# Extraction of Person Silhouettes from Surveillance Imagery using MRFs *

Vinay Sharma James W. Davis Ambrish Tyagi
Dept. of Computer Science and Engineering
Ohio State University
Columbus OH 43210 USA
{sharmav,jwdavis,tyagia}@cse.ohio-state.edu

## Abstract

*We present a method for the simultaneous detection and segmentation of objects from static images. We employ low-level contour features that enable us to learn the coarse object shape using a simple training phase requiring no manual segmentation. Based on the observation that most interesting objects (e.g., people) have regular and closed boundaries, we exploit relations between these features to extract mid-level cues, such as continuity and closure. For segmentation, we employ a Markov Random Field that combines these cues with information learned from training. The algorithm is evaluated for extracting person silhouettes from surveillance images, and quantitative results are presented.*

## 1. Introduction

Detecting instances of a particular object class from a single image is a challenging problem, and remains the attention of much research. An important application of such research is person detection from surveillance imagery. Several popular detection approaches [11, 3] use bounding boxes to indicate the location and extent of the person. While useful, such a result does not provide any information regarding the person shape. Though acquiring the target object shape is traditionally viewed as a segmentation task, most segmentation approaches extract boundaries (or silhouettes) given that the image already contains only a single instance of the object [1].

In this paper we present a method that attempts to simultaneously detect *and* segment people in static images. The approach is applicable to any object category, and is based on the simple observation that most natural and interesting object classes (e.g, people, vehicles, etc.) have shapes with regular and closed boundaries. Unlike related techniques [6, 10], our algorithm relies on a simple training phase and does not require any manual segmentation.

*Appears in *IEEE Workshop on Applications of Computer Vision*, Austin, TX, Feb 21, 2007

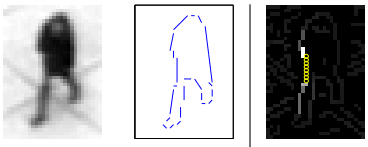
We introduce low-level contour features that enable us to learn the coarse object shape from weakly labeled training data. The relations between the features are exploited to extract mid-level cues such as contour continuity (local) and closure (global) that capture our expectation regarding object boundaries. We then employ a Markov Random Field (MRF), defined over the contour features, to obtain a contour segmentation by combining the local and global mid-level cues with likelihoods obtained during training. The approach is evaluated for the detection and segmentation of people from surveillance images.

## 2. Related Work

Several algorithms, such as [11, 3], generate satisfactory detection results, but provide no shape information of the object. In [5] a template based method was presented that matched an object with the most similar template from a database. The method required a large training dataset of fully segmented images. Recently, constellation models (e.g., implicit shape models [6]) have been used for the combined detection and segmentation of objects. However, this class of techniques also requires training sets with manually segmented foreground regions. In a related approach [8], boundary fragments (instead of regions) were used to learn the object geometry. Another approach using object boundaries was proposed in [10]. Both these methods employed a boosting technique during training and required some amount of segmentation during training (object centroids in [8], complete shape in [10]). Work such as [1] focus only on object segmentation, and discuss neither object detection, nor how the method works when the object is not present in the image.

## 3. Contour Features

The importance of first-order gradient information in estimating the shape and appearance of an object is well



(a) (b) (c)

**Figure 1. Contour features. (a) Input image. (b) Contours features extracted from object boundary. (c) Relative affinity between the marked contour (circles) and contours extracted from the entire image.**


(a) (b) (c)

**Figure 2. Contour closure. (a) Input image. (b) Examples of cycles. (c) $\mathcal{R}$ magnitudes computed for each contour.**

known [3, 7, 10]. We exploit this information by extracting contour-based features that capture the location, orientation, and magnitude of image gradients, and provide a simple means to describe object shape.

We extract short, nearly linear contour fragments based on changes in gradient direction. To ensure contour fragments of reasonable size, the edge orientations are initially quantized into a smaller number of bins. We represent a contour fragment, $c$, by a compact feature vector $f = [ep_1, ep_2, E_{mag}]$, where $ep_1$ and $ep_2$ are the coordinates of the two end-points and capture information regarding the orientation, extent, and location of the object gradients. The mean edge magnitude along the contour is represented by $E_{mag}$. We show an image region containing a person in Fig. 1(a), and in Fig. 1(b) we show the contour features extracted from along the person boundary. Each feature is denoted by a line segment with a thickness proportional to $E_{mag}$. The set of all contour features, $F = \{f_1, \ldots f_n\}$, extracted from the object gradients, forms a compact representation that implicitly captures object shape.

### 3.1. Contour Affinity

In order to capture the notion that objects have smooth, regular boundaries, we rely on the "affinity" between contours. Used in several computational figure completion methods [9], affinity measures how likely it is that two edge elements belong to the same underlying edge structure.

Given two contours $c_1$ and $c_2$, consider the simplest (lowest change in curvature) curve connecting an end-point of $c_1$ to an end-point of $c_2$. Based on [9], the affinity for this curve joining $c_1$ and $c_2$ is defined as

$$\mathcal{A} = e^{(-r/\sigma_r)} \cdot e^{(-\beta/\sigma_t)} \cdot e^{(-\Delta/\sigma_e)} \qquad (1)$$

where $r$ is distance between the end-points, and $\Delta = |E^{c_1}_{mag} - E^{c_2}_{mag}|$ (the absolute difference in contour intensity). The term $\beta = \theta_1^2 + \theta_2^2 - \theta_1 \cdot \theta_2$, where $\theta_1$ denotes the angle between the tangent vector at the end-point of $c_1$ and the line joining the end-points of $c_1$ and $c_2$. The angle $\theta_2$, formed at the end-point of $c_2$, is analogous to $\theta_1$. The normalization factors $\sigma_r$, $\sigma_t$, and $\sigma_e$ are written as $\sigma_r = R/w_1$, $\sigma_t = T/w_2$, and $\sigma_e = E/w_3$, where $R$, $T$, and $E$ equal the maximum possible value of $r$, $\beta$, and $\Delta$, and ($w_1$, $w_2$, $w_3$) are weights that can be used to change the relative influence of each term in the affinity calculation. Since $c_1$ and $c_2$ have two end-points each, there are four curves connecting the contours depending on which pair of end-points are connected. We define the contour affinity, *Aff*$(c_1, c_2)$, between contours $c_1$ and $c_2$ as the maximum affinity over the four possible curves.

We compute pairwise affinities between all the contour features extracted from an image. Features lying in close proximity along a common edge structure often align well, and have similar intensities and hence obtain high affinity values. In Fig. 1(c) we show the relative affinity values between the marked (in circles) contour and the rest of the contours extracted from the image. The figure clearly shows that in spite of the large number of contours in close proximity, the neighboring contours *along* the person boundary have the highest affinity.

### 3.2. Contour Closure

Apart from having smooth boundaries, a mid-level cue common to most object classes is that they have a finite extent bounded by a closed boundary. To capture this notion, using a method similar to [4], we determine if each contour belongs to a sequence of contours forming a closed loop.

We treat the contour features as nodes in a weighted, directed graph, where the weights on the arcs correspond to the affinity between the nodes. In order to create a sparse graph, we limit the out-degree of each node. Considering each node in turn, we compute the mean and standard-deviation of its affinity values with every other node. We then preserve only those arcs that have affinity values with a Mahalanobis distance greater than a threshold, $t$. We use a value of $t = 1$ for all the results reported here. The arcs of the graph are then assigned weights equal to the negative log of the affinity values (high affinity corresponds to low arc weight). This enables us to find the most likely cycle passing through a pair of contours using standard and efficient shortest-path algorithms (e.g., Dijkstra's algorithm).

If a cycle $C_{ij}$ exists between a pair of contours, $c_i$ and $c_j$, it is assigned a score, $S$, equal to the product of the area

of the cycle and the affinity of the arc with the maximum weight (minimum affinity) in the cycle. Thus, large cycles, formed by chains of high affinity contour features, are assigned better scores. Each contour is then assigned a value, $\mathcal{R}$, equal to the average score of the cycles passing through it and every other contour

$$\mathcal{R}(c_i) = \frac{\sum_{c_j \in F} S(C_{ij})}{n} \quad (2)$$

where $n$ is the number of contours in the image. In Fig. 2(b) we show examples of cycles, in descending order of $S$, extracted from the contour features of the image shown in Fig. 2(a). In Fig. 2(c) we show the contour features weighted by $\mathcal{R}$. Note that object contours generally have higher $\mathcal{R}$ values than background contours.

We propose to use this notion of closure as a *prior*, enforced using an MRF (Sect. 5). We choose not to include $\mathcal{R}$ in the feature descriptor $f$ due to two reasons. First, computing cycles passing through every pair of contours in an image is computationally intensive. Second, while object contours generally have high $\mathcal{R}$ values, we observe that it is possible for background contours also to have similarly high values (see small background contour in Fig. 2(c)).

## 4. Training Procedure

Using the features defined in Sect. 3, we employ a simple training procedure to learn a rough estimate of the shape of the target object class (e.g., person). The weakly labeled training data consists of separate sets of positive (with object), and negative (without object) cropped images. No manual annotation in the form of segmented object regions or marked object centroids is required.

We first extract features, $f$, as described in Sect. 3 from each image patch. These features populate a 5D space, where the dimensions represent the x and y coordinates of contour end-points ($ep_1$ and $ep_2$) and the edge magnitude ($E_{mag}$). In this 5D space, we create probability density functions (pdf) for the positive and negative features using normalized histograms. Other density estimation techniques could also be employed.

The modes of the positive pdf correspond to contour features common to most instances of the object class as seen in the training set. Given a new feature, the positive and negative pdfs are used to provide a likelihood measure of the feature belonging to the object or the background.

## 5. Contour labeling using MRF

Given that the positive and negative pdfs have been computed, we now describe how an input image is analyzed. We begin by extracting contour features $F = \{f_1, f_2, \ldots f_n\}$ from the input image, and aim to obtain a segmentation by assigning each feature a label from the set $\mathcal{L} = \{l_o, l_b\}$, corresponding to the "object" or "background" class.

Let $B$ denote a configuration of labels such that $\{f_1 = b_1, f_2 = b_2, \ldots f_n = b_n\}$, where $b_i \in \mathcal{L}$. We formulate the search for the optimal label configuration $B$ as a maximum a posteriori (MAP) problem. If we assume that the likelihood of a configuration of labels can be written as a product of the individual likelihoods, the MAP estimate is equivalent to minimizing the free energy [2]

$$E(B) = -\sum_i \log(p(f_i|b_i)) - \log(p(B)) \quad (3)$$

The first term corresponds to the likelihood of each contour feature belonging to the positive (object) or the negative (background) class. These likelihoods are learned during the training procedure, and capture the coarse shape of the specific object category. The second term corresponds to the prior probability of a shape, as defined by a given configuration of contour labels. As described in Sect. 3.1 and 3.2, the object classes of interest have regular, smooth shapes, and are bounded by a closed contour. In what follows we describe a MRF used to enforce these mid-level cues, while minimizing Eqn. 3.

We model the prior by employing a MRF defined over the set of contour features. In order to establish a neighborhood system for the MRF, we make use of contour affinity (see Eqn. 1). For each contour feature, $c_p$, we obtain the affinity value to all other features, and compute the mean and standard deviation of these values. Then, as described in Sect. 3.2, the features having affinity values greater than $t$ standard deviations from the mean are included in $\mathcal{N}_p$, the neighborhood of $c_p$.

Following the Hammersley-Clifford theorem, we define the probability of a configuration $P(B) \propto \exp(-\sum_k V_k(B))$, where $V_k$ denotes the clique potential defined over cliques $k$. We employ the generalized Potts model to define pairwise clique potentials as

$$V_{(p,q)}(b_p, b_q) = u_{(p,q)}(1 - \delta(b_p - b_q)) \quad (4)$$

where $p$ and $q$ are neighboring sites in the field, which in this case denote contour features. The quantity $u_{(p,q)}$ can be considered to be the cost of assigning different labels to $p$ and $q$. In most applications, the MRF is defined over a regular lattice (e.g., pixels) and the neighborhood of a site is formed by its 4- or 8-connected neighbors. In such cases, $u_{(p,q)}$ is often defined as a constant (well potential) giving a homogeneous MRF with isotropic clique potentials.

However, in contrast to most previous applications, the MRF described here is defined on contour features (not arranged in a regular grid). Further, the MRF is non-homogeneous, in that the clique potential across neighbor-

ing sites (contour features) depends on the properties of the sites. Instead of defining radially symmetric clique potentials, we wish to enforce a directional smoothness to the label configuration, such that if a contour feature has a positive (object) label, neighboring contours are assigned the same label only if they exhibit good continuity (high affinity) and closure (belong to a closed chain of contours).

We first identify a set $F'$ of *candidate* contours that are likely to belong to the object class using the thresholded log-likelihood ratio

$$f_i \in F', \text{ if } \ln\left(\frac{p(f_i|l_o)}{p(f_i|l_b)}\right) > T \tag{5}$$

Then, using the technique described in Sect. 3.2, we search for cycles connecting pairs of contours taken from $F'$. As we find cycles, $C_{ij}$, connecting contour $c_i$ with other contours $c_j$ in $F'$, we increment a pairwise interaction term, $\mathit{Cyc}(i,k)$, for all contours $c_k$ included in those cycles

$$\mathit{Cyc}(i,k) = \mathit{Cyc}(i,k) + \begin{cases} S(C_{ij}) & c_k \in C_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The value of $\mathit{Cyc}(i,k)$ is normalized by the number of contours in $F'$. Thus, a high value of $\mathit{Cyc}(i,k)$ suggests that, among cycles computed between contours in $F'$, many high scoring cycles passed through $c_i$ and $c_k$.

We combine both *Aff* and *Cyc* in order to define the penalty term $u_{(p,q)}$ in Eqn. 4 as

$$u_{(p,q)} = \begin{cases} \mathit{Aff}(p,q) \cdot e^{(-\sigma_c/\mathit{Cyc}(p,q))} & c_p \in F' \\ \mathit{Aff}(p,q) & \text{otherwise} \end{cases} \tag{7}$$

where $\sigma_c$ is a normalization constant. Thus, the cost of label discontinuity is greater for contour pairs with high affinity values. Furthermore, if a contour is in $F'$, this cost is greater for those pairs that have a high affinity and are likely to belong to a closed contour cycle.

Figure 3 shows an example illustrating the effect of *Aff* and *Cyc* on the clique potential. In Fig. 3(a) we show the original image and in Fig. 3 (b) we show, in white, the set of candidate contours $F'$. In Fig. 3(c) we show the relative strength of $u_{(p,q)}$ between the marked contour (in circles), and the contours in its neighborhood computed using only *Aff*. In Fig. 3(d) we see how combining *Cyc* with *Aff* changes the distribution of $u_{(p,q)}$. Notice that the potential along the person boundary has been magnified, and the other values have been reduced nearly to zero.

As shown in [2], minimizing the energy function $E(f)$ in Eqn. 3 is equivalent to solving the mincut problem on an appropriately constructed graph. Following [2], the graph is composed of two types of vertices, the c-vertices (contour features) and the l-vertices (labels, $l_o$ and $l_p$). Among the

(a) (b) (c) (d)

**Figure 3. Clique potential. (a) Input image. (b) Candidate contours. (c) Relative magnitude of $u_{(p,q)}$ using only *Aff*. (d) Relative magnitude of $u_{(p,q)}$ using *Aff* and *Cyc* (Eqn. 7).**

c-vertices, if $q$ is in the neighborhood of $p$, then $p$ and $q$ are connected by an arc with weight $w_{(p,q)} = 2u_{(p,q)}$. Each c-vertex also has an incoming directed arc from $l_o$ (source) and an outgoing directed arc to $l_b$ (sink) with a weight

$$w_p^l = (\ln(P(f_p|l)) + K) + \sum_{q \in \mathcal{N}_p} w_{(p,q)} \tag{8}$$

where $l \in \mathcal{L}$ and $K$ is a constant ensuring that the weights are positive. The min-cut of this graph ensures that each contour feature is connected to only one of the l-vertices, $l_o$ or $l_b$, and provides the required contour labeling.

## 6. Experiments

In this section we evaluate the performance of the algorithm for the extraction of person silhouettes from surveillance images. In all our experiments we used images produced by 6 different outdoor, roof-mounted, pan-tilt-zoom color cameras encompassing several different backgrounds, and view angles.

We first evaluated the ability of our algorithm to detect and segment person contours from cropped image regions. To train our system, we manually generated 2034 30x40 image patches (after left-right reflection) containing people in a wide variety of poses and orientation. For the negative training set, we randomly selected 5000 30x40 image patches from a collection of similar images not containing people. To test our algorithm we generated an additional 200 positive, and 1000 negative image patches from images not seen during training. In Fig. 5(a) we show examples from the positive test set. In Fig. 5(b) we show an intermediate result from the algorithm. The extracted contours are shown in dark gray, and the initial candidate contour fragments (see Eqn. 5) are shown in light gray. The final contour segmentation is shown in Fig. 5(c). These examples illustrate the ability of the algorithm to extract person contours in various scenarios, including different poses, background structure, and proximity to other people. Comparing Fig. 5(b) and (c) we also see the clear improvement achieved by enforcing shape priors (Sect. 5). The last row of Fig. 5 shows an instance of a negative example.

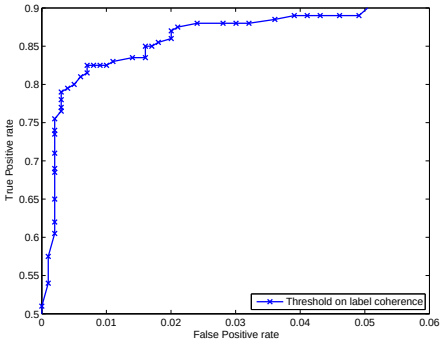



**Figure 4. ROC curve obtained by varying the threshold on label coherence.**

In order to quantitatively evaluate the algorithm, we manually segmented the images in the test set by labeling the contours in each image. For each image in the positive set, we computed Precision and Recall values in terms of the number of correctly labeled pixels. Over the 200 positive images tested, the segmentation produced by the algorithm had an average F-measure (harmonic mean of Precision and Recall) of 0.87. For the negative set, we computed for each image the ratio of the number of false positives (pixels labeled as belonging to an object contour) to the total number of contour pixels. For the 1000 images in the negative set, the algorithm generated an average false positive ratio of 0.014. These results suggest that while the algorithm is adept at segmenting the object when it is present in the image, it also generates very few (if at all) false positive labels when no object is present.

A simple measure of the coherence of the generated contours labels can be very effective at classifying an image patch as containing an object or not. In this case, we compute the average of the best cycle score and the median positive likelihood of the object contours as the measure of coherence. In Fig. 4 we show a ROC curve generated by varying the threshold on this coherence measure in order to separate the positive and negative test images. As can be seen, an appropriately chosen threshold can result in a very small percentage of mis-classifications.

Apart from measuring the accuracy of the contour segmentation, we also evaluate the quality of the silhouettes generated from these contours. In order to form silhouettes, we search for cycles formed by the positively labeled contours, and score them as described in Sect. 3.2. This process does not impose large computational overhead since the pairwise contour affinities are already computed, and the graph is sparse, consisting only of the positively labeled contours. The end-points of the contours belonging to the highest scoring cycle are then simply joined using straight lines giving a complete, closed outline. This outline is flood-filled to generate the final silhouette.

For the same positive test set used earlier, we generate silhouettes from the positively labeled contours and compared these with manually formed silhouettes (created by connecting together the hand-labeled contours). The average F-measure of the silhouettes generated by the algorithm for the 200 positive images was 0.93. This shows that we were able to generate silhouettes that were reasonably close to the actual shape of the objects. In Fig. 5(d) we show examples of silhouettes created by the algorithm, and in Fig. 5(e) the manually created silhouettes.

Next, we evaluated the performance of our algorithm using different combinations of features. We modified the MRF (see Sect. 5) to only enforce the affinity-based smoothness prior, without utilizing contour closure (Eqn. 7). The features extracted to learn the likelihoods were then varied to yield 4 different combinations as follows, $f^1 = [ep_1, ep_2]$ (position only), $f^2 = [ep_1, ep_2, \mathcal{R}]$ (position and closure), $f^3 = [ep_1, ep_2, E_{mag}, \mathcal{R}]$ (position, edge intensity, and closure), and $f^4 = [ep_1, ep_2, E_{mag}]$ (position and edge intensity). The performance of the algorithm using these different features was compared based on the 200 positive and 1000 negative test images used earlier.

In Table 1 we show the average F-measure obtained by each feature set ($f^1$-$f^4$) for the contour and silhouette segmentation of the positive test images. For comparison, the table also shows the results obtained using the proposed formulation ($f^*$), where feature $f^4$ is used during training, and both contour affinity *and* closure are enforced as priors. The corresponding F-measures computed using the candidate contours (prior to using the MRF) are shown in parenthesis. Comparing results of $f^3$ and $f^4$ against $f^1$ and $f^2$, we find that edge intensity combined with position provides pertinent information during training. As predicted (Sect. 4), comparing $f^3$ and $f^4$ we see that including $\mathcal{R}$ (computed over *all* contours) in the feature vector does not provide any improvement in performance. However, comparing these results with $f^*$, we see that including closure as a prior, along with contour affinity, does provide a distinct performance boost.

Table 2 shows the false positive rates for the 1000 negative images using the different feature combinations. Comparing the results we once again see that the proposed method, $f^*$, provides the best results. Further, in each of the settings, the final contour labels are found to be more accurate than the results generated before application of the prior (shown in brackets).

## 7. Summary and Future work

We presented a method to detect and segment people from static images. The algorithm utilizes low-level contour features, and relies on a simple training phase to first obtain a rough estimate of the target object shape. Based on the observation that objects (people) have regular, closed bound-

| | $f^1$ | $f^2$ | $f^3$ | $f^4$ | $f^*$ |
|---|---|---|---|---|---|
| Contour | 0 | 0.15 | 0.79 | 0.80 | 0.87 |
| | (0.34) | (0.39) | (0.64) | (0.70) | (0.70) |
| Silhouette | 0.00 | 0.07 | 0.72 | 0.84 | 0.94 |

**Table 1. F-measures for positive examples.**

| | $f^1$ | $f^2$ | $f^3$ | $f^4$ | $f^*$ |
|---|---|---|---|---|---|
| Contour | 0.008 | 0.010 | 0.029 | 0.020 | 0.014 |
| | (0.015) | (0.039) | (0.091) | (0.045) | (0.045) |

**Table 2. False positive rates for negative examples.**

aries, mid-level cues such as contour continuity and closure are computed. An MRF, defined over the contour features, is used to integrate the different sources of information and provide a contour segmentation. While the approach is applicable to general object categories, we presented results for detecting and segmenting people from surveillance imagery. The results were evaluated against manually marked data, and generated a high F-measure of Precision and Recall. We are currently working on an implementation that will enable us to scan images at multiple resolutions. We also plan to evaluate the method for other objects categories.

## References


[1] E. Borenstein and J. Malik. Shape guided object segmentation. In *Proc. Comp. Vis. and Pattern Rec.*, volume 1, pages 969–976, 2006.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 23(11):1222–1239, 2001.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. Int. Conf. Comp. Vis.*, pages 886–893, 2005.

[4] J. Elder and S. Zucker. Computing contour closure. In *Proc. Eur. Conf. Comp. Vis.*, pages 399–412, 1996.

[5] D. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. Eur. Conf. Comp. Vis.*, pages 37–49, 2000.

[6] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc. Eur. Conf. Comp. Vis. Workshop*, 2004.

[7] K. Mikolajczyk, A. Zisserman., and C. Schmid. Shape recognition with edge-based features. In *Brit. Mach. Vis. Conf.*, pages 779–788, 2003.

[8] A. Opelt et al. A boundary-fragment-model for object detection. In *Proc. Eur. Conf. Comp. Vis.*, 2006.

[9] E. Sharon, A. Brandt, and R. Basri. Completion energies and scale. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 22(10):1117–1131, 2000.

[10] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *Proc. Int. Conf. Comp. Vis.*, 2005.

[11] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. Int. Conf. Comp. Vis.*, pages 734–741, 2003.

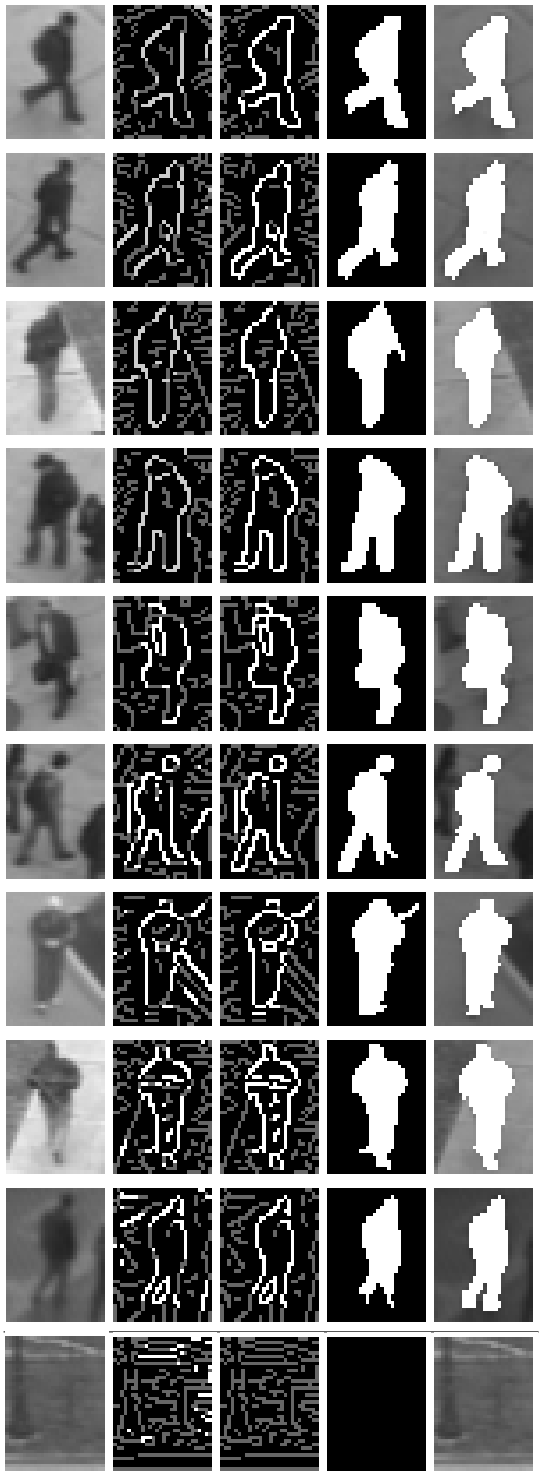



**Figure 5. Example results. (a) Input image. (b) Candidate contours. (c) Selected contours. (d) Silhouettes. (e) Ground-truth.**