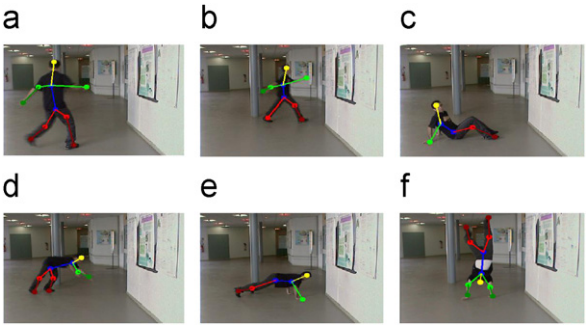

Fig. 12. Labeling results in various poses and viewpoints.

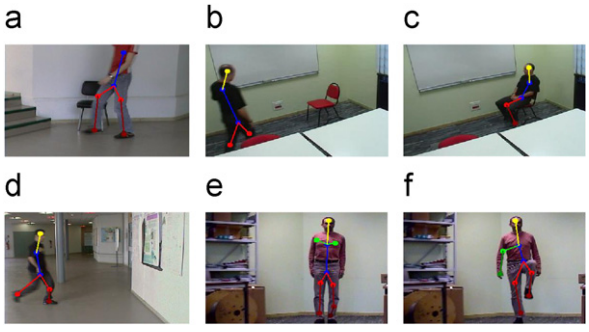

Fig. 13. Matching robustness to limb occlusions.

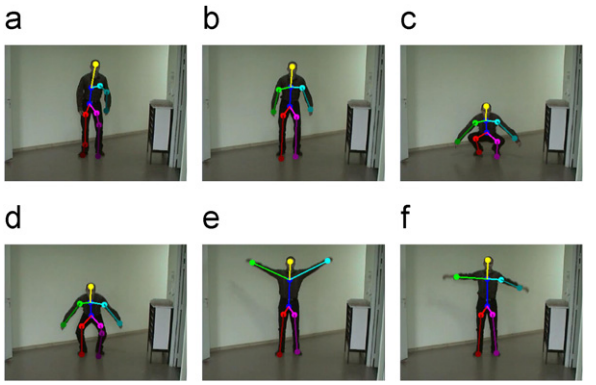

Fig. 14. Tracking results as a top/down verification example.

the left arm is not detected because of a background subtraction inaccuracy, but the algorithm is, however, able to properly label the remaining visible parts. These results illustrate the capacity of the algorithm to match graphs with a different number of nodes, and to identify topologically similar groups of segments.

Finally Fig. 14 points out the results of the tracking part of the system, that is the most important aspect where the verification step takes place. Indeed, symmetrical indistinguishable left/right ambiguities inevitably occur when only relying on topology, and a top/down verification is required to obtain a unique labeling. The color convention has been changed in this example to differentiate left/right arms/legs: green for the right arm, red for the right leg, cyan for the left arm, and magenta for the left leg. In Fig. 14(a), the person is detected for the first time in the sequence. The two legs are properly detected, but a single segment is extracted for the right arm.

At the top/left frame, the person is tracked and graph matching is performed. The left/right limbs are then initialized randomly. After that, each ambiguous configuration is checked against the tracked nodes, making it possible to enforce a unique coherent labeling over time. From this time step, each time a topological ambiguity appears, the set of correspondences are re-ranked using the tracking cost explained in Section 4.3.2.3.4. As illustrated in Figs. 14(b)–(f), the tracking is properly performed all along the sequence, making it possible to disambiguate the left/right labeling of the body parts.

6.2. Articulated appearance results

In this section, we present results illustrating the final goal of the proposed approach, i.e. how an AAM is learned and used to track people in video sequences.

Fig. 15 illustrates how the model is being updated. Each detected and labeled limb is rotated and scaled to update the geometrical model shown in Fig. 10(c). The generated appearance is presented in the third row. In the shown frames, all the body parts of the model are detected. Obviously, this is not always the case. This is actually a goal of this module to be robust enough not detecting limbs when the segmentation yields poor results because it prevents wrong updating of the model. For instance,

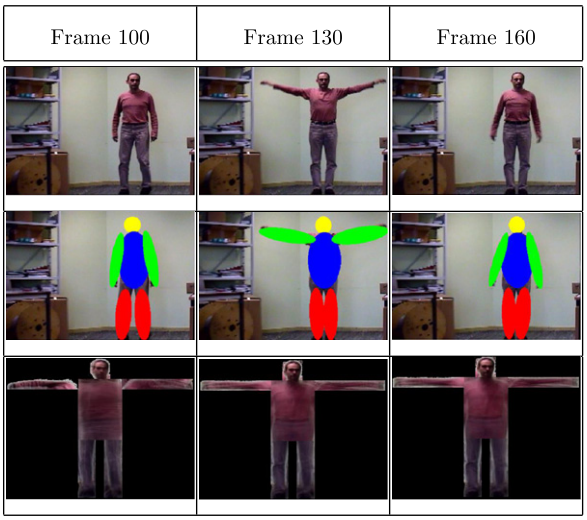

Fig. 15. Articulated appearance model updating.

between frames 130 and 160 the legs are often not detected because the man is squatting himself, resulting in a single blob. As we can see, the model update results are very good. The textural template will provide an interesting feature for recognition.

6.2.1. Experimental validation, comparison and discussion

We evaluate the proposed tracking algorithm on a sample of 40 sequences, containing manually collected videos and sequences downloaded from public datasets. The samples correspond to videos acquired from eight different indoor sites and from six different outdoor sites, with different viewpoints and lighting conditions. Each sequence contains from two and five people simultaneously present in the sequence, with complex interactions making the tracking challenging: strong occlusions (with people crossing and coming back to their paths), people going out of the field of view and possibly re-appearing, etc. We propose here to evaluate the tracking performances by estimating the recognition rate using the learned AAM, i.e. how many times the human identification is properly carried out when a split is detected at the blob level (or when a new person enters the scene). Thus, in our overall dataset, 212 identifications have to be made by the algorithm. The same set of parameters for learning and recognizing appearance (Sections 4 and 5, respectively) has been used for the overall testing. The performances are shown in Table 1. We compare the performances with methods modeling appearance in a more simplistic way, such as global templates or PDF models. The recognition rates have been evaluated with the global template proposed by Zhao et al. [57] that extends Haritaoglu et al. [20] approach by integrating color in the appearance model. For PDF, we propose a comparison with a color histogram (in the RGB space) modeling, as proposed in [22,54,28]. As the low-level steps (background subtraction and blob tracking) are very similar in our algorithm than in the compared

Table 1
Tracking performances: evaluating recognition rates

	# Split	# Identified people	Recognition rate (%)
AAM	212	197	93
Global template	212	170	80
Color histograms	212	151	71

methods, we claim that the comparison between the strategies for modeling human appearance is meaningful.

As we can see in Table 1, the proposed AAM significantly outperforms both global templates and PDF models: the global recognition rate is about 93%, since it reaches 80% and 71% for the global template and color histograms, respectively. This experimental validation confirms our intuition (presented in Section 2.3) that our approach is more adapted to learn human appearance than global templates or PDF. Indeed, PDF modeling ignores spatial information, and are thus not accurate enough to discriminate different humans. On the other hand, global templates are not adapted to learn human appearance due to the complex articulated structure of the human body, and commonly suffer from update errors in common cases such as those illustrated in Fig. 1. We now give examples from the studied dataset illustrating these points.

Fig. 16 illustrates how people appearance can successfully be learned and used for recognition with our approach, as it would have failed using a PDF modeling. This sequence has been downloaded from Duric's web page (http://www.cs.gmu.edu/~zduric/Demos/) and was used to validate his work dedicated to combining color and edge information to perform background subtraction [22,28]. It must be pointed out that the goal of their approach was to provide a segmentation mask for people, not to detect them individually, track or recognize them.

In this sequence, our algorithm processes as follows. At frame 40, the red-framed and green-framed persons enter the scene. They are detected by the background subtraction algorithm and tracked by the dynamical region liking and the temporal coherence enforcement, without any ambiguity until the frame 100. Thus, AAMs are learned for each person. At frame 100, an occlusion is detected by a merge at the region level. At frame 150, the people separate from each other, and a split is detected at the region level. The identification is successfully performed by matching the previously learned AAMs. Contrarily, the recognition fails when using a color histogram as input feature, because this descriptor is not discriminative enough. Indeed, the two color histograms are essentially the same, and present modes next to the dark and light blue region in the color space. On the other hand, the AAMs can be discriminated by the torso, faces and arms templates, that significantly differ. Moreover, the textural features encoded in the appearance template are here very useful to model the checked shirts by its ability to capture its periodical pattern. Thus, the AAM can efficiently discriminate the two torsos, since the red-framed person template is texture free. In this sequence,

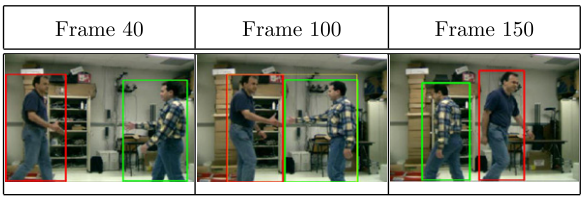

Fig. 16. The articulated appearance model captures a spatial information that makes it possible to discriminate different textures. See text.

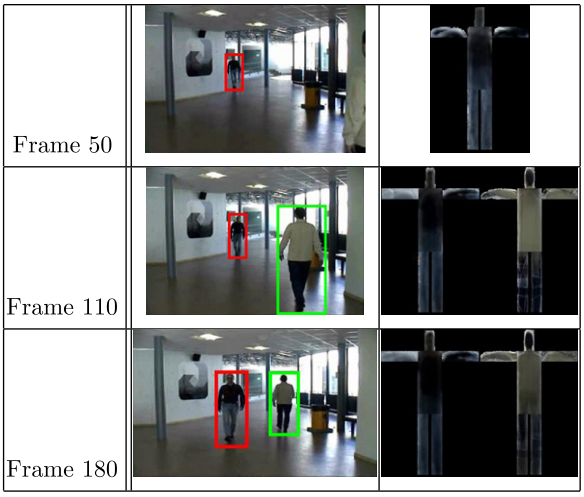

Fig. 17. Scale invariance of the model.

we can note that using a global template makes it possible to successfully identify the two people, because their apparent sizes remain the same, and because the template update is easy to achieve.

Fig. 17 illustrates how our strategy for model updating can manage difficult situations such as a person's partial occlusion, and important changes in size due to perspective effects.

At frame 50, the two people enter the scene, one being far away from the camera with respect to the viewing direction, the other one being very close to it. Let us define P_1 being the red-framed, and P_2 the green-framed one. From frames 50 and 110, P_2 is partially occluded. At the beginning only its torso, head and legs are visible. Progressively its legs appear until frame 110 where he is entirely inside the field of view. The third column shows how its appearance model is being robustly built at frame 100, meaning that the body parts have been properly identified and only visible parts have been updated. Between frames 110 and 180, P_1 and P_2 walk in opposite directions, resulting in a large variation in size of their silhouette due to perspective effects. As we can notice, the AAMs shown in the third column are properly updated. Indeed, each time a limb is detected, it is rescaled to match its model size, resulting in a proper update of the corresponding template. Oppositely, using a global template with an approach similar to [20] or [57] makes the template update poor, as illustrated in Fig. 1. Indeed,

the global templates with a fixed size are neither robust to apparent changes in size nor to partial occlusions of the limbs. Thus, in the same sequence, P_2 goes out of the field of view and re-appear later: the learned template with a global model makes the identification fail, since it succeeds with our AAM. The recognition using color histogram manages to properly discriminate P_1 from P_2 in this specific case. However, we again point out the fact that spatial information is lost when using histograms. Thus, P_2 may be confused with another person (e.g. wearing a blue shirt and yellow pants) in a more complex sequence.

Fig. 18 shows an example in an outdoor environment. It illustrates the ability of the approach to track people during an occlusion, and how this occlusion handling improves the identification performance and the overall tracking robustness. The first row shows the results as the two isolated people enter the space. We can notice the quality of the textural templates, presented in the last column, although the silhouette is very small in the image. At frame 300, the people start crossing and a single region is now extracted by the background subtraction algorithm: a merge is thus detected at region level. The third row illustrates the use of the particle filter for tracking during occlusion. At the three successive times, thin rectangles represent the particles propagated for each person, and the thick rectangle is the estimated position output by the filter, corresponding to its mean state. As we can see, even with the simplistic assumption carried out (simple state vector, 1st order motion model, simple weighting measurement, etc.), the tracking is surprisingly satisfying. The probabilistic framework of the filter indeed enables flexibility, and illustrates the fact that it is particularly adapted for tracking during occlusions. The main reason is that when the likelihood becomes multimodal, because the people occlude each other, the particle filter makes it possible to explore the search space at a large scale. When the people separate, the likelihood shows strong peaks around the people appearance. Then, the filter is able to densely re-sample particles around the right position, and the tracking can be properly maintained. Finally, frame 360 shows how the matching between AAMs again makes it possible to recognize people. It can be noticed that the people recognition is actually performed by the conjoint use of the appearance models and the positions estimated by the particle filter, as explained in Section 5.2. We can point out the fact that the people were walking on a side view, as illustrated by the shown textural templates. It has neither altered the appearance model generation nor the matching, as explained in Section 5.1. In this example, the recognition fails using color histograms and global templates. Indeed, some segmentation errors occur from 200 to frame 300, making the global updates inadequate. However, using the particle filter output (in addition to the appearance learned) makes it possible to properly identify the two persons.

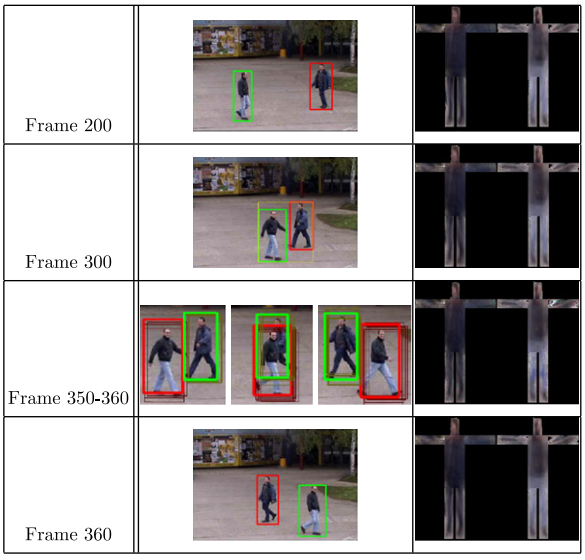

Fig. 18. People recognition for a side view and tracking during occlusion.

In the 40 analyzed sequences, our algorithm only fails at identifying 15 people. We can classify the observed recognition errors in the following way. Most of them come from sequences where many people have very similar clothing, making the different AAMs very confusing. In these sequences, the different people are mainly recognizable by their face. However, the matching score defined in Eq. (11) for the head template does not pretend to be competitive for face recognition with respect to the state of the art standards. Moreover, the difficulty of the recognition is exacerbated by the low resolution images extracted in our context. The second type of error is related to the recognition of people re-entering the scene. The parameter D_a^T for thresholding the dissimilarity D_a between appearance models (Eq. (12)) has been difficult to setup in the experiments. Indeed, we have to set D_a^T so that a new person entering the scene is not identified as the most similar previously tracked one, but we must be able to recognize the same person in a different viewpoint. As the AAM is by essence 2D, it has difficulties in managing such situations. A solution would be to estimate the viewing direction, as proposed in the conclusion (see Section 7).

6.3. Processing time

Table 2 gathers the complexity of the main steps of the proposed tracking algorithm. The experiments were carried out on a Pentium IV at 2.66 GHz, with 512 MB of RAM. The software was built in C++ using Microsoft Visual studio 2005 (version 7.0). The video sequences were composed of image sequence with CIF resolution (i.e. of size 320×240).

As we can see, the background subtraction is the most computational demanding step. Indeed, it is performed for each image pixel. In addition, we fix the maximal number of Gaussian distribution (see Section 3.2.1) to $K_{max} = 3$ in our experiments. The data association corresponding to the blob tracking presented in Section 3.2.2 only requires some ms. Thus, the two previous low-level steps require around 22 ms to be carried out. This processing time is

Table 2
Computation time for the main steps of the tracking approach

	Motion detection	Blob tracker	Limbs labeling	Learning appearance	Particle filtering
Time (ms)	20	2	4	1	10

constant, i.e. it does not depend on the content of the video sequence. Oppositely, the processing time for the next three steps (limbs labeling, learning appearance and particle filtering) is given per tracked person. Thus, we can notice that the body part labeling using the graph matching algorithm detailed in Section 4 only requires 4 ms. As previously mentioned, the complexity of the graph matching algorithms is actually not a problem in our context: as we usually extract few segments from the silhouette, the proposed approach is computationally efficient. Learning appearance as detailed in Section 5.1 is insignificant. Finally, the particle filtering approach requires approximately 10 ms to track to people inside a single region. It is worth mentioning that the tracking algorithm can be successfully achieved without the particle filtering. This step is indeed used for tracking during occlusion, and provides additional information for performing the recognition, as explained in Section 5.2. Therefore, the tracking of k people with the learning/identification proposed algorithm requires: $22 + 5 * k$ ms. The proposed approach can thus be used for tracking up to five person in real-time. If we only request 10 images per second to be analyzed, which can be sufficient for appearance learning and identification, up to 15 people can be handled. However, it must be noted that the algorithm is not dedicated to managing very crowded environments, because the limbs labeling and appearance model updating require individual person to be detected.

7. Conclusion and future works

We propose an hybrid tracking approach, that is able to deal with several people and run in real-time. The algorithm firstly uses a simple blob tracker dedicated to detecting easy situations, and takes advantage of them to learn people appearance. For properly updating the appearance models over time, we chose to detect and label the different limbs from the silhouette, that constitute rigid parts of the articulated structure. This task is managed by a graph matching strategy. Importantly, the approach only encodes topological information for performing the labeling, making it possible to label body parts for any pose (standing, lying, sitting, etc.) and viewpoint. The limbs labeling enables us to generate a person-specific appearance model, that provides a discriminative feature used to identify people and maintain the tracking in complex situations. We propose two main directions for future works. Firstly, building a 3D appearance model would make the identification step completely viewpoint independent, and is therefore appealing. However, it would require to estimate the person viewing direction, and is not straightforward. Secondly, tracking body parts individually seems to be a very promising way for increasing the system robustness. At the moment, this step is mainly dedicated to disambiguating the body part labeling. It could be interesting to use each body part appearance to improve the tracking performances. It would include occlusion detecting, and the ability to keep tracking limbs during self-occlusions (arms and torso for example). More generally, additional top–down verifications may be thought to analyze the blob tracker errors.

References

[1] D. Attali, A. Montanvert, Computing and simplifying 2d and 3d continuous skeletons, Comput. Vision Image Understanding 3 (1997) 261–273.
[2] G. Ball, D. Hall, A clustering technique for summarizing multivariate data, Behavioral Sci. 12 (1) (1976) 153–155.
[3] S. Birchfield, Elliptical head tracking using intensity gradients and color histograms, in: CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 1998, p. 232.
[4] G.R. Bradski, Computer vision face tracking for use in a perceptual user interface, Intel Technol. J. 1 (Q2) (1998) 15.
[5] L. Carminati, J. Benois Pineau, Gaussian mixture classification for moving object detection in video surveillance environment, ICIP, 2005, pp. 113–116.
[6] J. Cesar, R.M.E. Bengoetxea, I. Bloch, P. Larranaga, Inexact graph matching for model-based recognition: evaluation and comparison of optimization algorithms, Pattern Recognition 19 (2005) 2099–2113.
[7] H. Chen, T. Liu, Trust-region methods for real-time tracking, in: ICCV, 2001, pp. 717–722.
[8] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: CVPR, 2000, pp. 142–151.
[9] D.M. Cvetkovi, P. Rowlinson, S.Simic, Eigenspaces of Graphs, Cambridge University Press, Cambridge, UK.
[10] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: C. Schmid, S. Soatto, C. Tomasi (Eds.), International Conference on Computer Vision and Pattern Recognition, vol. 2, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot, 2005, pp. 886–893.
[11] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: European Conference on Computer Vision, 2006.
[12] A. Elgammal, D. Harwood, L. Davis, Non-parametric model for background subtraction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1999.
[13] R. Fabbri, L. Estozi, L. da F. Costa, On Voronoi diagrams and medial axes, J. Math. Imaging Vision 1 (2002) 27–40.
[14] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial structures for object recognition, Internat. J. Comput. Vision 61 (1) (2005) 55–79.
[15] P. Fieguth, D. Terzopoulos, Color-based tracking of heads and other mobile objects at video frame rates, in: CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), IEEE Computer Society, Washington, DC, USA, 1997, p. 21.
[16] H.N. Gabow, M.X. Goemans, D.P. Williamson, An efficient approximation algorithm for the survivable network design problem, in: IPCO, 1993, pp. 57–74.
[17] D.M. Gavrila, The visual analysis of human movement: a survey, Comput. Vision Image Understanding CVIU 73 (1) (1999) 82–98.
[18] D.M. Gavrila, V. Philomin, Real-time object detection for smart vehicles, in: International Conference on Computer Vision and Pattern Recognition, 1999, pp. 87–93.
[19] V. Girondel, A. Caplier, L. Bonnaud, Real time tracking of multiple persons by Kalman filtering and face pursuit for multimedia applications, in: IEEE Southwest Symposium on Image Analysis and Interpretation, Lake Tahoe, Nevada, USA, pp. 201–205.
[20] I. Haritaoglu, D. Harwood, L. Davis, W4: real-time surveillance of people and their activities, IEEE Trans. Pattern Anal. Machine Intell., Vol. 22, 2000, pp. 809–830.
[21] M. Isard, A. Blake, Condensation-conditional density propagation for visual tracking, Internat. J. Comput. Vision 29 (1) (1998) 5–28.
[22] S. Jabri, Z. Duric, H. Wechsler, A. Rosenfeld, Detection and location of people in video images using adaptive fusion of color and edge

information, in: Proceedings of the 15th International Conference on Pattern Recognition, vol. 4, Barcelona, Spain, 2000, pp. 627–630.
[23] A. Jepson, D. Fleet, T. Maraghi, Robust online appearance models for visual tracking, in: CVPR, 2001.
[24] J.-M. Jolion, W.G. Kropatsch, M. Vento, Graph based representations, in: Third IAPR International Workshop on Graph Based Representations, 2001, ISBN: 88-7146-579-2.
[25] R. Kalman, A new approach to linear filtering and prediction problems, Trans. ACME J. Basic Eng. (1960) 343–356.
[26] B. Kégl, A. Krzyzak, Piecewise linear skeletonization using principal curves, IEEE Trans. Pattern Anal. Machine Intell. 24 (2002) 59–74.
[27] G. Malandain, S. Fernandez-Vidal, Euclidean skeletons, Image Vision Comput. 16 (1998) 317–327.
[28] S.J. McKenna, S. Jabri, Z. Duricand, A. Rosenfeld, H. Wechsler, Tracking groups of people, Comput. Vision Image Understanding 80 (2000) 4256.
[29] D. Merad, Reconnaissance 2d/2d et 2d/3d d'objets à partir de leurs squelettes, Thèse de doctorat à l'université d'Evry Val d'Essonne, 2004.
[30] D. Merad, J. Didier, M. Scuturici, Tracking 3d free form object in video sequence, in: Third Canadian Conference on Computer and Robot Vision, IAPR-CRV 2006, Quebec, June 2006.
[31] K. Mikolajczyk, C. Schmid, A. Zisserman, Human detection based on a probabilistic assembly of robust part detectors, in: ECCV, vol. I, 2004, pp. 69–81.
[32] T.B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Comput. Vision Image Understanding 104 (2) (2006) 90–126.
[33] V. Morellas, I. Pavlidis, P. Tsiamyrtzis, Deter: detection of events for threat evaluation and recognition, Machine Vision Appl. 15 (1) (2003) 29–45.
[34] G. Mori, Recovering 3d human body configurations using shape contexts, IEEE Trans. Pattern Anal. Machine Intell. 28 (7) (2006) 1052–1062 (senior Member-Jitendra Malik).
[35] G. Mori, X. Ren, A. Efros, J. Malik, Recovering human body configurations: combining segmentation and recognition, in: CVPR, vol. 2, 2004, pp. 326–333.
[36] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, T. Poggio, Pedestrian detection using wavelet templates, in: Proceedings of the Computer Vision and Pattern Recognition, 1997, pp. 193–199.
[37] D. Ramanan, D.A. Forsyth, A. Zisserman, Tracking people by learning their appearance, IEEE Trans. Pattern Anal. Machine Intell. 29 (1) (2007) 65–81.
[38] D. Reid, An algorithm for tracking multiple targets, IEEE Trans. Automat. Control 24 (6) (1979) 843–854.
[39] X. Ren, A.C. Berg, J. Malik, Recovering human body configurations using pairwise constraints between parts, in: Proceedings of the 10th International Conference on Computer Vision, vol. 1, 2005, pp. 824–831.
[40] S.W. Reyner, An analysis of a good algorithm for the subtree problem, SIAM J. Comput. 6 (4) (1977) 730–732.
[41] E. Salvador, P. Green, T. Ebrahimi, Shadow identification and classification using invariant color models, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, 2001, pp. 1545–1548.
[42] H. Schweitzer, J.W. Bell, F. Wu, Very fast template matching, in: ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV, Springer, London, UK, 2002, pp. 358–372.
[43] A. Shokoufandeh, S.J. Dickinson, A unified framework for indexing and matching hierarchical shape structures, in: IWVF-4: Proceedings of the 4th International Workshop on Visual Form, Springer, London, UK, 2001, pp. 67–84.
[44] A. Shokoufandeh, D. Macrini, S. Dickinson, K. Siddiqi, S.W. Zucker, Indexing hierarchical structures using graph spectra, IEEE Trans. Pattern Anal. Machine Intell., 2005.
[45] K. Siddiqi, A. Shokoufandeh, S. Dickinson, S.W. Zucker, Shock-graphs and shape matching, Internat. J. Comput. Vision 30 (1999) 1–24.
[46] C. Sminchisescu, B. Triggs, Estimating articulated human motion with covariance scaled sampling, Internat. J. Robotics Res. 22 (6) (2003) 371–391 (special issue on Visual analysis of human movement).
[47] C. Stauffer, W. Grimson, Learning patterns of activity using real-time tracking, IEEE Trans. Pattern Anal. Machine Intell. 22 (8) (2000) 747–757.
[48] G. Stewart, J. Sun, Matrix Perturbation Theory, Academic Press, San Diego, 1990.
[49] H. Tao, H.S. Sawhney, R. Kumar, Object tracking with Bayesian estimation of dynamic layer representations, IEEE Trans. Pattern Anal. Machine Intell. 2002, pp. 75–89.
[50] N. Thome, S. Miguet, A robust appearance model for tracking human motions, in: International Conference on Advanced Video and Signal-Based Surveillance, 2005.
[51] P. Viola, M. Jones, Robust real-time object detection, Internat. J. Comput. Vision, 2002.
[52] P. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, in: ICCV, vol. 02, 2003, p. 734.
[53] J. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, England, 1965.
[54] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE Trans. Pattern Anal. Machine Intell. 19 (7) (1997) 780–785.
[55] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38 (4) (2006) 13.
[56] T. Zahn, R. Roskies, Fourier descriptors for plane closed curves, Nature Neurosci. 21 (1972) 269–281.
[57] T. Zhao, R. Nevatia, Tracking multiple humans in complex situations, PAMI 26 (2004) 1208–1221.