Iterative Human Segmentation from Detection Windows using Contour Segment Analysis

Cyrille Migniot, Pascal Bertolino and Jean-Marc Chassery
CNRS Gipsa-Lab DIS, 961 rue de la Houille Blanche, BP 46-38402, Grenoble Cedex, France

Keywords: Pedestrian, Segmentation, Silhouette, Contour Segment, Oriented Graph.

Abstract: This paper presents a new algorithm for human segmentation in images. The human silhouette is estimated in positive windows that are already obtained with an existing efficient detection method. This accurate segmentation uses the data previously computed in the detection. First, a pre-segmentation step computes the likelihood of contour segments as being a part of a human silhouette. Then, a contour segment oriented graph is constructed from the shape continuity cue and the prior cue obtained by the pre-segmentation. Segmentation is so posed as the computation of the shortest-path cycle which corresponds to the human silhouette. Additionally, the process is achieved iteratively to eliminate irrelevant paths and to increase the segmentation performance. The approach is tested on a human image database and the segmentation performance is evaluated quantitatively.

1 INTRODUCTION

Human segmentation is of fundamental interest in computer vision due to the variations in human pose and clothing. Moreover, an accurate segmentation is needed in many applications such as human-computer interaction, video indexing, image editing or movie special effects.

Recognizing person can't be done from color and texture. On the contrary, since a person can be recognized only from its silhouette, shape is a more descriptive cue. In the proposed method, the contour map of an image is obtained with the Canny's operator (see Figure 1(b)). Nearly linear contour fragments are modeled with segments (see Figure 1(c)) which are relevant parts of the image for our study. Indeed, the human silhouette can be rebuilt from them. The segmentation is then performed by a reconstruction of the silhouette from the contour segments.

Traditionally, when the detection and the segmentation are performed simultaneously, the detection process is chosen to be well-adapted to the segmentation. Conversely, we aim at realizing the segmentation from one of the most efficient existing detection method. Giving good performance, the Dalal's algorithm (Dalal and Triggs, 2005) based on Histograms of Oriented Gradients (HOG) descriptor with Support Vector Machine (SVM) classifier is used in numerous papers (Felzenszwalb et al., 2010) (Alonso et al., 2007) (Bertozzi et al., 2007) (Zhu et al., 2006).

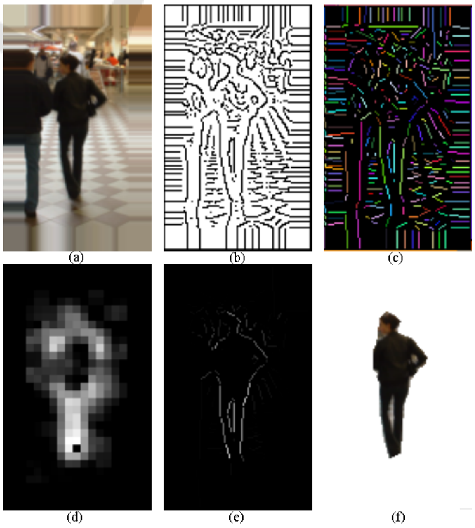


Figure 1: A detection window containing a pedestrian (a), contour image computed with the Canny's algorithm (b), contour pixels gathered in contour segments (c), cells likelihood provided by SVM (d), likely segments computed by the pre-segmentation (e) and segmented silhouette obtained by our method (f).

In our work, the segmentation is carried out from the detection. Indeed, the Dalal's detection provides detection windows (see Figure 1(a)) from the computation of the HOG. Then, our method uses these HOG

Migniot C., Bertolino P. and Chassery J..
Iterative Human Segmentation from Detection Windows using Contour Segment Analysis.
DOI: 10.5220/0004209404050412
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 405-412
ISBN: 978-989-8565-47-1

to achieve the segmentation in the detection windows. This novel proposed segmentation method is composed of two steps:

Firstly, the HOG and SVM detection process computed for the whole window in (Dalal and Triggs, 2005) is used in sub-parts of the window to provide more local shape information. The likelihood of each contour segment of the window as being a part of a human silhouette is computed and gives a pre-segmentation (see Figure 1(e)) where the gray level of a segment is proportional to its likelihood).

Secondly, the contour segment cycle that is the most representative of a pedestrian is obtained by a Dijkstra's algorithm in an oriented graph. This graph is made with the contour segments as vertices and the neighborhood between couple of close contour segments as edges. The integration of the knowledge on the researched class (here the pedestrians) is obtained by weighting the edges of the graph with the pre-segmentation data. The optimal cycle finally gives the human silhouette and provides the segmentation (see Figure 1(f)).

Due to the human shape complexity, errors frequently appear in the obtained results. Nevertheless, some of them can be easily located. To this end the process depicted above is iterated in the problematic areas with updated graph features. Thus, each iteration may improve the result.

The remainder of the paper is organized as follows: Section 2 reviews the human segmentation and the use of contour segments. Section 3 describes the pre-segmentation process. Section 4 presents the oriented graph approach used for segmentation. Section 5 develops the iterative algorithm. Experimental results are presented in Section 6, followed by conclusions in Section 7.

2 RELATED WORK

Human Detection and Segmentation. The descriptor and classifier combination is the most used framework in human detection. The descriptor converts an image into a vector of discriminative features and the classifier compares the features of a tested image to the features of images of an annotated database. HOG (Dalal and Triggs, 2005) and Haar wavelets (Oren et al., 1997) are the most used descriptors. SVM (Vapnik, 1995) and Adaboost (Freund and Schapire, 1995) are the most used classifiers.

Simultaneous detection and segmentation can stem from the research of the region of interest (ROI) which can be based on depth (with stereo as in (Kang et al., 2002)) or color (by normalized cut as in (Mori et al., 2007)). Otherwise, the silhouette can be found by a template matching (Lin and Davis, 2010) (Munder and Gavrila, 2006) where the image contours are compared to the silhouettes of a codebook. The relevance of ROI or the similarity to a template of the codebook gives the detection. Gathering of the ROI or finding template delineates the silhouette and also achieves the segmentation.

Hernandez (Hernandez et al., 2010) performs face detection and skin color model for seed initialization in a graph cut process. This initialization is provided by a previously computed pose estimation in (Pishchulin et al., 2012). Wang (Wang and Koller, 2011) finally minimizes an energy that simultaneously takes into account the body parts localization and the segmentation.

Contour Segment Approaches. As silhouette shape is well-descriptive of the human class, there is a range of methods based on the analysis of its parts. Indeed Shotton (Shotton et al., 2008) demonstrates that a few number of fragments of outline contours permit human recognition. For segmentation, Ferrari (Ferrari et al., 2006) focuses on the succession of descriptive contour segments. Wu (Wu and Nevatia, 2007) builds a classifier to recognize human parts from edgelet features (detection) and a classifier to recognize the foreground pixels (segmentation). The two classifiers are used together to carry out the two processes simultaneously. Gao (Gao et al., 2009) generates from the contour a feature named Adaptive Contour Feature that at the same time defines a weak classifier for human detection and segmentation. Hariharan (Hariharan et al., 2011) combines information from different part detectors to classify category-specific object contours. Lastly, Sharma (Sharma and Davis, 2007) finds the relevant contour segment cycles from an oriented graph. Then, the cycles are integrated in a Markov Random Field and a graph cut selects the one which are related to silhouette and achieves the segmentation.

We want that segmentation deals with an usual and efficient detection method. Our approach, which inversely as (Sharma and Davis, 2007), searches the prior cue first and then the cycle, is so well-adapted.

3 PRE-SEGMENTATION

In (Dalal and Triggs, 2005), the HOG and SVM combination allows the detection. For each detection window, the only obtained information is the decision about the presence of a person in the window. Segmentation needs a process at a smaller scale. The detection window is partitioned in square areas named *cells*. The HOG of each cell is computed. Regular cell gatherings are formed and named *block*. Thus, several HOG are associated to a single block to increase the descriptiveness of its features. Then, SVM gives a classification for each block and provides a value S^{SVM}_{block} which corresponds to the likelihood of elements contained by the block as being a part of the human silhouette. Blocks overlap so each cell belongs to several blocks. The value S^{SVM}_{cell} is associated to each cell (Figure 1(d)). It is the mean of the classification values for all the blocks that contain the cell.

The classification gives information on each cell but our study needs to deal with elements directly related to the silhouette. The Canny's algorithm determines the contour pixels (Figure 1(b)) which are gathered in segments (Figure 1(c)). We aim to reconstruct the silhouette from these segments. For each contour segment *seg*, L_{seg} is the likelihood of the segment as being a part of the silhouette (Figure 1(e)). It is defined by:

$$L_{seg} = \underset{p\in seg}{mean}(G(p)S^{SVM}_{c_p}) \qquad (1)$$

where c_p is the cell that contains the pixel p and $G(p)$ is the intensity of gradient of the pixel p.

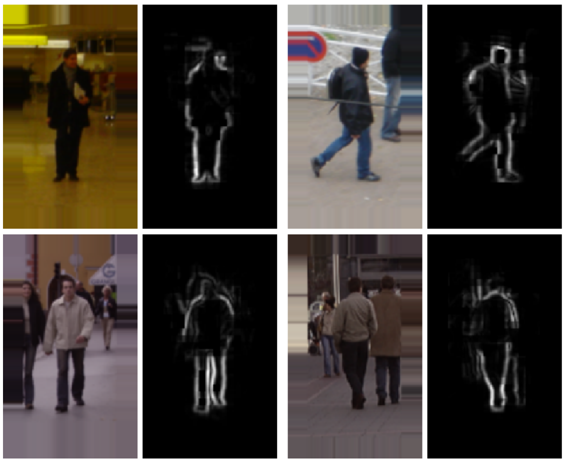


Figure 2: Product of the likelihood provided by SVM with the gradient of the image for 4 examples. These values used in equation 1 give accurate clues on the silhouette contour.

These data provide information that guides the segmentation (Figure 2). A pre-segmentation of the person is achieved.

4 SILHOUETTE REFORMATION

The most likely silhouette segments must be linked to form the contour silhouette. Similarly to (Elder and Zucker, 1996), an oriented graph from contour parts (here contour segments) is studied. The graph edges are weighted so as to give silhouette continuity and its likelihood as human one. Hence, the likelihood computed during the pre-segmentation step will be integrated in the process.

4.1 Building the Graph of Contour Segments

We create an oriented graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ in which a path is a sequence of connected contour segments. The weights are set so that the searched shortest path corresponds to the silhouette.

A contour map of the detection window is obtained with the Canny's edge detector and is vectorized to provide the contour segments which are the vertices $\mathcal{V}$ of the graph.

Nevertheless, some contours of the silhouette may be absent due to lack of local contrast and introduce gaps in the silhouette (Figure 3(a)). As in (Elder and Zucker, 1996), contour need to be closed. Additional segments called *transitions* are so introduced. They connect the extremities of the contour segments (Figure 3(b)). These transitions are the edges $\mathcal{E}$ of the graph. Connecting the right segments with the right transitions requires large computing. Moreover, contour segments that are spatially far have little probability to be consecutive in the silhouette path. Consequently, only the transitions whose size is under a determinate threshold $\mathcal{T}_t$ make edges in $\mathcal{E}$. The sequence of vertices and edges of a path in $\mathcal{G}$ represents a sequence of contour segments and transition segments which gives a silhouette.

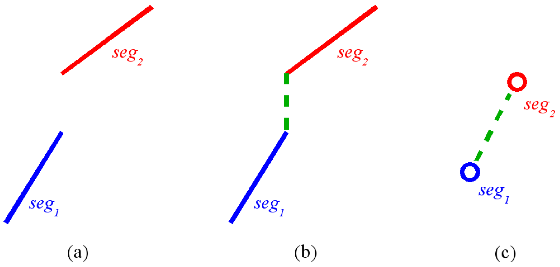


Figure 3: The lacks in the contour detection generate gaps in the silhouette (a). A transition (dotted line) connects the extremities of two existing segments (b). The corresponding piece of graph (c).

Finally, weights are associated to the edges. To do it, a local interaction term is defined: the **affinity** is related to the probability of a contour segment seg_2 to follow a contour segment seg_1. It associates two notions: continuity and likelihood.

Continuity corresponds to the path coherence from seg_1 to seg_2. It is usually related to the spatial distance, the magnitude difference or the orientation difference between the two segments. Here, since only the contours are studied, the magnitude difference is not available. Moreover, the irregularities of human silhouette prevent from using orientation continuity. Consequently, only the spatial distance is used.

Likelihood takes into account the knowledge on the searched class. If a segment is likely to be a part of a human silhouette, it should be promoted.

The affinity of the path from segment seg_1 to segment seg_2 is firstly defined by:

$$Affinity(seg_1, seg_2) = e^{-\frac{1}{L_c}} e^{-\alpha S_t} \tag{2}$$

where L_c is the likelihood of seg_2 as defined in equation 1, S_t is the size of the transition segment that connects seg_1 to seg_2 and α is related to the transition segment influence.

However, the size of the contour segments is various. The long segments must be more weighted in the graph because they mean important parts in the path. Moreover, the transition segments which are unlikely to be a part of a human silhouette may be penalized. Thus, the definition of affinity is modified as follows:

$$Affinity(seg_1, seg_2) = e^{-\frac{S_c}{L_c}} e^{-\alpha \frac{S_t}{L_t}} \tag{3}$$

where L_t is the transition segment likelihood as defined in equation 1 and S_c is the size of seg_2.

Inverse logarithm is finally used to compute the weight ω associated to the edge between the two contour segments.

$$\omega = -log(Affinity(seg_1, seg_2)) \tag{4}$$

4.2 The Silhouette as an Optimal Cycle

Once the graph is built, the segmentation is seen as a shortest-path problem where the goal is, starting from a confidence segment, to find the shortest path in the graph (in terms of edge weights) that makes a cycle. To do so, we use the well known Dijkstra's algorithm.

Since human silhouette is complex and because cumulative weights are used, the Dijkstra's algorithm promotes spatially short paths that can miss large parts of the silhouette. To avoid this bias the path is forced to pass through the two spatially farthermost segments. So, these are actually two shortest paths linking these two segments that are searched. The concatenation of these two paths then provides the optimal cycle. As we are dealing with pedestrian, it is assumed that these segments correspond to the top of the head (top) and the bottom of the feet ($bottom$).

top and $bottom$ are found automatically. Their choice is made using the location, orientation and likelihood of the segments. For a segment seg, let (x,y) be the coordinates of its middle, θ its orientation and L its likelihood. A Gaussian function whose parameters are set experimentally is defined for each of these four features f:

$$G_{\mu,\sigma}(f) = e^{-\frac{(f-\mu)^2}{2\sigma^2}} \tag{5}$$

where μ is the mean value of the feature in the dataset and σ its standard variation.
Then the probabilities $P_t(seg)$ to be an appropriate top segment and $P_b(seg)$ to be a bottom segment are defined by:

$$\begin{cases} P_t(seg) = G_{\mu^t_x,\sigma^t_x}(x).G_{\mu^t_y,\sigma^t_y}(y).G_{\mu^t_\theta,\sigma^t_\theta}(\theta).G_{\mu^t_L,\sigma^t_L}(L) \\ P_b(seg) = G_{\mu^b_x,\sigma^b_x}(x).G_{\mu^b_y,\sigma^b_y}(y).G_{\mu^b_\theta,\sigma^b_\theta}(\theta).G_{\mu^b_L,\sigma^b_L}(L) \end{cases} \tag{6}$$

The segments that maximize these probabilities are chosen:

$$\begin{cases} top = \underset{seg}{\text{argmax}} P_t(s) \\ bottom = \underset{seg}{\text{argmax}} P_b(s) \end{cases} \tag{7}$$

aa

5 ITERATIVE PROCESS

When some contours of the silhouette are missing, the transitions between successive segments in the optimal cycle may be long. But, the threshold $\mathcal{T}_t$ defined in Section 4.1 prevents from too long transitions. Moreover, the Dijkstra's algorithm does not adapt perfectly the pre-segmentation. New iterations of the process with the data and the models of the pre-segmentation are so achieved to improve the segmentation. They are applied on wrong parts of the segmentation with more adapted features. The iterative process is summed up in Figure 5.

Segmentation Evaluation. The segments of the cycle provide a segmentation mask that is presented to the SVM classifier already used in the pre-segmentation step. The likelihood L_{seg} of each one is calculated using equation 1.

Updating the Cycle. The segments of the cycle whose likelihood is under a threshold $\mathcal{T}_e$ are considered to be wrong. For each sequence of successive wrong segments in the cycle, a shortest path search is done using a relaxed threshold and locally updated weights: the threshold $\mathcal{T}_t$ is increased at each iteration to permit longer transitions. On the other hand, the wrong segments of the sequence are penalized.

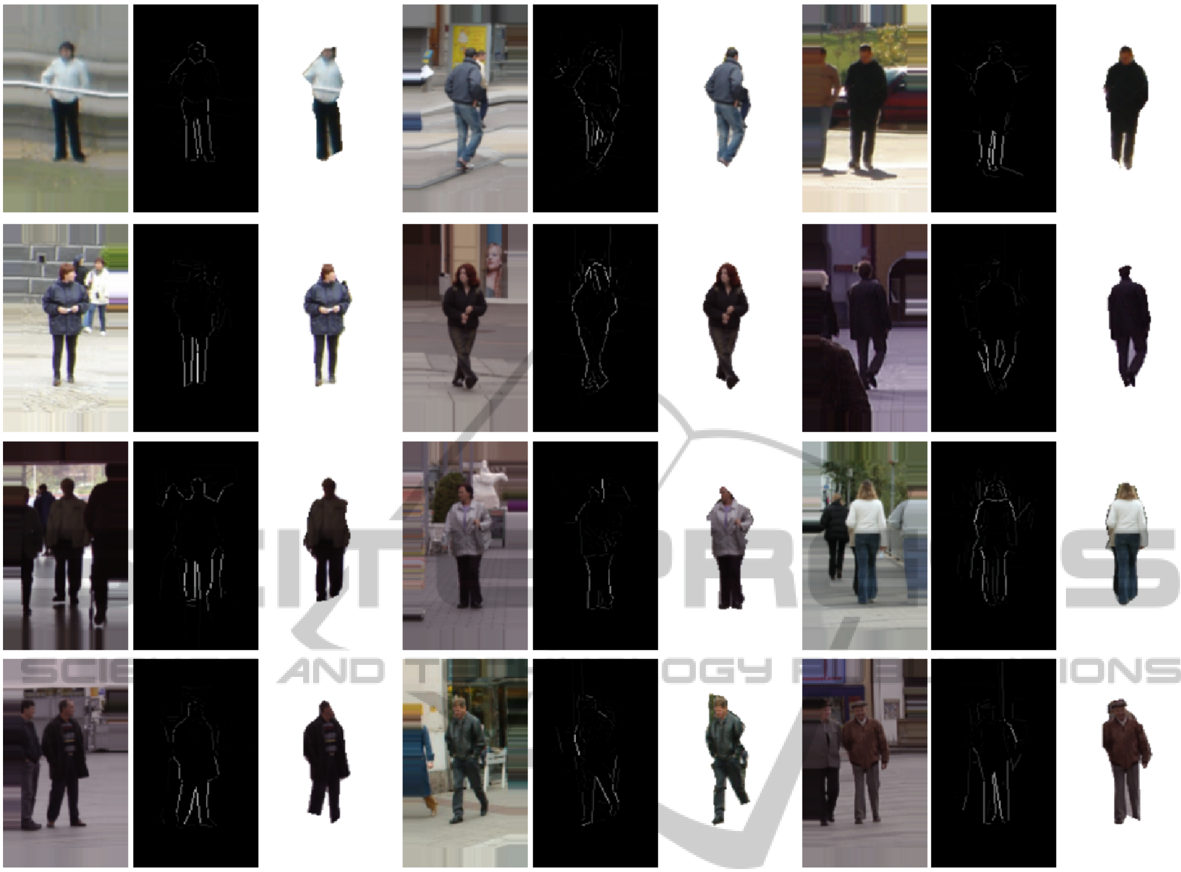
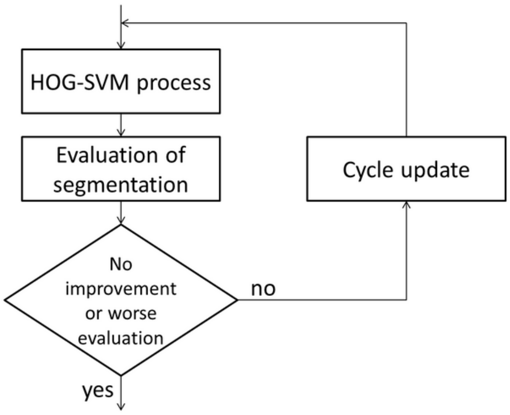


Figure 4: Segmentation of 12 pedestrians in cluttered scenes. From left to right: initial detection window, likelihood of contour segments as being a part of a human silhouette and segmentation obtained with the iterative process.



HOG-SVM process
Evaluation of segmentation
Cycle update
No improvement or worse evaluation
no
yes

Figure 5: An overview of the iterative process. A new iteration is achieved on wrong parts of the cycle as long as it improves the segmentation quality.

Let an edge of $\mathcal{G}$ that goes to a wrong segment *seg* ($L_{seg} \leqslant \mathcal{T}_e$). The weight ω of the edge is updated as follows:

$$\omega \leftarrow \frac{1+\mathcal{T}_e}{1+L_{seg}}\omega \tag{8}$$

The new path replaces the previous one in the current cycle.

Process Termination. At each iteration, an evaluation of the segments in the new current cycle is made to check if it is better than the previous one (Figure 7(d)). The new cycle is evaluated by calculating the mean likelihood L of the cycle weighted by the segment size S_{seg}:

$$L = \frac{\sum\limits_{k \in cycle} S_k L_k}{\sum\limits_{k \in cycle} S_k} \tag{9}$$

If the value of L is smaller or equal to the previous one, the previous cycle is kept and the process is stopped. Otherwise, an improvement is still possible and a new iteration is performed.

6 EXPERIMENTS

The learning required for the SVM model computing is achieved using 400 positive examples from a binary human silhouette image database created for this work and from 200 negative examples of the *INRIA Static Person Data Set*. The algorithm used for classification in the pre-segmentation step is SVM-light (Joachims, 1999). It requires $\mathcal{T}_e$=0. The Canny operator parameters are adapted to the dataset. Thus we choose the ones taken in (Dalal and Triggs, 2005).

The evaluation of the segmentation is based on three measures advised by (Philipp-Foliguet and Guigues, 2008). They involve a ground truth which constitutes a reference segmentation. We manually made the ground truths of all the testing windows.

- The $F_{measure}$ considers the compromise between the precision and the recall of the assignation of pixel to the foreground or the background.

$$F_{measure} = \frac{2.precision.recall}{precision + recall} \qquad (10)$$

- The Martin measure checks the true assignation of important regions.
- The Yasnoff measure computes the distance between ill-assigned pixels and the nearest pixel belonging to its true region. It is closely related to the human perception of the quality of the segmentation.

The $F_{measure}$ and the Martin measure give a value in $[0,1]$, whereas the Yasnoff measure gives a value in $[0,+\infty[$. The Martin and Yasnoff measures decrease with the segmentation performance and the $F_{measure}$ increases with the segmentation performance.

In the experiments, 400 images from the *INRIA Static Person Data Set* were tested and compared to the manually made ground-truths. The evaluation of the segmentation is estimated from the mean of the measures for all the tested images.

6.1 Single Iteration Evaluation

First, the experiments are only conducted on the method without iteration. To optimize the algorithm, the appropriate value of the threshold $\mathcal{T}_t$, related to the maximum distance between two consecutive segments, and the appropriate influence factor α of the transition in the graph weights (see equation 3) need to be fixed. Figure 6 shows the $F_{measure}$ for various values of these two parameters. $F_{measure}$ promotes the values $\mathcal{T}_t$=14 and α=4 that are used in the sequel.

Using a non optimized C++ implementation on a 3GHz Pentium D machine, the first iteration excluding the pre-segmentation stage is processed in a mean time of 23 ms.

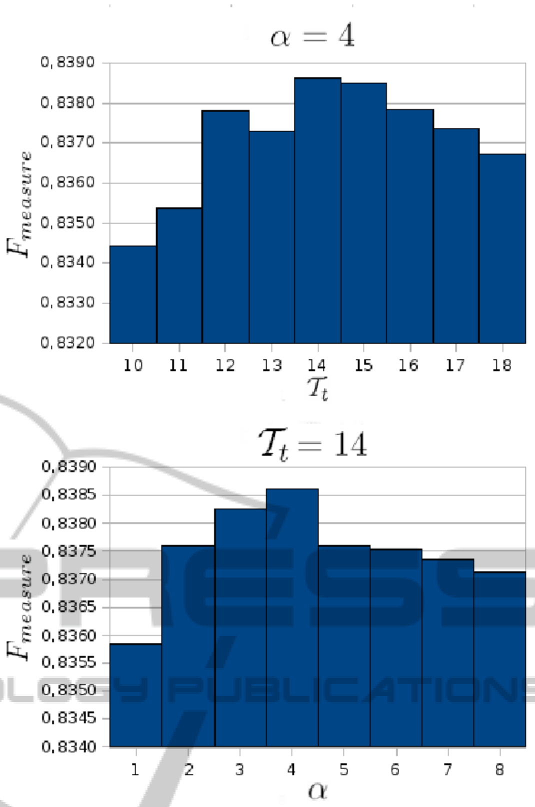


Figure 6: Segmentation evaluation by the $F_{measure}$ to evaluate the optimal value of the threshold $\mathcal{T}_t$ and the factor α. We chose from these evaluations a threshold of 14 and a factor α of 4.

6.2 Evaluation of the Iterative Process

On the 400 tests of the experiments, the mean number of required iterations is 2,14. That demonstrates that convergence is fast and computational cost is not too high.

Multiple iterations eliminate the illogical paths (see Figure 7). Actually, the three evaluation measures (see Table 1) confirm an important segmentation improvement with the iterative process. The iterations eliminate the false detections and particularly the far ill-assigned pixel. Some examples of segmentation obtained by this method can be shown in Figure 4.

Table 1: Evaluation measures with the Dijkstra's algorithm for a single iteration and with the iterative process. The three measures demonstrate that several iterations improve the segmentation. In order to facilitate the reading, the sign ↓ indicates a measure to minimize and the sign ↑ a measure to maximize.

Mesure	amp; First iteration	amp; Iterative process
$F_{measure}$ (↑)	amp; 0,8386	amp; **0,8405**
Martin (↓)	amp; 0,0495	amp; **0,0490**
Yasnoff (↓)	amp; 0,6482	amp; **0,6346**

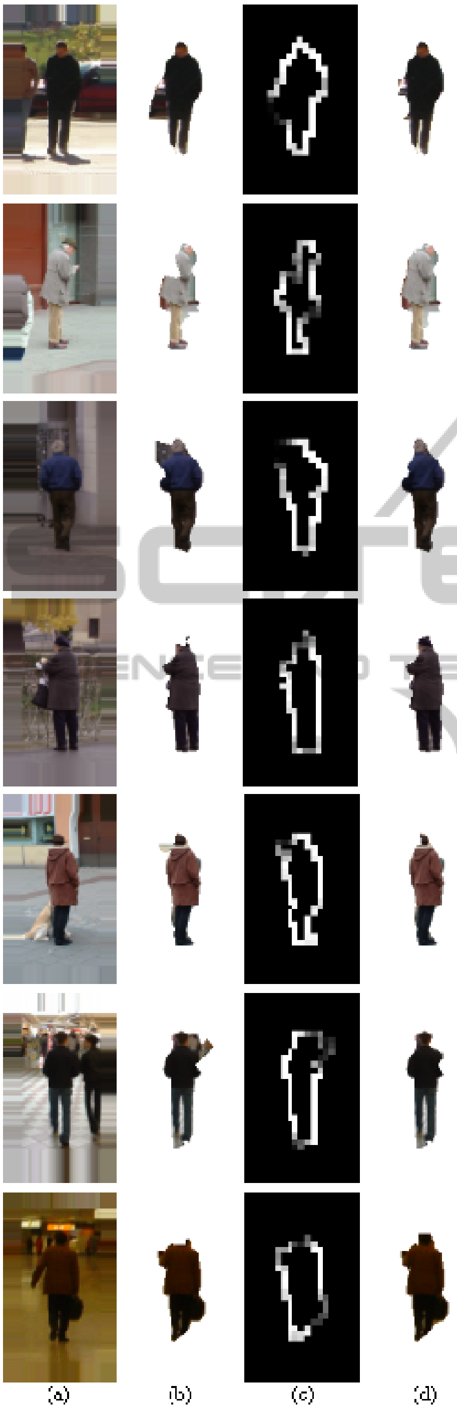


Figure 7: Segmentation improvement by the iterative process for 8 examples: initial detection window (a), segmentation obtained by a single iteration (b), evaluation of the segments of the cycle in the first iteration (c) and segmentation obtained at the end of the iterative process (d). Multiple iterations remove wrong parts of the silhouette when needed.

7 CONCLUSIONS

In this paper, we have proposed to directly adapt a new human segmentation process to an already existing efficient human detection method. Both processes are closely related and data previously computed in the detection are used in the segmentation. In this way, a pre-segmentation based on a HOG and SVM framework gives local information on the contour segments. Then, the segmentation is performed with the integration of the pre-segmentation cue in an oriented graph. Detection and segmentation can thus be achieved simultaneously. The quality of the segmentation is increased by an iterative process.

Future research directions will involve different issues. First of all, we have only studied static images. We could enrich descriptiveness by integrating a human motion cue. Additionally, the same process could be adapted to others classes than “human” (for example for cows, cars or dogs). Nevertheless, “human” is one of the most shape descriptive class. The same process should be less effective with other classes. Finally, some interactions with the user could improve performance and better deal with hard cases.

REFERENCES

Alonso, I., Llorca, D., Sotelo, M., Bergasa, L., Toro, P. D., Nuevo, J., Ocania, M., and Garrido, M. (2007). Combination of feature extraction methods for svm pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 30:292–307.

Bertozzi, M., Broggi, A., Rose, M. D., Felisa, M., Rakotomamonjy, A., and Suard, F. (2007). A pedestrian detector using histograms of oriented gradients and a support vector machine classifier. *IEEE Intelligent Transportation Systems Conference*, pages 143–148.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2:886–893.

Elder, J. and Zucker, S. (1996). Computing contour closure. *European Conference on Computer Vision*, 1:399–412.

Felzenszwalb, P., Girshik, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645.

Ferrari, V., Tuytelaars, T., and Gool, L. V. (2006). Object detection by contour segment networks. *European Conference on Computer Vision*, 3953:14–28.

Freund, Y. and Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *European Conference on Computational Learning Theory*, pages 23–37.

Gao, W., Ai, H., and Lao, S. (2009). Adaptive contour features in oriented granular space for human detection and segmentation. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1786–1793.

Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., and Malik, J. (2011). Semantic contours from inverse detectors. *IEEE International Conference in Computer Vision*, pages 991–998.

Hernandez, A., Reyes, M., Escalera, S., and Radeva, P. (2010). Spatio-temporal grabcut human segmentation for face and pose recovery. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 33–40.

Joachims, T. (1999). Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning*.

Kang, S., Byun, H., and Lee, S. (2002). Real-time pedestrian detection using support vector machines. *International Journal of Pattern Recognition and Artificial Intelligence*, pages 268–277.

Lin, Z. and Davis, L. (2010). Shape-based human detection and segmentation via hierarchical part-template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:604–618.

Mori, G., Ren, X., Efros, A., and Malik, J. (2007). Recovering human body configurations: Combining segmentation and recognition. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2:326–333.

Munder, S. and Gavrila, D. (2006). An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1863–1868.

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., and Poggio, T. (1997). Pedestrian detection using wavelet templates. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 193–199.

Philipp-Foliguet, S. and Guigues, L. (2008). Multi-scale criteria for the evaluation of image segmentation algorithms. *Journal of Multimedia*, pages 42–56.

Pishchulin, L., Jain, A., Andriluka, M., Thormaehlen, T., and Schiele, B. (2012). Articulated people detection and pose estimation: Reshaping the future. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Sharma, V. and Davis, J. (2007). Integrating appearance and motion cues for simultaneous detection and segmentation of pedestrians. *IEEE International Conference on Computer Vision*, pages 1–8.

Shotton, J., Blake, A., and Cipolla, R. (2008). Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1270–1281.

Vapnik, V. (1995). The nature of statistical learning theory. *Springer-Verlag*.

Wang, H. and Koller, D. (2011). Multi-level inference by relaxed dual decomposition for human pose segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2433–2440.

Wu, B. and Nevatia, R. (2007). Simultaneous object detection and segmentation by boosting local shape feature based classifier. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Zhu, Q., Yeh, M., Cheng, K., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:1491–1498.