

Human Detection using Iterative Feature Selection and Logistic Principal Component Analysis

Wael Abd-Almageed Larry Davis
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742
wamagaed, lsd@umiacs.umd.edu

Abstract—We present a fast feature selection algorithm suitable for object detection applications where the image being tested must be scanned repeatedly to detect the object of interest at different locations and scales. The algorithm iteratively estimates the belongness probability of image pixels to foreground of the image. To prove the validity of the algorithm, we apply it to a human detection problem. The edge map is filtered using a feature selection algorithm. The filtered edge map is then projected onto an eigen space of human shapes to determine if the image contains a human. Since the edge maps are binary in nature, Logistic Principal Component Analysis is used to obtain the eigen human shape space. Experimental results illustrate the accuracy of the human detector.

I. INTRODUCTION

Human detection has been an active research area for almost a decade. In visual surveillance systems [2] and automatic navigation systems [1][7], detecting human objects is a prerequisite to applying higher level steps such as activity recognition or path planning.

We present an algorithm for detecting humans in upright postures from still images. Edge maps are first computed for the test image. The edge map is then iteratively filtered using a novel, fast feature selection algorithm. The feature selection algorithm probabilistically assigns features to either foreground or background clusters. Probabilities are updated iteratively. The distance between the filtered edges and a human-silhouette eigen shape-space is used to determine the presence of humans. The eigen shapes are obtained using Logistic Principal Component Analysis (LPCA) [11].

This paper is organized as follows. In Section II we briefly summarize related human detection work. Section III presents an iterative feature selection algorithm using kernel density estimation. Building the human shape space using Logistic Principal Component Analysis is presented in Section IV. Experimental results are presented in Section V. Finally, we conclude the paper in Section VI.

II. HUMAN DETECTION FROM UNMANNED GROUND VEHICLES

Human detection algorithms can be broadly classified into two main categories – (1) shape-based human detectors such

This work was funded, in part, by Army Research Laboratorys Robotics Collaborative Technology Alliance program; contract number DAAD 19-012-0012 ARL-CTA-DJH.

as such as the work of Gavrilu and Philomin [8], Gavrilu and Munder [7], Dalal and Triggs[5] and Zhu et al. [13], and (2) the motion-based human detectors, such as the work of Cutler and Davis [4], Ran et al. [10] and Abd-Almageed et al. [1].

The shape-based methods aim to detect humans in still images by extracting gradient-based (or edge-based) features and either matching these features against a number of human templates (e.g. [8]) or using a binary classifier to decide if a human object exists (e.g. [5][13]). The motion-based human detection methods depend on tracking an object of interest for a short period of time (typically 2–4 seconds) and analyzing the motion pattern of the object. A tracked object is classified as a human if it exhibits a twin-pendulum-like periodic motion (e.g. [1]).

The main limitation of shape-based detectors is the relatively high false alarm rate since the entire image has to be scanned (repeatedly at an arbitrary number of scales) to find all human figures. On the other hand, the main advantage of the motion-based methods is the low false detection rate because they depends on tracking the object for a period of time, which decreases the uncertainty about the object's nature. However, an object of interest must first be detected in order to employ the motion-based methods. Also, the object must be tracked successfully.

Hussein et al. in [9] have shown that a successful strategy to reduce the overall false detection rate is to integrate both shape-based and motion-based methods. Shape-based methods are first used to detect potential human objects. An object tracker is then used to track the object for a sufficient period of time. Finally, the motion of the tracked object is analyzed to verify if it resembles the motion of a human.

Since the algorithms presented in this paper mainly improve the shape-based methods, we will briefly highlight the previous work in this area.

Gavrilu and Philomin in [8] introduced a fast human detection algorithm based on the distance transform (chamfer distance) [3]. They collected a database of silhouette images of humans in different poses. During the training phase, K-means is used to cluster the silhouette database based on the pair-wise distance, into three clusters. K-means is repeatedly applied to each cluster to further cluster it into three sub-clusters. The process is repeated yielding a hierarchy of

human silhouettes. To detect if a human exists in a given image (of the same size as the training images), edges are extracted from the test image and the distance is computed between the edge map and silhouettes in the hierarchy. A human is detected if the computed distance is smaller than a pre-specified threshold across all levels of the hierarchy. The main advantage of this algorithm is speed. However, the algorithm is highly sensitive to image clutter and noise. Recently, Gavrilu and Munder [7] integrated the detection algorithm with stereo vision to lower the false detections.

Dalal and Triggs in [5] introduced a learning-based algorithm to detect humans from a single image. The image is divided into 16×16 rectangular neighborhoods and a feature vector called the Histogram of Oriented Gradients (HoG) is computed for each neighborhood. The HoG represents the probability distribution of gradient orientation (quantized into a pre-defined number of histogram bins) over a specific neighborhood. All HoGs from all image neighborhoods are concatenated to form a larger feature vector describing the image. A Support Vector Machine (SVM) is used to determine if the given image contains a human. In general, this method has a lower false alarm rate than Gavrilu and Philomin [8]. However, to check for humans at different scales, computing the HoG feature vector and using the SVM to classify it becomes very computationally expensive and therefore the detector is relatively very slow. It was reported in [5] that the algorithm runs at 1 frame per second if they scan 800 detection windows in an 320×240 image.

In [13], Zhu et al. used a cascaded Adaboost algorithm [6][12] to rapidly detect humans in static images using the HoG feature vector. However, in order to improve the overall performance, they used integral images [12] to compute the feature vector. A Support Vector Machine classifier was used as the weak classifier of the Adaboost algorithm.

The common limitation of all of these methods, when used to detect humans from a UGV or a smart vehicle, is the high false detection rate.

III. ITERATIVE FEATURE SELECTION

As we highlighted in Section II, the only low-level feature that could be used to detect humans from a single frame is the gradient (or edge map) of the gray levels of the image. Consequently, the background clutter highly influences the false detection rate. In this Section we present a method for differentiating background features from foreground features.

Let $\mathbf{e} = (x, y)$ be a feature point at location (x, y) . The probabilities that \mathbf{e} belongs to the foreground object or the background clutter are $P(\mathbf{F}|\mathbf{e})$ and $P(\mathbf{B}|\mathbf{e})$, respectively, where $P(\mathbf{F}|\mathbf{e}) + P(\mathbf{B}|\mathbf{e}) = 1$. Both $P(\mathbf{F}|\mathbf{e})$ and $P(\mathbf{B}|\mathbf{e})$ represent the spatial distribution of feature points in the foreground and the background, respectively. The probability that a feature point \mathbf{e}_j belongs to the foreground, $P^i(\mathbf{F}|\mathbf{e}_j)$ can be estimated using Equation (1)

$$P^i(\mathbf{F}|\mathbf{e}_j) = \frac{1}{N |\mathbf{H}|} P^{i-1}(\mathbf{F}|\mathbf{e}_j) \sum_{n=1}^N P^{i-1}(\mathbf{F}|\mathbf{e}_n) \mathcal{K}(\mathbf{H}^{-1}(\mathbf{x}_{\mathbf{e}_n} - \mathbf{x}_{\mathbf{e}_j})) \quad (1)$$

where $\mathbf{x}_{\mathbf{e}_j}$ is a D -dimensional feature vector representing the feature point \mathbf{e}_j , $\mathcal{K}(\cdot)$ is a multivariate kernel function, H is the kernel bandwidth and N is the number of feature points in the image. We assume that the attributes of the feature vector \mathbf{x} are statistically uncorrelated. Therefore, H is a diagonal matrix. The super script i represents the value of $P^i(\cdot)$ at iteration i .

Many methods exist for evaluating the kernel function. Here we use the classic product kernel as shown in Equation (2)

$$\mathcal{K}(\mathbf{H}^{-1}(\mathbf{x}_{\mathbf{e}_n} - \mathbf{x}_{\mathbf{e}_j})) = \prod_{d=1}^D h_d^{-1} \exp\left(-0.5 \frac{(\mathbf{x}_{\mathbf{e}_n}(d) - \mathbf{x}_{\mathbf{e}_j}(d))^2}{h_d}\right) \quad (2)$$

where $x(d)$ is the d^{th} attribute of \mathbf{x} and h_d is the d^{th} element of the diagonal of H . Similarly, the probability that a feature point \mathbf{e}_j belongs to the background, $P^i(\mathbf{B}|\mathbf{e}_j)$, can be estimated using Equation (3).

$$P^i(\mathbf{B}|\mathbf{e}_j) = \frac{1}{N |\mathbf{H}|} P^{i-1}(\mathbf{B}|\mathbf{e}_j) \sum_{n=1}^N P^{i-1}(\mathbf{B}|\mathbf{e}_n) \mathcal{K}(\mathbf{H}^{-1}(\mathbf{x}_{\mathbf{e}_n} - \mathbf{x}_{\mathbf{e}_j})) \quad (3)$$

Equations (1) and (3) are kernel density estimates of $P^i(\mathbf{F}|\mathbf{e}_j)$ and $P^i(\mathbf{B}|\mathbf{e}_j)$, respectively, regularized by the probability of finding the feature point \mathbf{e}_j at location (x, y) .

To satisfy the stochastic condition that $P(\mathbf{F}|\mathbf{e}) + P(\mathbf{B}|\mathbf{e}) = 1$, the probabilities are normalized as shown in Equation (4).

$$P^i(\mathbf{F}|\mathbf{e}_j) = \frac{P^i(\mathbf{F}|\mathbf{e}_j)}{P^i(\mathbf{F}|\mathbf{e}_j) + P^i(\mathbf{B}|\mathbf{e}_j)} \quad (4)$$

$$P^i(\mathbf{B}|\mathbf{e}_j) = \frac{P^i(\mathbf{B}|\mathbf{e}_j)}{P^i(\mathbf{F}|\mathbf{e}_j) + P^i(\mathbf{B}|\mathbf{e}_j)}$$

The kernel values $\mathcal{K}(\mathbf{H}^{-1}(\mathbf{x}_{\mathbf{e}_n} - \mathbf{x}_{\mathbf{e}_j}))$ are not dependent on the iteration number i . Therefore, we can pre-compute an $N \times N$ symmetric matrix

$$\mathbf{K}(n, j) = \mathcal{K}(\mathbf{H}^{-1}(\mathbf{x}_{\mathbf{e}_n} - \mathbf{x}_{\mathbf{e}_j})) \quad \text{and} \quad n, j = 1, \dots, N \quad (5)$$

in $\mathcal{O}(\frac{N^2-N}{2})$. By arranging the probabilities $P^i(\cdot)$ in $N \times 1$ vectors, Equations (1) and (3) can now be rewritten in vector-matrix product form as

$$\mathbf{P}^i(\mathbf{F}) = \frac{1}{N |\mathbf{H}|} \mathbf{P}^{i-1}(\mathbf{F}) .* (\mathbf{P}^{i-1}(\mathbf{F}) \mathbf{K}) \quad (6)$$

and

$$\mathbf{P}^i(\mathbf{B}) = \frac{1}{N |\mathbf{H}|} \mathbf{P}^{i-1}(\mathbf{B}) .* (\mathbf{P}^{i-1}(\mathbf{B}) \mathbf{K}) \quad (7)$$

where $.*$ indicates element-by-element multiplication.

$P^0(\mathbf{F}|\mathbf{e}_j)$ represent our a priori information about finding a feature point at a certain location. In order to estimate the initial probabilities $P^0(\mathbf{F}|\mathbf{e}_j)$ and $P^0(\mathbf{B}|\mathbf{e}_j)$, we use a training database. A large number of human images is collected from real-world images. The images are then segmented and cleaned as shown in Figure 1.

Given the shape database, we can now estimate the probability of finding a feature point at a specific image location by simply averaging the shape database. Figure 2 shows the

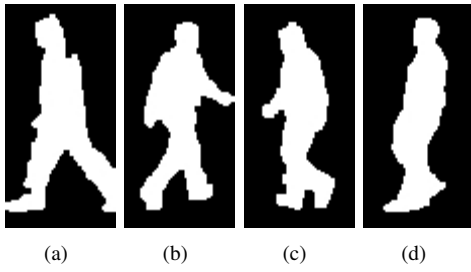


Fig. 1. Examples of the human silhouette database used for training

spatial probability density estimated from the shape database and used to compute $P^0(F|e)$ and $P^0(B|e)$.

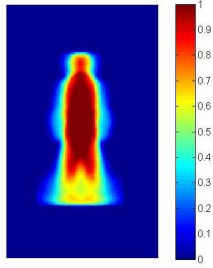


Fig. 2. Initial probability distribution of human silhouette features obtained from a shape database

The vector x_{e_n} encodes the characteristics of the image at the feature point e_n . In our algorithm, we use a 5d feature vector illustrated in Equation (8)

$$x_{e_n} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T \quad (8)$$

where

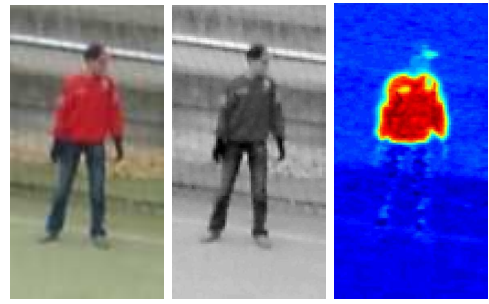
- x_1 is the luminance value at e_n ,
- x_2 is the a-component of CIE L*a*b* color space at e_n ,
- x_3 is the b-component of CIE L*a*b* color space at e_n ,
- x_4 is the entropy of the image in a 7×7 neighborhood around e_n , and
- x_5 is the standard deviation of luminance in a 5×5 neighborhood around e_n .

Attributes x_1 , x_2 and x_3 encode the color information at e_n while x_4 and x_5 encode the texture around e_n . Figure 3 illustrates an example of the used features for a real human image.

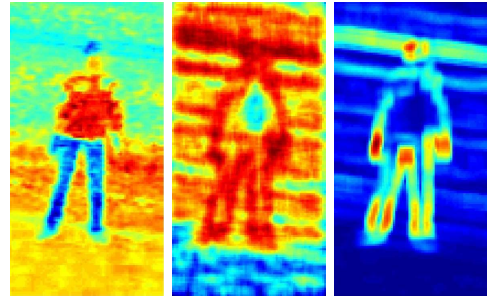
By initializing the spatial probabilities and iteratively applying Equations (6), (7) and (4), the set of image features are assigned to the foreground if $P^{\text{final}}(F|e) > P^{\text{final}}(B|e)$. Figure 4 demonstrates the evolution of the feature probabilities when applying the filtering algorithm to the image in Figure 3.a.

IV. LOGISTIC PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) has been proven a very effective tool in many areas such as dimensionality reduction, face detection and recognition, etc. PCA finds linear projections (of the data) on which the projected data



(a) Color image (b) L-component (c) a-component



(d) b-component (e) Entropy (f) Std. deviation

Fig. 3. Original image and feature images used in the feature selection algorithms. Figures 3.b, 3.c and 3.d represent the image in the CIE L*a*b* color space. Figures 3.e and 3.f represent the entropy and standard deviation features of the image using a 5×5 neighborhood

will have maximum variance. Therefore, the regular real-valued PCA (i.e. linear PCA) is not suitable for data of binary nature.

Using linear PCA with binary data have been addressed in a number of papers. Schein et al. in [11] overcame this shortcoming by relating binary-data PCA to linear PCA the same way logistic regressions is related to linear regression. The same approach was employed by Zivkovic and Verbeek in [14]. However, they extended the binary-data PCA to mixtures of binary PCA. We briefly overview the Logistic PCA (LPCA) of [11].

A human silhouette image I (e.g. Figure 1) is transformed into a M -dimensional column feature vector F by concatenating the rows of the image, where $M = \text{width} \times \text{height}$. The features of the vector are binary random variables such that $f_i \in \{0, 1\}$. The feature vector is a sparse vector containing mostly 0's except for the pixels where edges exist. Each feature can be modeled using a Bernoulli distribution as shown in Equation (9)

$$P(f_i|p_i) = p_i^{f_i} (1 - p_i)^{1-f_i} \quad (9)$$

where p is the distribution mean. The distribution can be rewritten as

$$P(f_i|p_i) = \sigma(\theta_i)^{f_i} \sigma(-\theta_i)^{1-f_i} \quad (10)$$

where $\theta_i = \log\left(\frac{p_i}{1-p_i}\right)$ is the log-odds parameter and $\sigma(\theta_i) = (1 + e^{-\theta_i})^{-1}$ is the logistic function (hence the name Logistic PCA).

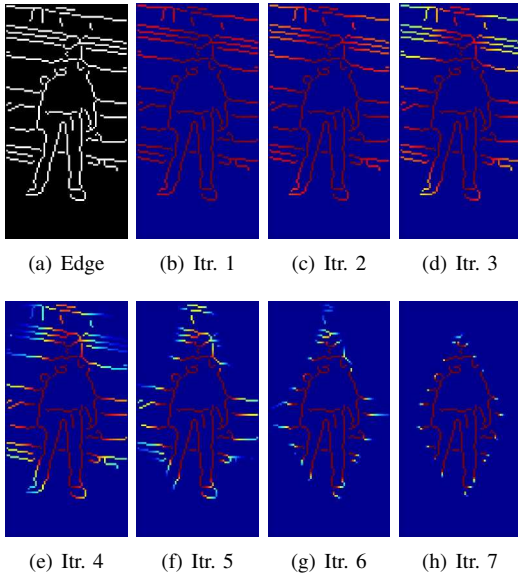


Fig. 4. An example of applying the feature selection algorithm on the edge map of 3.a. The colors of 4.b-4.h represent the evolution of the feature probability. Warmer colors represent higher probabilities

Given K training images, an observation matrix X is constructed as

$$X = [F_1 F_2 \dots F_K]^T \quad (11)$$

yielding a $K \times M$ matrix. The probability distribution of Equation (10) can now be generalized over the multivariate observation matrix X by Equation (12).

$$P(X|\Theta) = \prod_{km} \sigma(\Theta_{km})^{X_{km}} \sigma(-\Theta)^{1-X_{km}}. \quad (12)$$

The log-likelihood of the data under Equation (12) is

$$\mathcal{L} = \sum_{km} \{X_{km} \log(\sigma(\Theta_{km})) + (1 - X_{km}) \log(\sigma(-\Theta_{km}))\}. \quad (13)$$

The dimensionality of the log-odds matrix Θ can be reduced by assuming a compact representation for Θ and maximizing the log-likelihood of Equation (13) given the compact representation. Let

$$\Theta_{km} = \sum_l U_{kl} V_{lm} \quad (14)$$

where V and U are the projection matrix and projection coefficients, respectively. The projection matrix V is $L \times M$ (where L is user-defined and $L \ll M$.) Similarly, U is $K \times L$. Substituting Equation (14) into Equation (13) and maximizing \mathcal{L} with respect to U and V yields a compact eigen representation for Θ . Schrein et al. in [11] used least squares to find optimal values for U and V where they fixed one parameter and maximized the likelihood with respect to the other parameter and then alternate between the two parameters. For more details about the optimization method, the reader is referred to [11].

The rows of the projection matrix V represent the eigen vectors of the shape space. On the other hand, the rows of the coefficients matrix U represent the projections of the

K training examples onto the L eigen shapes. In order to determine if a given window of an image contains a human, the edge filtering algorithm of Section III is first applied to the windows of interest. The resulting edge map E (in vector form) is then projected into the shape space specified by U and V . The projection coefficients vector Y is computed using Equation (15)

$$Y = V(E - \bar{E}) \quad (15)$$

where \bar{E} is the average edge image obtained from the training shape database. The vector Y is an L dimensional vector representing the location of the filtered edge map E in the shape space V . The distance between E and the shape space is then

$$\epsilon^2 = \min_i \|Y - U_i\| \quad \text{and} \quad i = 1, \dots, K \quad (16)$$

where U_i is the i^{th} row of U . In other words, the distance between E and the shape space is the minimum distance between the projection of E and projections of the training shape data. A human exists in the test image if $\epsilon^2 < \delta$ and δ is an imperially determined threshold.

V. EXPERIMENTAL RESULTS

Figure 5.a shows a human walking in front of a cluttered background. The background produces too many unwanted edges in the edge map as shown in Figure 5.b. By applying the feature selection algorithm of Section III, most of the unwanted edges have been filtered out, leaving only the edges that belong to the foreground object (a human in this case.) Figures 5.c-5.g demonstrate the evolution of the edge probabilities and Figure 5.h shows the resulting edge map where $P^{\text{final}}(F|e) > P^{\text{final}}(B|e)$.

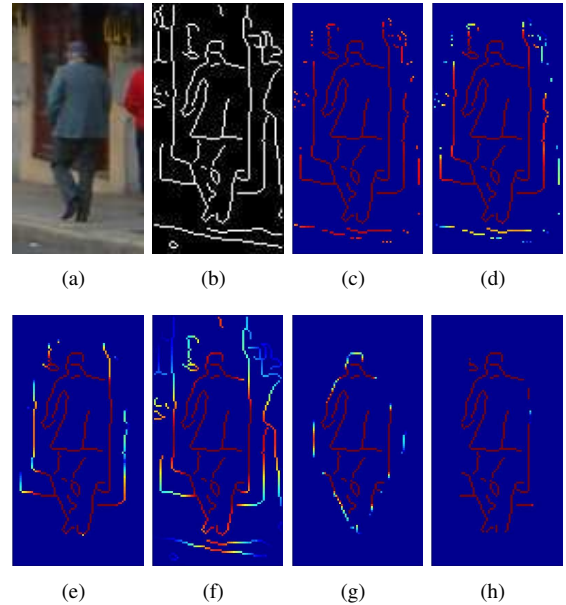


Fig. 5. A human image, the evolution of the edge probabilities and the final edge assignment

If the image being filtered does not have a foreground object or contains a non-human object, the remaining edges

after applying the filtering algorithm will generally not take the shape of a human because of the color and texture features used. Figure 6.a is a typical cluttered image without a foreground human. The final result of applying the filtering algorithm, shown in Figure 6.h, does not resemble a human.

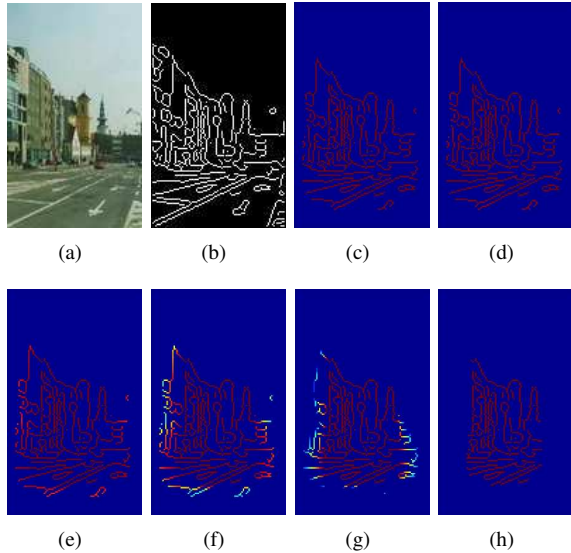


Fig. 6. A cluttered non-human image, the evolution of the edge probabilities and the final edge assignment

The edge-filtered maps are projected onto the human shape space obtained using LPCA constructed from a training data database of 1000 human silhouettes, as shown in Figure 1. We use $L = 10$ eigen shapes illustrated in Figure 7. Figure 8 shows the distance plots between human or non-human objects from one side and the shape space from another. The blue curve represents the distance between the projection of the filtered edges of Figure 5.h and the projections of the training database onto the shape space. The same distance of 6.h is represented by the red curve. It is clear that a large classification margin between the two curves exists. A simple thresholding is then sufficient to determine the existence of a human in the image of interest.

We used positive and negative test images from the INRIA human detection database which was used in [5]. The images are of size 70×134 . Humans are approximately of size 96×32 in the center of the image. Tables I and II demonstrate the results of applying the detection algorithm to a number of negative and positive test images, respectively. The distance ε^2 of the negative examples is at least one order of magnitude larger than that of the positive examples. By empirically setting the threshold $\delta = 800000$, negative and positive examples are correctly classified.

Applying the detector to images of different scale is straightforward. Two pre-processing steps are required. The spatial density function must be computed from the training database and the eigen shapes must be reestimated. Normally, with any shape-based detector, different detectors of an arbitrary number of scales are trained offline before being applied at run time.

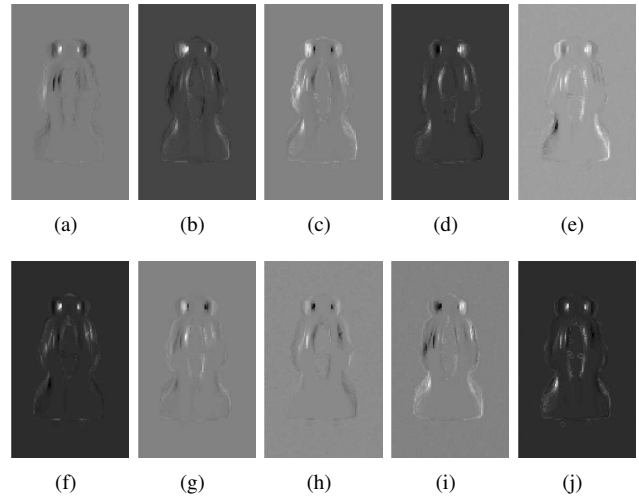


Fig. 7. An eigen human silhouette shape composed of 10 eigen shapes. The compact shape space is computed by applying LPCA in a database of human silhouettes such that of Figure 1

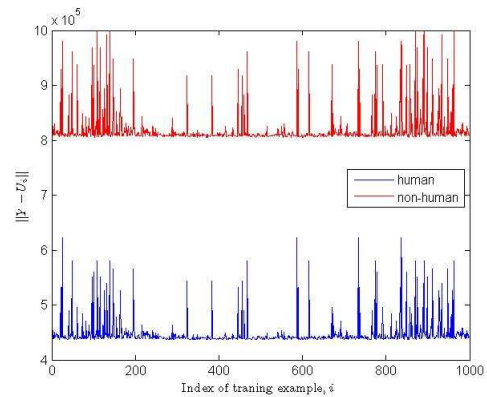


Fig. 8. Comparison between the distance of test projection vector and the projection vector of the training database. The distance of a human image (blue curve) is always smaller than that of a non-human image (red curve)

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a new human detection algorithm. The edge map of image of interest is filtered using a novel probabilistic feature assignment algorithm based on kernel density estimation. The algorithm assigns edge pixels to the foreground or the background using a multidimensional feature vector. The assignment probabilities of the feature points are iteratively updated for a few number of iterations.

The resulting edge map of the foreground object is projected onto a human-silhouette shape-space to measure the similarity between the foreground object and the human shape space. Since the shape space is based on binary data, Logistic Principal Component Analysis is used to obtain the eigen shapes of the human silhouettes.

The presented algorithm is used to detect humans in video data acquired from a moving robot. Results prove the effectiveness of the algorithm at finding humans at different






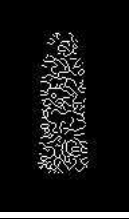





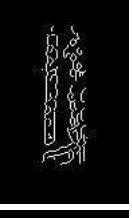



Original Image	Original Edges	Filtered Edges	ϵ^2
			1185949.097
			2093075.204
			1128784.317
			2325075.309
			1145388.299

TABLE I

EXAMPLE IMAGE WITHOUT A HUMAN ALONG WITH THE FILTERED EDGE MAP. THE DISTANCE ϵ^2 FROM THE HUMAN SHAPE SPACE IS VERY LARGE


















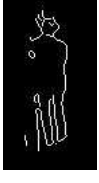
Original Image	Original Edges	Filtered Edges	ϵ^2
			330558.713
			88627.822
			171946.914
			179497.567
			216093.727
			204732.020

TABLE II

EXAMPLE IMAGE CONTAINING A HUMAN ALONG WITH THE FILTERED EDGE MAP. THE DISTANCE ϵ^2 FROM THE HUMAN SHAPE SPACE IS ONE ORDER OF MAGNITUDE SMALLER THAT THAT OF TABLE I

scales.

REFERENCES

- [1] W. Abd-Almageed, B. Burns, and L. Davis, "Identifying and segmenting human motion for mobile robot navigation using alignment errors," in *IEEE International Conference on Advanced Robotics*, 2005.
- [2] W. Abd-Almageed and L. Davis, "Robust appearance modeling for pedestrian and vehicle tracking," in *Lecture Notes in Computer Science 4122, Proc. of Classification of Events, Activities and Relationships (CLEAR) Workshop*, R. Stiefelagen and J. Garofolo, Eds., 2007, pp. 209–215.
- [3] G. Borgefors, "Distance transformations in digital images," *Computer Vision, Graphics and Image Processing*, vol. 34, no. 3, 1986.
- [4] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000.
- [5] N. Dalal and B. Triggs, "Histogram of oriented gradients for human detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.
- [6] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, 1997.
- [7] D. M. Gavrilu and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision*, vol. 73, no. 1, June 2007.
- [8] D. M. Gavrilu and V. Philomin, "Real-time object detection for smart vehicles," in *IEEE International Conference on Computer Vision*, 1999.
- [9] M. Hussein, W. Abd-Almageed, Y. Ran, and L. Davis, "A real-time system for human detection, tracking and verification from a moving camera," in *IEEE International Conference on Computer Vision Systems*, 2006.
- [10] Y. Ran, I. Weiss, Q. Zhen, L. Davis, and W. Abd-Almageed, "Pedestrian classification from moving platforms using cyclic motion pat-

- tern,” in *IEEE International Conference on Image Processing*, 2005.
- [11] A. I. Schein, A. Popescul, L. Ungar, and D. M. Pennock, “A generalized linear model for principal component analysis of binary data,” in *9th International Workshop on Artificial Intelligence and Statistics*, 2003.
 - [12] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2001.
 - [13] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng, “Fast human detection using a cascade of histogram of oriented gradients,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
 - [14] Z. Zivkovic and J. Verbeek, “Transformation invariant component analysis for binary images,” in *In Proc.IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 254–259.