O. Barnich, S. Jodogne, and M. Van Droogenbroeck. **Robust analysis of silhouettes by morphological size distributions**, volume 4179 of *Lecture Notes on Computer Science*, pages 734-745. Springer Verlag, 2006. http://orbi.ulg.ac.be/handle/2268/14121

Robust analysis of silhouettes by morphological size distributions

Olivier Barnich[*], Sébastien Jodogne, and Marc Van Droogenbroeck [**]

Department of Electricity, Electronics and Computer Science,
Institut Montefiore B-28, Sart Tilman, B-4000 Liège, Belgium

Abstract. We address the topic of real-time analysis and recognition of silhouettes. The method that we propose first produces object features obtained by a new type of morphological operators, which can be seen as an extension of existing granulometric filters, and then insert them into a tailored classification scheme.

Intuitively, given a binary segmented image, our operator produces the set of all the largest rectangles that can be wedged inside any connected component of the image. The later are obtained by a standard background subtraction technique and morphological filtering. To classify connected components into one of the known object categories, the rectangles of a connected component are submitted to a machine learning algorithm called EXtremely RAndomized trees (Extra-trees). The machine learning algorithm is fed with a static database of silhouettes that contains both positive and negative instances. The whole process, including image processing and rectangle classification, is carried out in real-time.

Finally we evaluate our approach on one of today's hot topic: the detection of human silhouettes. We discuss experimental results and show that our method is stable and computationally effective. Therefore, we assess that algorithms like ours introduce new ways for the detection of human in video sequences.

1 Introduction

During the recent years, the rising of cheap sensors has made of video surveillance a topic of very active research and wide economical interest. In this field, one of the expected major breakthrough would be to design automatic image processing systems able to detect, to track, and to analyze human activities. Unfortunately the amount of data generated by cameras is prohibitively huge, although the informative part of such signals is very tight with respect to their raw content.

Several algorithms in computer vision have been developed to summarize such informative patterns as a set of *visual feature*s. These algorithms generally rely on the detection of discontinuities in the signal selected by *interest point detectors* [1]. Then, a local description of the neighbourhood of the interest points is computed [2] and this description serves to track a feature in successive frames of a video sequence. Methods like this, referred to as *local-appearance methods*, have been used with some success in computer vision applications such as image matching, image retrieval, and object recognition (see [3,4]).

From current literature, it is still unclear whether such local-appearance descriptors are appropriate for tracking human silhouettes, or more specifically for gait analysis. Indeed, they are rather computationally expensive, and as they are inherently local, it is impossible for them to represent the overall geometry of a silhouette. There are two potential solutions to this problem: (1) introduce higher-level descriptors able to represent the relative spatial

[*] Olivier Barnich has a grant funded by the FRIA, Belgium.

[**] This work was supported by the Belgian Walloon Region (http://www.wallonie.be), under the CINEMA project.

arrangements between visual features [5], or (2) take global appearance (such as contours) into consideration instead of local appearance.

Gait analysis techniques based on the global geometry of the objects have been discussed by BOULGOURIS *et al.* [6]. According to them, techniques that employ binary images are believed to be particularly suited for most practical applications since color or texture informations might not be available or appropriate. The contour of a silhouette is probably the most sensible visual feature in this class. A direct use of it is possible, or it can be transformed into a series of Fourier descriptors as common in shape description. Alternatively the width of silhouette, horizontal and vertical projections, and angular representation are other candidates that have been proposed.

In this paper, we propose a novel approach that is at the crossroad between local- and global-appearance techniques. Our approach innovates in that we propose a new family of visual features that rely on a surfacic description of a silhouette. Intuitively, we cover the silhouette by the set of all the largest rectangles that can be wedged inside of it. More precisely, each (local) position in the silhouette is linked to the subset of the largest rectangles that cover it and that are entirely included in the (global) silhouette.

Surfacic descriptors, like the morphological skeleton [7], have already been studied in the scope of shape compression whose goal is to reduce the amount of redundant information. In general, they require large computation times, which makes them less suitable for real-time applications. This contrasts with our features, as it is possible to compute them in real-time, if enough care is taken in the implementation.

This paper describes an attempt to take advantage of such novel features. To illustrate our approach, we focus on the detection of human bodies in a video stream, like in [8,9]. Basically, we apply *machine learning* algorithms on the rectangles of a silhouette to decide, in real-time, whether this silhouette corresponds to that of a learned instance of a human silhouette. This decision is a compulsory step for any gait recognition task, and improvements in this area will impact on the overall performances of algorithms that deal with the automatic analysis of human behavior. Our results show how promising an approach like ours can be.

The paper is organized as follows. We start by describing the architecture of our silhouettes detection and analysis technique in Section 2, which mainly consists in three steps (silhouettes extraction, description, and classification) respectively detailed in Sections 3, 4, and 5. Experimental results, which consist in the application of our method for the detection of human people in video sequences, are discussed in Section 6.

2 Overall architecture

The overall architecture of our silhouettes detection, analysis and classification system is depicted in Figure 1. It comprises three main modules.

1. The first module consists in extracting the candidate silhouettes from the video stream. It is described in Section 3.
2. One of the major difficulties in classification lies in finding appropriate feature measures. In the second module, we use our new granulometric operator to produce a set of features (largest wedged rectangles) describing the extracted silhouettes.
3. The task of the third module is to classify rectangle features to decide whether or not the silhouettes belong to the class of interest. The classification is achieved by the means of an extra-tree learning algorithm as explained in Section 5.

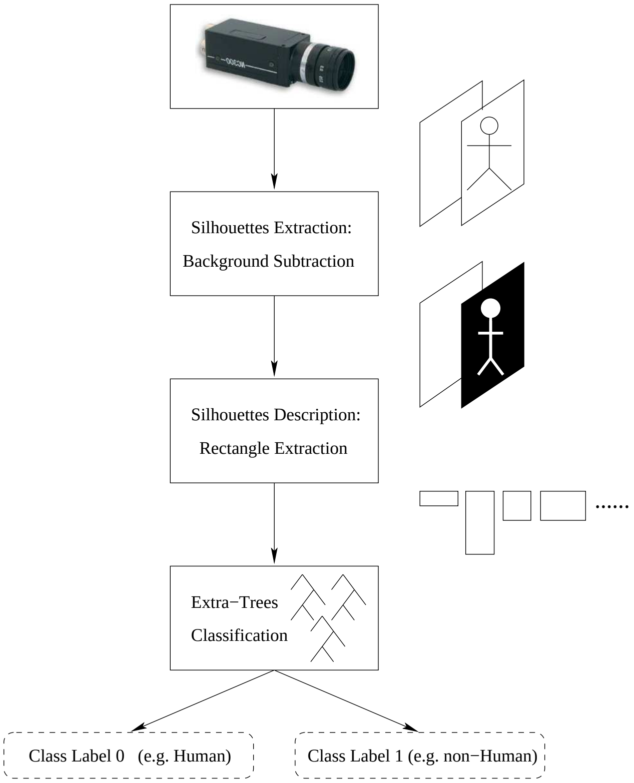

Fig. 1. Overall architecture.

3 Extraction of silhouettes

The first step of our system consists in the segmentation of the input video stream in order to produce binary silhouettes, which will be fed into the silhouettes description module. We achieve this by a motion segmentation based on an adaptive background subtraction method.

Background segmentation methods are numerous and one can find a survey of the existing methods in [10]. The method we have chosen is based on an adaptive modeling of each pixel as a mixture of Gaussians, each of which corresponds to the probability of observing a particular intensity or color for this pixel. In each Gaussian cluster, the mean accounts for the average color or intensity of the pixel, whereas the variance is used to model illumination variations, surface texture, and camera noise. The whole algorithm relies on the assumptions that the background is visible more frequently than the foreground and that its variance is relatively low, which are common assumptions for any background subtraction technique. Extensive description of the algorithm can be found in [11,12] and a tutorial is available at [13]. The technical description is given hereafter.

If X_t is the color or intensity value observed at time t for a particular pixel in the image, the history $\{X_1, \ldots, X_t\}$ is modeled as a mixture of K Gaussian distributions. The probability of observing a particular color or intensity value at time t is expressed as

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \tag{1}$$

where

- K is the number of Gaussian clusters used to model the history of the pixel,
- $\omega_{i,t}$ is the weight associated with the ith cluster at time t –it models the amount of data represented by the ith Gaussian–,
- $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean and covariance matrix of the ith Gaussian, and
- η is a Gaussian probability density function.

For computational efficiency reasons, the covariance matrix $\Sigma_{i,t}$ is assumed to be isotropic and diagonal

$$\Sigma_{i,t} = \sigma_k^2 \mathbf{I}. \tag{2}$$

The Gaussian distributions are sorted in decreasing order of the ratio $\frac{\omega_{i,t}}{\sigma_{i,t}}$. The j first Gaussians are considered to account for the background, while the rest of them accounts for the foreground. The j factor is dynamically estimated by accumulating the $\omega_{i,t}$ values, according to the computed order of the Gaussians, until a given threshold value T is reached. For this to work we assume that the background is visible more often than the foreground and that its variance is relatively low.

Every new pixel value is checked against the K distributions until a match is found, in which case the pixel receives its class label (background or foreground) according to that of the matched distribution. A match is defined as a pixel value within 2.5 times the standard deviation of a distribution. If no match is found, the pixel is considered as belonging to the foreground. In this case, a new distribution, centered on the pixel color or intensity, is initialized to replace the weakest distribution present in the mixture model. This new distribution is of high initial variance and low prior weight.

Once the new pixel value is classified, the model has to be updated. A standard method would be to use the *expectation maximisation* algorithm. Unfortunately, that would be prohibitively computationally expensive. In [11,12], STAUFFER and GRIMSON give an on-line K-means approximation efficient enough to be performed in real-time on a standard VGA image (640×480 pixels). They propose the following rule to update the weights

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}), \tag{3}$$

where α is the learning rate, and $M_{k,t}$ equals 1 for the matched cluster and 0 for the remaining ones. The μ and σ parameters of the unmatched Gaussian distributions are kept unchanged while those of the matched distribution are adjusted as follows

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t, \tag{4}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t-\mu_t)^T(X_t-\mu_t), \tag{5}$$

where

$$\rho_t = \alpha\eta(X_t|\mu_t, \sigma_{t-1}) \tag{6}$$

can be replaced by a constant ρ to reduce the computational load.

After the foreground has been computed by subtraction, foreground pixels are aggregated by a connected component algorithm. This guarantees that a unique label is assigned to each connected region. Then each connected region is considered as a distinctive input for both the silhouettes description and silhouettes classification modules. Examples of extracted candidate silhouettes by the mixture of Gaussians algorithm are shown on Figure 2.

In the silhouettes description module, each candidate silhouette will be handled as if it was the unique region in the image. There are thus as many silhouettes as connected regions for which an algorithm has to decide whether or not it belongs to a known shape pattern.

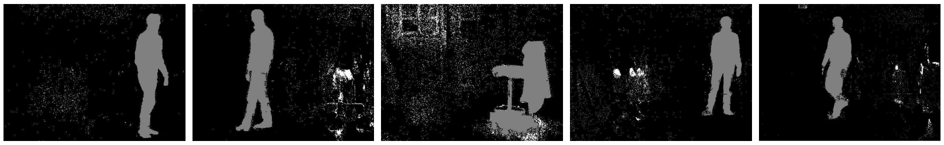

Fig. 2. Examples of extracted silhouettes with the Gaussian mixture model background subtraction technique.

4 Features based on a granulometric description by rectangles

Most surfacic descriptors can be described in terms of the theory of mathematical morphology. Therefore we will use this framework to describe our new feature set.

After a brief introduction to some notations, we will present the framework of granulometries that proved to be the starting point of our development. Then we provide a formal description of our new operator.

4.1 Morphological operators on sets

Hereafter we briefly recall some definitions and notations used in mathematical morphology that serves as the framework to define our new feature space. Consider a space $\mathcal{E}$, which is the continuous Euclidean space $\mathbb{R}^n$ or the discrete space $\mathbb{Z}^n$, where $n \geq 1$ is an integer. Given a set $X \subseteq \mathcal{E}$ and a vector $b \in \mathcal{E}$, the translate X_b is defined by $X_b = \{x+b | \, x \in X\}$.

Let us take two subsets X and B of $\mathcal{E}$. MINKOWSKI defined the addition and subtraction of these sets, respectively as

$$X \oplus B = \bigcup_{b \in B} X_b = \bigcup_{x \in X} B_x = \{x + b | \, x \in X, \, b \in B\} \tag{7}$$

$$X \ominus B = \bigcap_{b \in B} X_{-b} = \{p \in \mathcal{E} | \, B_p \subseteq X\}. \tag{8}$$

For $X \oplus B$, X and B are interchangeable, but X and B play a different role in the case of $X \ominus B$. Therefore B is referred to as the *structuring element*, and we call $X \oplus B$ and $X \ominus B$ respectively the *dilation* and *erosion* of X by B.

Dilation and erosion are not inverse operators. If X is eroded by B and then dilated by B, one may end up with a smaller set than the original set X. This set, denoted by $X \circ B$, is called the *opening* of X by B and defined by $X \circ B = (X \ominus B) \oplus B$. The geometric interpretation of an opening is that it is the union of all translated versions B included in X, or in mathematical terms, $X \circ B = \{B_p | \, p \in \mathcal{E}, \, B_p \in X\}$. Note that this geometrical interpretation is valid for a given set of fixed size. We have to enlarge it to encompass the notion of size or family of structuring elements, which leads us to granulometries.

4.2 Granulometries

The concept of granulometry was introduced by MATHERON [14]. It is based on the following definition.

Let $\mathit{\Psi} = (\psi_\lambda)_{\lambda \geq 0}$ be a family of image transformations depending on a parameter λ. This family constitutes a granulometry if and only if the following properties are satisfied:

$$\forall \lambda \geq 0,\ \ \psi_\lambda \text{ is increasing} \tag{9}$$

$$\forall \lambda \geq 0,\ \ \psi_\lambda \text{ is anti-extensive} \tag{10}$$

$$\forall \lambda \geq 0,\ \mu \geq 0,\ \ \psi_\mu \psi_\lambda = \psi_\lambda \psi_\mu = \psi_{\max(\lambda,\mu)}. \tag{11}$$

The third property implies that, for every $\lambda \geq 0$, ψ_λ is an idempotent transformation, that is: $\psi_\lambda \psi_\lambda = \psi_\lambda$. As these properties reflect those of an opening, openings fit nicely in this framework as long as we can order the openings with a scalar. For example, assume that $X \circ rB$ is the opening by a ball of radius r. Then $\mathit{\Psi} = (\psi_r)_{r \geq 0} = (X \circ rB)_{r \geq 0}$ is a granulometry. Of particular interest are granulometries generated by openings by scaled versions of a convex structuring element.

Granulometries, and some measures taken of them, have been applied to problems of texture classification [15], image segmentation, and more recently to the analysis of document images [16].

4.3 Granulometric curves and features

MARAGOS [15] has described several useful measurements for granulometries defined by a single scale factor: the *size distribution* and the *pattern spectrum*. The size distribution is a curve that gives the probability of a point belonging to an object to remain into that object after openings with respect to a size factor. The pattern spectrum is defined likewise as the derivative of the size distribution. Note that other measures than the area have been suggested as well. They have led to the concepts of openings by attributes [17].

All these measures are taken on operator residues driven by a one-dimensional criterion. They are not applicable to a family of arbitrary structuring elements nor are they capable to produce uncorrelated multi-dimensional features. Therefore we define a new operator that produces a *cover*.

Definition 1. [Cover] *Let $\mathcal{S}$ be a family of I arbitrary structuring elements $\mathcal{S} = \left\{S^{i \in I}\right\}$. A* cover *of a set X* by *$\mathcal{S}$ is defined as a non-redundant union of translated elements of $\mathcal{S}$ that are included in X. More precisely, if $C(X)$ is a cover of X then*

$$C(X) = \left\{S^j_z \middle|\, z \in \mathcal{E} \text{ and } S^j \in \mathcal{S}\right\} \tag{12}$$

where, if $S^{j'}_{z'}$ and $S^{j''}_{z''}$ both belongs to $C(X)$, none of them is totally included in the other one.

As a consequence of this definition, any element of $C(X)$ comprises at least one pixel that uniquely belongs to it. But the upper bound of uniquely covered pixels can be as large as the area of element S^j.

In our application we consider the simplest two-dimensional opening which is of practical interest and practically tractable: an opening by a rectangular structuring element $B = mH \oplus nV$ where mH, nV respectively are m-wide horizontal and n-wide vertical segments [1]. Based on the family $\mathcal{B}$ of all possible rectangle sizes, $C(X)$ will be the union of all the largest non-redundant rectangles included in X. Such rectangles are shown in Figure 3. A fast algorithm for computing this cover is given in [18].

[1] Note that ordered openings with respect to m or to n constitute granulometries. However mixing m and n size factors do not generate granulometries.

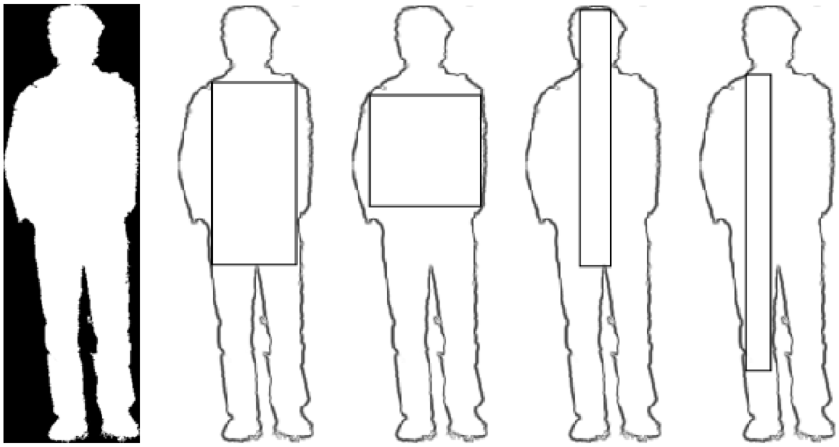

Fig. 3. Examples of largest wedged rectangles contained in a human silhouette.

The main advantage of using a cover is that we have a family of structuring elements, describing the surface of an object X, whose members might overlap but all of them uniquely fit somewhere inside of X.

The next step is to extract features from the cover. Since B are rectangles, features like width, height, perimeter, and area spring to mind. For classification purposes however care should be taken to avoid redundant features because it would not increase the performances and could even be counterproductive. For example, HADWIGER [19] has shown that any continuous, additive, and translation and rotation invariant measure on a set X must be a linear combination of the perimeter, area, and EULER-POINCARRÉ number of X.

Since translations and scales have some significance in the analysis of silhouettes we reintroduce them by taking the position of elements of $C(X)$ relative to the center of X. Finally we select 5 features on any elements of $C(X)$: width, height[2], 2 relative coordinates of its center, and the percentage of uniquely covered pixels to its area. This last feature is a consequence of taking the cover $C(X)$ to describe X.

5 Silhouettes classification

Once the set of all the features describing a silhouette has been extracted (see Figure 4 for an illustration of the density of covered pixels), it becomes possible to exploit a machine learning algorithm to map this set into a class. Indeed, such mappings are especially hard to derive by hand and should be *learned* by the system. In our framework, as we are interested in the detection of human silhouettes, only two classes of interest are considered: the class of the human silhouettes, and the class of any other silhouette.

The machine learning approach requires to take two difficulties into account: (1) there is a need of a classifier with excellent generalization abilities not subject to overfitting, and (2) we must define a way to apply this classifier on a set of rectangles, the number of which may widely vary between silhouettes.

To this aim, we propose to use *EXTremely RAndomized* trees (Extra-trees), a fast, yet accurate and versatile machine learning algorithm [20]. Reasons for using extra-trees in our context are threefold: (1) extra-trees have proven successful for solving some color image

[2] Note that the perimeter and area derives from the width and height of a rectangle so that it is unnecessary to add them the list of features.

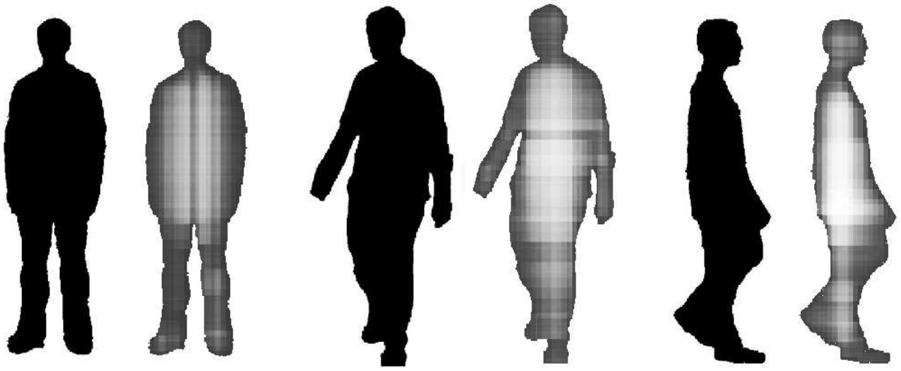

Fig. 4. Examples of rectangles size distributions for human shaped silhouettes. The pixel intensities account for the number of overlapping rectangles that cover each location in the image.

classification tasks [21], (2) they form a non-parametric function approximation architecture, which do not require previous knowledge, and (3) they have low bias and variance, as well as good performances in generalization.

5.1 Classification based on extremely randomized trees

We first describe how extra-trees can be used to map a single rectangle to a class. Then we will explain how to map a *set* of rectangles to a class. We will restrict our study of extra-trees to the case where all the input attributes are numerals, which is obviously the case of our rectangular features. Indeed, as mentioned earlier, the input attributes for the rectangles are their width, height, relative positions, and an information about the cover.

Intuitively, extra-trees can be thought of as a crossover between *bagging* [22] and *random forests* [23]. They consist in a forest of M independent binary decision trees. Each of their internal nodes is labeled by a threshold on one of the input attributes, that is to be tested in that node. As for the leaves, they are labeled by the classification output. To classify a rectangle through an extra-tree model, this rectangle is independently classified by each tree. This is achieved by starting at the root node, then progressing down the tree according to the result of the tests on the threshold found during the descent, until a leaf is reached. Doing so, each sub-tree votes for a class. Finally, the class that obtains the majority of votes is assigned to the rectangle.

The sub-trees are built in a top-down fashion, by successively splitting the leaf nodes where the output variable does vary. For each input variable, the algorithm computes its variation bounds and uniformly chooses one random threshold between those bounds –this is similar to the case of random forests. Once a threshold has been chosen for every input variable, the split that gives the best information-theoretic score on the classification output is kept –this is similar to bagging. This will guarantee that the variance in the model is reduced (thanks to the presence of a forest of independent sub-trees), as well as bias (thanks to the random selection of the thresholds), while taking advantage of an information measure that guides the search for good splits.

Pseudo-code describing how an extra-tree model can be built from a set of rectangles is given in Algorithm 1. In this pseudo-code, $A \in \mathbb{R}^{m \times n}$ is a the table containing the matrix of input attributes in the learning set, $\boldsymbol{b} \in \mathbb{B}^m$ is the observed classification output for this learning set [3], and M is the number of sub-trees in the forest. We assume the existence of a

[3] We use the symbol $\mathbb{B}$ to represent the set of Booleans: the *true* symbol corresponds to the class of human silhouettes, and the *false* symbol to that of other silhouettes.

Algorithm 1 — General structure for the learning of extra-trees

```
extra-trees($A, \boldsymbol{b}, M$) :–
    $\mathcal{T} := \phi$
    for $i := 1$ to $M$ do
        $\mathcal{T} := \mathcal{T} \cup \{$subtree$(A, \boldsymbol{b})\}$
    end for
    return $\mathcal{T}$
```

Algorithm 2 — Recursive induction of one single sub-tree

```
subtree($A, \boldsymbol{b}$) :–
    $\mathcal{V} := \phi$
    for $v := 1$ to $n$ do
        $r := \min_i\{A_{i,v}\}$
        $s := \max_i\{A_{i,v}\}$
        if $r \neq s$ then
            $\mathcal{V} := \mathcal{V} \cup \{v\}$
            $t[v] :=$ random value in $[r, s]$
            $s[v] :=$ score$(A, \boldsymbol{b}, v, t[v])$
        end if
    end for
    if $|\mathcal{V}| > \sqrt{n}$ then
        $\mathcal{V} :=$ randomly draw $\sqrt{n}$ variables without replacement out of $\mathcal{V}$ [4]
    end if
    $v^* := \text{argmax}_{v \in \mathcal{V}}\{s[v]\}$
    if $s[v^*] = 0$ then
        $o := \text{argmax}_{c \in \mathbb{B}} |\{i \mid b_i = c\}|$
        return a leaf labeled with output class $o$
    else
        $i_\oplus := \{i \mid A_{i,v*} < t[v^*]\}$
        $i_\ominus := \{i \mid A_{i,v*} \geq t[v^*]\}$
        $T_\oplus :=$ subtree$(\{A_i \mid i \in i_\oplus\}, \{b_i \mid i \in i_\oplus\})$
        $T_\ominus :=$ subtree$(\{A_i \mid i \in i_\ominus\}, \{b_i \mid i \in i_\ominus\})$
        return a binary decision node $\langle t[v^*], T_\oplus, T_\ominus \rangle$
    end if
```

function score$(A,\ \boldsymbol{b},\ v,\ t)$ that gives the information-theoretic score of the threshold t on the variable v in the database $(A,\ \boldsymbol{b})$. In our case, Shannon's information gain was used as the $score$ function.

5.2 Classification of silhouettes

We have just described the process of classifying *one* rectangle. But a silhouette X is described by a *set* $C(X)$ of rectangles. Furthermore, two distinct silhouettes can have a different number of rectangles inside them. We must therefore introduce a meta-rule over the extra-trees for mapping a set $C(X)$ to a class. In this work, we exploit an idea that is similar to that of Marée *et al.*, which was used in the context of image classification [21].

Let M be a fixed positive integer. Given the set $C(X)$ of rectangles that shapes the silhouette X, we select the first M rectangles inside this set, which induces a subset $C_M(X) \subseteq C(X)$. Then, we apply the extra-trees model onto each rectangle inside $C_M(X)$. This process generates one vote per rectangle. Finally, the silhouette X is mapped into the class that has obtained the majority of the votes.

[4] This line determines the strength of the attribute selection process. The choice of $\sqrt{n}$ is discussed in [20].

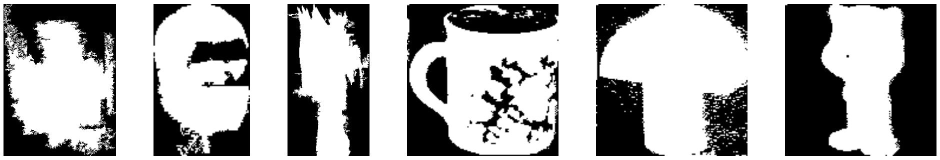

Fig. 5. A few examples of negative instances contained in the training dataset.

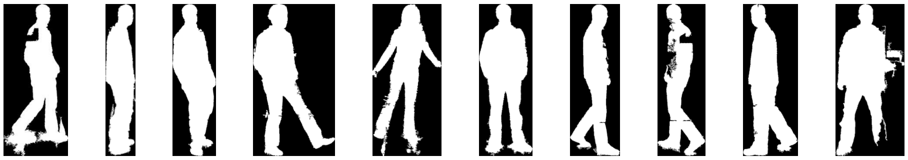

Fig. 6. Subset of positive instances contained in the training dataset.

6 Experimental results

6.1 Dataset collection

As mentioned in the introduction, we have focused our experiments on the detection of human silhouettes in a video stream. The extra-trees have been trained on a dataset of silhouettes that contains both silhouettes of human bodies and silhouettes of other kind of objects. We have fed the learning set with a large number of instances for each of those two classes.

Some instances of non-human silhouettes, called negative instances, are displayed in Figure 5. The negative samples are the union of non-human silhouettes that were extracted from a live video stream by the background subtraction technique presented in Section 3, and of images that were taken from the COIL-100 database [24]. There are about 12,000 images in this dataset. As for the positive instances, we have about 3,000 human silhouettes. Some of them are represented in Figure 6. Those two datasets have been converted to a database that has been fed into the extra-trees learning algorithm (cf. Section 5).

6.2 Tests on real-world images

We have tested our algorithms on a color video stream of 640×480 pixels that was captured with a FireWire CCD camera. The whole process (including silhouettes extraction, description, and classification) was carried out at approximately five frames per second on a Pentium IV computer at 3.4 GHz.

The detection of human silhouettes is very robust since the number of correct classifications largely outnumbers misclassifications, although we ignored any correlation between successive frames. Example images of correct (resp. wrong) classifications are shown in Figure 7 (resp. in Figure 8). Our method might be subject to improvements, one of them being the use of a prediction scheme between successive frames, but these first results demonstrate that on single images our system is capable to recognize specific silhouettes in a semi-controlled environment.

7 Conclusions

In this paper we propose a new system for the real-time detection and classification of binary silhouettes. The silhouettes are extracted from an input video stream using a standard background subtraction algorithm. Then, each silhouette is treated by a new kind of

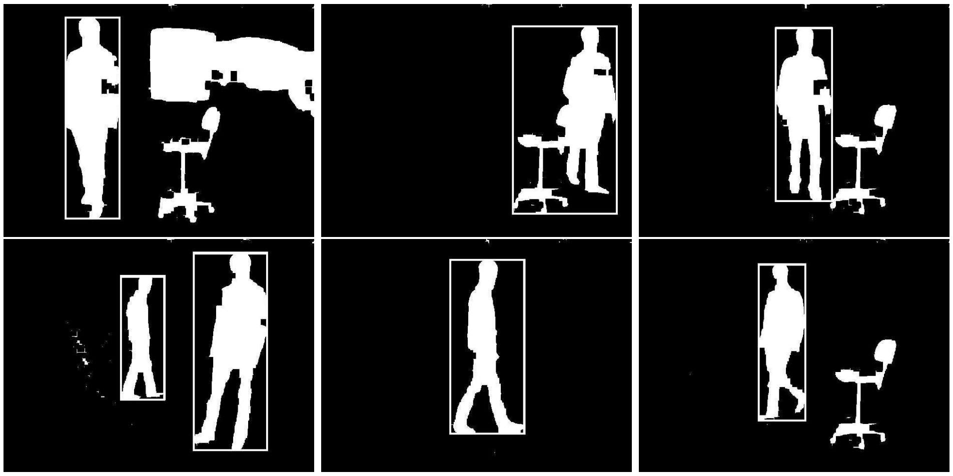

Fig. 7. Examples of silhouettes classified correctly. A white frame around an object indicates that the system classifies it as a human silhouette.

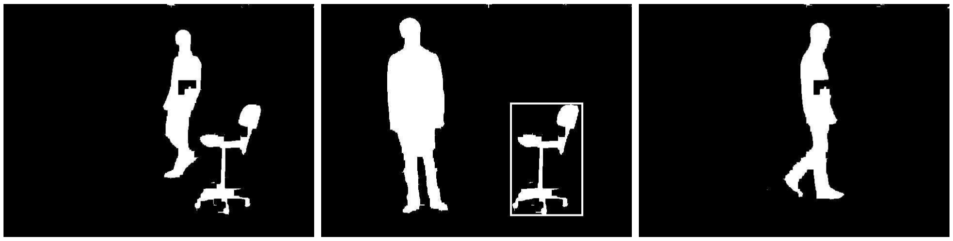

Fig. 8. Examples of misclassified silhouettes.

granulometric filter that produces a morphological cover of the silhouette and characterizes it as the set of all the largest rectangles that can be wedged inside of it. One of the major achievements is that we managed to implement the feature extraction step in real-time, which is uncommon for surface-based descriptors. The rectangle features are then fed into an extra-trees classifier that assigns a class label to each detected silhouette. Thanks to the simple tree-based structure of extra-trees, the classification step is also very fast. As a consequence, the whole process that consists of silhouettes detection, analysis and classification can be carried out in real-time on a common computer.

Empirical results that consisted in the application of our method to images captured with a CCD camera put in an environment unknown to the learning process show that our method manages to detect human silhouettes with a high level of confidence. Future work will feature a systematic evaluation of the performances of our approach. We will also investigate its exploitation in more complex tasks such as gait recognition, human tracking, or even general object tracking.

References

1. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. International Journal of Computer Vision **37** (2000) 151–172
2. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., Madison (WI, USA) (2003) 257–263
3. Schmid, C., Mohr, R.: Local greyvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 530–535

4. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
5. Mathes, T., Piater, J.: Robust non-rigid object tracking using point distribution models. In: Proc. of the British Machine Vision Conference, Oxford (UK) (2005) 849–858
6. Boulgouris, N., Hatzinakos, D., Plataniotis, K.: Gait recognition: a challenging signal processing technology for biometric identification. ispmag **22** (2005) 78–90
7. Serra, J.: Image analysis and mathematical morphology. Academic Press, New York (1982)
8. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T.: Pedestrian detection using wavelet templates. In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, IEEE Computer Society (1997) 193
9. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 780–785
10. Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: A systematic survey. IEEE transactions on image processing **14** (2005) 294–307
11. Stauffer, C., Grimson, E.: Adaptive background mixture models for real-time tracking. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. (1999) 246–252
12. Stauffer, C., Grimson, E.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 747–757
13. Power, P., Schoonees, J.: Understanding background mixture models for foreground segmentation. In: Proc. Images and Vision Computing, Auckland, NZ (2002)
14. Matheron, G.: Eléments pour une théorie des milieux poreux. Masson, Paris (1967)
15. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **11** (1989) 701–716
16. Bagdanov, A., Worring, M.: Granulometric analysis of document images. In IEEE, ed.: Proceedings of the International Conference on Pattern Recognition, Volume I. (2002) 478–481
17. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding **64** (1996) 377–389
18. Van Droogenbroeck, M.: Algorithms for openings of binary and label images with rectangular structuring elements. In Talbot, H., Beare, R., eds.: Mathematical morphology. CSIRO Publishing, Sydney, Australia (2002) 197–207
19. Hadwiger, H.: Vorlesungen über inhalt, oberfläche and isoperimetric. Springer Verlag (1957)
20. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning **36** (2006) 3–42
21. Marée, R., Geurts, P., Piater, J., Wehenkel, L.: Random subwindows for robust image classification. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1., San Diego (CA, USA) (2005) 34–40
22. Breiman, L.: Bagging predictors. Machine Learning **26** (1996) 123–140
23. Breiman, L.: Random forests. Machine Learning **45** (2001) 5–32
24. Nene, S., Nayar, S., Murase, H.: Columbia object image library: Coil-100. Technical report (1996)