INTEGRATED DETECTION, TRACKING AND RECOGNITION FOR IR VIDEO-BASED VEHICLE CLASSIFICATION

Xue Mei, Shaohua Kevin Zhou[+] and Hao Wu

Center for Automation Research and ECE Department
University of Maryland
College Park, MD 20742
{xuemei,wh2003}@cfar.umd.edu

[+]Integrated Data Systems Department
Siemens Corporate Research
Princeton, NJ 08540
kzhou@scr.siemens.com

ABSTRACT

We present an approach for vehicle classification in IR video sequences by integrating detection, tracking and recognition. The method has two steps. First, the moving target is automatically detected using a detection algorithm. Next, we perform simultaneous tracking and recognition using an appearance-model based particle filter. The tracking result is evaluated at each frame. Low confidence in tracking performance initiates a new cycle of detection, tracking and classification. We demonstrate the robustness of the proposed method using outdoor IR video sequences.

1. INTRODUCTION

Recently, video-based vehicle classification has gained much attention, especially in automatic traffic management, surveillance and battlefield awareness. Typically, detection and tracking are often solved before classification. [1] discusses pose determination and recognition of vehicles in traffic scenes, which under normal conditions stand on the ground-plane. In [2], a segmentation algorithm uses deformable template models to segment a vehicle of interest both from the stationary complex background and other moving vehicles in an image sequence. In addition to segmentation, the deformable template algorithm also classifies the vehicle of interest. In [3], the author describes a system for automatic recognition of vehicle type from frontal views. They only use images and it does not involve tracking. In [4], a method for recognizing a vehicle's maker and model is proposed. It first creates a compressed database of local features of target vehicles from training images and then matches them with the local features of the probe image for recognition.

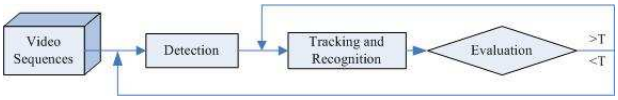

Fig. 1. A flow chart of our system.

In this paper, we tackle the problem of vehicle classification by integrating detection, tracking and recognition. In our system, the moving vehicle is automatically detected, tracked and recognized without any interruptions. The flow chart of our system is shown in Fig.1. The video sequences are input to our system. The moving target is detected using temporal variance analysis. The target is tracked and classified simultaneously using an appearance model and mixtures of probabilistic principal component analysis [5](PPCA). Evaluation of the tracking performance is performed at each frame. If the performance falls below some threshold, the cycle of detection, tracking and classification is re-initiated, otherwise the tracking and classification propagates to the next frame. The target-to-background contrast is very low for the IR images. This adds much difficulty for detection and tracking of the moving target.

Unlike Zhou et al.[6]'s method which manually selects the moving target in the first frame, we automatically select it using temporal variance analysis algorithm. Because of the presence of smoke and dust in IR videos, it is hard to position a tight rectangular bounding box from the detection algorithm. Consequently, the tracker drifts quickly. This brings a need for the evaluation of the tracking performance. The evaluation generates a confidence measure to indicate whether we should restart the detection once the tracking confidence falls below a threshold. In [6], Zhou et al. use sum of squared distance(SSD) for the tracked object and template to give the probability of tracking. Therefore, it gives the same weight to each pixel. Here we propose to use two template matching algorithms, Image Euclidean Distance and Image Weighted Distance, to substitute SSD. These two algorithms are robust to small perturbation and background clutter.

We use mixtures of PPCA[5] for appearance modelling. We then compute the posterior probability of finding the appearance of each object in the given video and assign the label corresponding to the maximum.

The rest of this paper is organized as follows. Section 2 briefly describes detection algorithm. Section 3 describes the template matching algorithms used for the tracking. Section 4 describes tracking and classification algorithm. Section 5 describes the simultaneous evaluation for the tracking and section 6 describes the experiment. Finally, conclusion and future work are discussed in section 7.

2. TARGET DETECTION

Detection plays an important role in our system. It is a prerequisite for tracking and places an initial bounding box around the target and re-initialize the target if tracking confidence measure is low. We briefly review the temporal variance analysis for object detection in the following.

Given a video sequences $\{I_i\}$, we set $m_1 = I_1$ and $mv_1 = I_1 \times I_1$. The operator $\times$ is the element-by-element product of two matrices. The following m_i, mv_i and $imvar_i$ are defined as

$$m_i = ((N-1) * m_{i-1} + I_i)/N \quad (1)$$

$$mv_i = ((N-1) * mv_{i-1} + I_i \times I_i)/N \quad (2)$$

$$imvar_i = \sqrt{mv_i - m_i \times m_i}, \quad (3)$$

where N is the window size for detection which is 150 in our experiment.

For the element $p(i,j)$ in $imvar_i$, we will set $p(i,j) = 1$ if $p(i,j) > T$, otherwise $p(i,j) = 0$, where T is the threshold. Now $imvar_i$ is converted to a binary image which we call the variance image. We then select the rectangular bounding box for the moving target by checking $p(i,j) = 1$ in the image.

3. TEMPLATE MATCHING

Suppose that two images z and t are rasterized to form two vectors, $z = (z_1, z_2, \cdots, z_{MN})$, $t = (t_1, t_2, \cdots, t_{MN})$, where z is the tracking result and t is the template. In [6], the probability of the tracked object given the template is defined as

$$p(z|t) = exp\{-d^2_{SSD}(z,t)\} = exp\{-\sum_{i=1}^{MN}(z_i - t_i)^2\} \quad (4)$$

SSD gives the same weight to each pixel which is not robust to small perturbation and background clutter. In order to solve these difficulties, we propose two template matching algorithms in the following section. The sum of the two distances replaces the SSD $d^2_{SSD}(z,t)$ to give a probability of the tracking result.

3.1. Image Euclidean Distance

Wang et al. [7] propose a new Euclidean distance for images, which is dubbed as Image Euclidean Distance(IMED). Unlike the traditional Euclidean distance, IMED takes into account the spatial relationships of pixels. Therefore, it is robust to small perturbation. The IMED is given by

$$d^2_{IME}(z,t) = \frac{1}{2\pi}\sum_{i,j=1}^{MN} exp\{-|P_i - P_j|^2/2\}(z_i - t_i)(z_j - t_j). \quad (5)$$

where P_i and P_j are two pixels associated with z_i, t_i and z_j, t_j respectively, $|P_i - P_j|$ is the pixel distance.

3.2. Image Weighted Distance

When we use a rectangle or ellipse to select the region of interest, we inevitably include some background in the region of interest. The background will contaminate the template and contribute to tracking failure. Inspired by [8], we propose an image weighted distance method to overcome this problem.

The image weighted distance is given by

$$d^2_{IMW}(z,t) = \sum_{i=1}^{MN} w_i(z_i - t_i)^2 \quad (6)$$

where w_i is the weight assigned to the squared difference of each pixel. The weights are smaller for pixels that are farther from the center. Using these weights increases the robustness of matching since the peripheral pixels are the least reliable, being often affected by occlusion, clutter or interference from the background. The weight function is a 2D Gaussian kernel. Suppose w and h are the width and height of the image, respectively. The weight for the pixel at location (x,y) is

$$w(x,y) = 1 - \frac{1}{2}\{(\frac{x - x_0}{w/2})^2 + (\frac{y - y_0}{h/2})^2\} \quad (7)$$

where x_0 and y_0 is the center of the template.

4. TARGET TRACKING AND CLASSIFICATION

This section describes the vehicle tracking and classification algorithm. In section 4.1, the state space model used for tracking and classification is described. Tracking and classification are implemented simultaneously by estimating the posterior distribution . In section 4.2, mixtures of PPCA is briefly described which is used to estimate the distribution of identity variable for the classification.

4.1. State Space Model

A time series state space model uses the state variable $x_t = \{n_t, \theta_t\}$, which includes identity variable n_t and 2D affine transformation motion parameters θ_t. The system equation is written as

$$n_t = n_{t-1} \qquad \theta_t = \theta_{t-1} + u_t,\ t \geq 1 \quad (8)$$

where we assume that the motion variable follows a Markov process with u_t as a white Gaussian noise process. $n_t \in N = \{1, 2, \cdots, N\}$ indexes the gallery set $\{I_1, I_2, \cdots, I_N\}$.

A simple formulation of the observation equation can be characterized as

$$Z_t = T\{Y_t; \theta_t\} = I_{n_t} + V_t \quad (9)$$

Where Z_t is the image patch of interest in the video frame, T is an affine transformation to normalize the image to the same size of the gallery images, and V_t is the noise. The observation equation is equivalently characterized by the likelihood $p(Y_t|n_t, \theta_t) = p(Z_t|n_t)$. In the next section, we define $p(Z_t|n_t)$ as mixtures of PPCA.

The essence of the framework is posterior probability computation, i.e. computing $p(n_t, \theta_t|Y_{1:t})$, whose marginal posterior probability $p(n_t|Y_{1:t})$ solves the classification task and marginal posterior probability $p(\theta_t|Y_{1:t})$ solves the tracking task.

Classification is based on a Maximum A Posteriori (MAP) decision rule, namely finding n_t that maximizes $p(n_t|Y_{1:t})$. The Sequential Importance Sampling(SIS)[9] method is used to approximate and propagate the posterior probability $p(n_t, \theta_t|Y_{1:t})$, and marginalization over variable θ_t is carried out before applying the classification rule. Detailed descriptions can be found in [6].

4.2. Mixtures of Probabilistic PCA

Subspace analysis techniques have attracted growing interest in computer vision research. In particular, eigenvector decomposition has been shown to be an effective tool for solving problems by using low-dimensional vector to represent high-dimensional vector. Here we will follow [5] for the mixtures of PPCA.

Given a set of m by n images $\{Z_i\}$, we form a set of vectors $\{t_i\}$, where $t_i \in R^{d=mn}$, by lexicographic ordering of the pixel elements of each image Z_i. For any t in $\{t_i\}$, we relate it to a corresponding γ-dimensional vector variable x as: $t = Wx + \mu + \epsilon$, where $d >> \gamma$ and μ is the mean of t. First, in PPCA, it is assumed that $x \sim N(0, I)$ and $\epsilon \sim N(0, \sigma^2 I)$. Thus, we obtain the distribution of t as

$$p(t) = (2\pi)^{-d/2}|C|^{-1/2}exp\{-\frac{1}{2}(t-\mu)^T C^{-1}(t-\mu)\}, \quad (10)$$

where the covariance is $C = \sigma^2 I + WW^T$.

The mixtures of PPCA can model more complex data structures. The model parameters are determined using maximum likelihood estimation. The mixture model is defined as

$$p(t) = \sum_{i=1}^{M} \pi_i p(t|i) \quad (11)$$

where $p(t|i)$ is a single PPCA model and π_i is the corresponding mixing proportion, with $\pi_i \geq 0$ and $\sum \pi_i = 1$. Now the three parameters μ, W and σ^2 are associated with each of the M mixture components. We use an iterative EM algorithm for estimation of the model parameters.

5. TRACKING EVALUATION

Most practical tracking systems often fail under some situations. This could be either because of illumination changes, pose variation or occlusion. Therefore, the need for automatic performance evaluation emerges in these applications. Fig.2 shows the tracking result after running the tracker for some time. The bounding box is so large that one concludes that the tracker is already failing. Hence, evaluation is necessary to help us terminate tracking and restart the detection-tracking-classification sequence.

Fig. 2. The vehicle is off tracking.

Our evaluation algorithm is based on measuring the appearance similarity and tracking uncertainty. The following features are examined in our evaluation:

1. Trace complexity q_{tc}: We define the trace complexity as the ratio of the curve length and straight length between the target centroids in different frames.
2. Motion step q_{ms}: It is defined as the distance between the box centers in two consecutive frames.
3. Scale change q_{sc}: To examine changes in object scale, we use two clues. One is the ratio of the current area to the initial area, the other is the scale change velocity.
4. Shape similarity q_{ss}: The change in the aspect ratio of the bounding box is also useful in providing some information about the object shape. It is defined as the ratio of the current aspect ratio over the initial ratio.
5. Appearance change q_{ac}: Three measures are used in our algorithm, the first one is the absolute pixel by pixel change between the current frame and the initial frame, the second one is the histogram difference between the current frame and the initial frame and the last one is related to the tracking algorithm over which the proposed algorithm was tested.

To obtain a comprehensive measure of the tracking performance, we combine the above five indicators. We first use empirical thresholds to find whether the tracker is uncertain according to the above five metrics, then we sum the five indicators using different weights to arrive at a confidence measure q. If the sum drops below some threshold, we conclude that the tracking performance is poor and needs re-initialization.

$$q = \sum_{j \in J} w_j I[q_j < \lambda_j], \; J \in \{tc, ms, sc, ss, ac\} \qquad (12)$$

where w_j and λ_j are the corresponding weights and thresholds for the evaluation.

6. EXPERIMENTS

In this section, we give details of our implementation. Training and testing are described in the next two sections respectively. In our experiment, the vehicle motion is characterized by $\theta = (a_1, a_2, a_3, a_4, t_x, t_y)$, where $\{a_1, a_2, a_3, a_4\}$ are the deformation parameters and (t_x, t_y) are the 2D translation parameters. By applying an affine transformation using θ as parameters, we crop the region of interest so that it has the same size as the still template in the gallery and perform zero-mean-unit-variance normalization. The region of interest is 24×30 in size.

6.1. Training

We use one video sequence for each vehicle and obtain the tracking result. Then we select 36 images for each vehicle in the gallery. There are a total of 144 images in the gallery. They are 'm60', 'brdm', 'wetting' and 'bmp'. The pertinent parameters for the experiment are $M = 2$ and $\gamma = 15$. After we have the gallery images, we use mixtures of PPCA to estimate the parameters π_i,μ_i, W_i and σ_i^2.

6.2. Testing

For each frame, we get the motion parameters after tracking and cropping out the region of interest from the original image. After performing zero mean and unit variance operation, we use the result to estimate the posterior probabilities of observing each vehicle. We pick the vehicle which has the highest probability as our classification result after normalization. The probabilities propagate to the next frame. In each frame, if the confidence measure is below some threshold, the detection will restart 20 frames before the drifting point and tracking and classification will restart too.

Fig. 3 shows the tracking and recognition results for 'wetting1' and Fig. 4 is for 'bmp1'. In Fig. 3, The image to the left is the tracking result for the current frame. We put a bounding box for the vehicle which we are tracking in each frame with a different color for different vehicles. The image in the middle is the classification score which is the probability of seeing each vehicle in the video. It shows the result from the first frame to the current frame. The image to the right is the tracking confidence measure which represents the probability of the correct tracking result. We will restart detection and tracking if the measure falls below the threshold of 0.5. The same description applies to Fig. 4.

From Fig. 3, we observe that the recognition result for the 'wetting1' is very good because a high probability is associated with 'wetting' (dotted blue line) on almost every frame. There are several peaks and valleys for the dotted blue line due to the re-initialization of the tracking and the evaluation probability on the right drops very quickly at corresponding frames. In Fig. 4, for the recognition of 'bmp1', it is confused by 'brdm' for first half of the sequence. The tracker quickly drifts away after about 40 frames given the initial location. The result becomes stable and correct after 400 frames. After running the whole video sequence, the correct recognition result is quite good. For this situation, we will classify that the vehicle we are tracking is 'bmp' which yields the correct result.

We divide the twenty probe video sequences into five groups. Each group has each of the four different vehicles. The classification results of one group are summarized in Table 1. Each number in

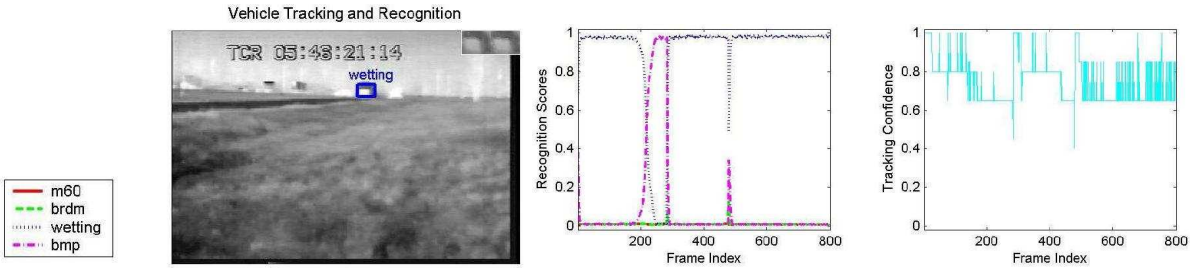

Fig. 3. Tracking and recognition results for 'wetting1'. The results are from frame 1 to 799. The left panel shows the original image and tracking result, the middle panel shows the recognition density $p(n_t|Y_{1:t})$, and the right panel shows the tracking confidence q.

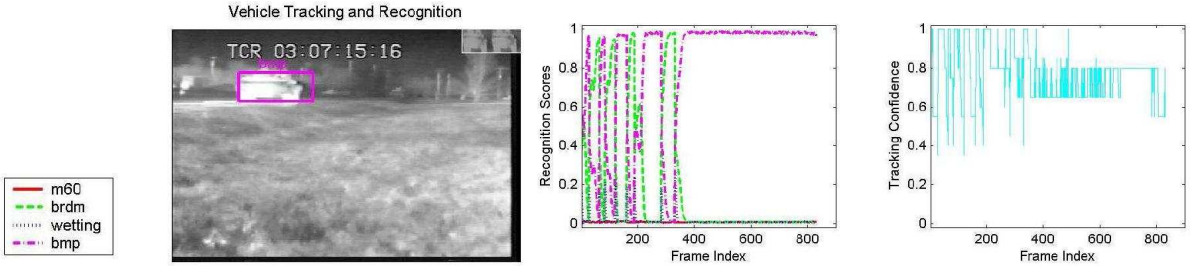

Fig. 4. Tracking and recognition results for 'bmp1'. The results are from frame 1 to 830.

	m60	brdm	wetting	bmp
m60	93.82%	3.17%	0	3.01%
brdm	0	85.64%	0	14.36%
wetting	0	0	95.65%	4.35%
bmp	0	18.85%	0	81.15%

Table 1. Confusion matrix for vehicle classification experiment.

a row is the recognition percentage of the vehicle. Taking the second row as an example, 93.82% of the whole sequence recognizes the vehicle as 'bmp', while 3.17% as 'brdm' and 3.01% as 'bmp'. No frame recognizes it as 'wetting'. The elements in the diagonal give the correct recognition score for our experiment. The overall accuracy of the recognition is 89.07%. Our experiment results show all the twenty probe video sequences can be classified correctly using our proposed method.

7. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an approach for vehicle classification by integrating detection, tracking and recognition. The experiment results prove our method's robustness and effectiveness.

Our future work will include improving detection, tracking and evaluation algorithms and developing a more robust and stable recognition algorithm. Large data set will also be tested to obtain a more general analysis.

8. ACKNOWLEDGEMENTS

We would like to thank Prof. Rama Chellappa for very helpful and encouraging suggestions.

9. REFERENCES

[1] T. N. Tan, G. D. Sullivan, and K. D. Baker, "Model-based localisation and recognition of road vehicles," *International Journal of Computer Vision*, vol. 27, pp. 5–25, January 1998.

[2] M.-P. D. Jolly, S. Lakshmanan, and A. K. Jain, "Vehicle segmentation and classification using deformable templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 293–308, March 1996.

[3] V.S. Petrovic and T.F. Cootes, "Vehicle type recognition with match refinement," *International Conference on Pattern Recognition*, vol. 3, pp. 95–98, August 2004.

[4] M. Kagesawa, S. Ueno, K. Ikeuchi, and H. Kashiwagi, "Recognizing vehicles in infrared images using imap parallel vision board," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, pp. 10–17, March 2001.

[5] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic princial component analysers," *Neural Computing*, vol. 11, pp. 443–482, 1999.

[6] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, pp. 1057–7149, 2004.

[7] L. Wang, Y. Zhang, and J. Feng, "On the euclidean distance of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1334–1339, 2005.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 564–577, 2003.

[9] J. S. Liu and R. Chen, "Sequential monte carlo for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1031–1041, 1998.