Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Towards Detection and Tracking of On-Road Objects

Roland Goecke[1,2], Niklas Pettersson[1,2] and Lars Petersson[1,2]
[1]Vision Science, Technology and Applications, NICTA Canberra Research Laboratory, Canberra, Australia
[2]Dept. of Information Engineering, RSISE, Australian National University, Canberra, Australia
roland.goecke@anu.edu.au {niklas.pettersson, lars.petersson}@nicta.com.au

Abstract— **In this paper, we present a system capable of detecting and tracking on-road objects in the scene, in particular vehicles. Such a system is a useful part of a driver assistance system. This system employs two different techniques in the detection phase to increase the robustness. A large part of this paper is devoted to reducing the computational amount required of the overall algorithm by quickly excluding pixels above the horizon and on the road surface. The number of pixels that require further, computationally expensive processing is reduced by up to 65% in the sequences used in the experimental evaluation. Objects are detected in the remaining image areas by an improved boosting approach of weak classifiers based on the well-known AdaBoost and RealBoost approaches. The tracking is then done by a combination of periodically running the detection algorithm, while using adaptable templates at other times which allow for changes in shape and appearance as the car and the other vehicles travel along the road.**

I. INTRODUCTION

Driver assistance systems have been the focus of much research in recent years and the results of that research can increasingly be found in current production cars. The aim of such systems is to improve the safety for the driver, passengers, and other road traffic participants (e.g. drivers of other vehicles, pedestrians, cyclists). Various forms of driver assistance systems exist; the work presented in this paper is concerned with the visual detection and tracking of vehicles - and generally speaking, other on-road objects - around the driver's car. The detection and tracking of such objects is a pre-requisite for the detection of potential collisions and, thus, for warning the driver of these.

The detection of on-road objects, i.e. the separation of foreground objects is a non-trivial problem. Both the driver's car as well as the on-road objects may be moving, often at different speeds, the scene background is constantly changing, the visual appearance of the road might be changing, and the environmental conditions (illumination due to weather and time of day) might be changing. We believe that only a multiple cue system can deliver the robustness required for practical applications in such conditions. Single cues are prone to give erroneous results in one condition or another. Integrating multiple, complementary cues can overcome these problems and lead to improved robustness. Our approach to the detection of on-road objects therefore uses a colour background model of the visual appearance of the road surface and the entropy of image patches as individual cues. Other cues such as optical flow and 2D wavelet analysis are also possible and are currently investigated.

Once potential on-road objects have been detected, adaptable (or dynamic) template matching is used for object tracking. A static template matching approach would quickly fail as the shape and appearance of the on-road objects changes as the driver's car and the objects move and hence the pose changes. Similarly, changes in the environmental conditions can lead to changes in the visual appearance of the objects. In adaptable template matching, the templates are updated at every time step, so that the tracking always works with the most recent object shape and appearance.

The remainder of this paper is organised as follows. Section II gives an overview of related work in the area of on-road object detection and tracking. Next, Section III shows a way of estimating the location of the horizon in the video frame, because only points below the horizon line need to be further processed, thus reducing the computational cost. Section IV presents our method for the detection of on-road objects and Section V details the method for tracking such objects. Results and the discussion can be found in Section VI. Finally, the conclusions are drawn in Section VII.

II. RELATED WORK

The task of separating foreground, on-road objects from the road and scene background is similar to the task of background modelling in many computer vision applications. Many background modelling algorithms have been proposed in the literature; only some can be mentioned here. Pfinder [1] utilised 2D statistical models of the object (e.g. a person) and modelled the background as a texture surface with each background pixel being modelled by a Gaussian. These foreground and background models are updated at each time step. Stauffer and Grimson [2] presented a real-time background subtraction algorithm that uses a Gaussian mixture model to model a changing background and to adapt to chaning visual conditions. Ridder *et al.* [3] employ Kalman filtering for estimating an adaptive background, so that their method can tolerate illumination changes. Ohta [4] presented another statistical approach to model a changing background. The model represents the relation between the pixel values, the reflection index of an object point, and the illumination intensity of an object point. Elgammal *et al.* [5] proposed a method using a kernel estimator function to obtain a non-parametric model of the probability distribution of each pixel's intensity value. The model is said to adapt quickly to changes in the scene caused by objects with small motion (e.g. tree branches). Monnet *et al.* [6] followed a

1-4244-1068-1/07/$25.00 ©2007 IEEE.

dynamic texture approach with an online, auto-regressive model to model dynamic scenes. Kahl *et al.* [7] proposed an adaptive background model based on a linear PCA model in combination with local spatial transformations.

A number of vanishing line estimation methods have been proposed in the past. These can be broadly classified into voting approaches using the Hough Transform and statistical approaches. Nakatani *et al.* [8] proposed a method based on counting points on a sinusoidal curve in the Hough plane. McLean and Koyyuri [9] proposed a method of vanishing point detection by line clustering based on fan-shaped error.

A popular method in object detection is the AdaBoost approach proposed by Viola and Jones [10] which uses a cascade of weak classifiers, Haar-like features, to learn a detector from a large number of training examples. The approach provides a fast way of detecting objects at any scale due to using the concept of integral images. The Haar-like features themselves are limited but their performance can be drastically improved through a boosting process. The choice of a single threshold in the original AdaBoost boosting method is sub-optimal, as shown by [11], since AdaBoost does not consider all training examples equally. In [12], performance was improved by using real-valued weak classifiers, thus, creating a RealBoost algorithm. Rasoldazeh *et al.* [11] generalised these methods to a multiple thresholded classifier, defined by a *maximum a posteriori* rule, which turns out to be a specific implementation of the higher level concept of response binning. We use this approach for the object detection phase in the work presented here.

A common method in object tracking is template matching, in which a part of the original image is moved across the current image while computing some similarity measure. The position with the highest similarity measure is then deemed to be the position of the original image patch in the current image. Similarity measures include, for example, the sum of absolute differences, the sum of squared differences, or the normalised cross correlation. If neither of the tracked object's shape, visual appearance, and pose change, static template matching is in an appropriate tracking method. However, such an approach quickly fails when the object's shape, appearance, or pose change. One way to overcome these problems is to use adaptable (or dynamic) templates. Here, the templates are updated regularly, so that the tracking always works with the current object shape, appearance, and pose. Such an approach was described by Loy *et al.* in [13]. There, the updated template $T_i(k)$ for the k^{th} video frame was a weighted average of the initial template $T_i(0)$ and the best match of the previous template image in frame $k-1$

$$T_i(k) \;=\; \beta\, T_i(0) \;+\; (1-\beta)\, T_i(k-1), \quad \beta \in [0,1]. \quad (1)$$

The weighting factor β determined the contribution of the initial template image. According to Loy *et al.* [14], 'grounding' the template image is necessary, because fully updated templates have the tendency to 'drift' over the image after some time, due to the quantisation error and possible mismatches. In the work presented here, we avoid the 'drifting' problem by re-initialising the template once per second through the response binning detection process.

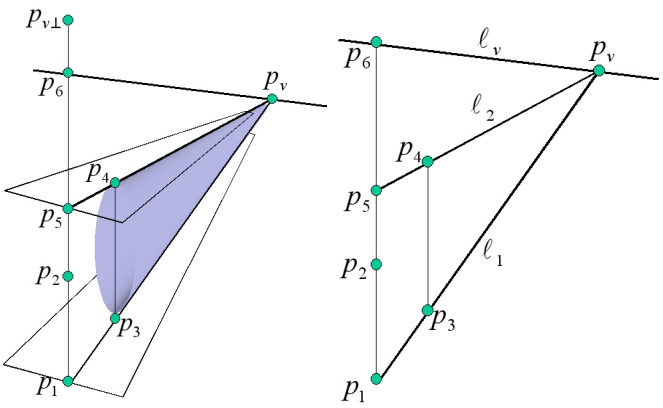

Fig. 1. Left: Geometry of the photogrammetry process using parallel planes. Right: Geometry of the pencil used to estimate the height of an on-road object and the vanishing line parameters.

III. Estimating the Horizon

Before turning our attention to the problem of detecting on-road objects, let us first estimate the horizon in the perspective camera projection. We can do so by estimating the location of the vanishing line which will serve as horizon in further processing. Only image pixels below the horizon line are processed which can save a significant amount of computation, depending on the scene geometry as seen by the camera. Automatically estimating the horizon is a safer option than using a heuristic value for the horizon because the apparent location of the horizon can change quickly when the vehicle drives over a bumpy surface, e.g. a speed bump.

It is a safe assumption that any on-road object is located on the ground plane. In the left-hand panel of Figure 1, $p_3 = [x_3, y_3, 1]^T$ and $p_4 = [x_4, y_4, 1]^T$ correspond to the bottom and top of an on-road object, respectively, $p_1 = [x_1, y_1, 1]^T$ and $p_2 = [x_2, y_2, 1]^T$ are reference points, $p_v = [x_v, y_v, 1]^T$ is the vanishing point on the horizon, i.e. the vanishing line $\ell_v = m_v x + b_v y$, and $p_{v\perp} = [x_{v\perp}, y_{v\perp}, 1]^T$ is the point at infinity on the line perpendicular to the ground plane. The point $p_6 = [x_6, y_6, 1]^T$ is at the intersection of the line going through p_1 and p_2 with the vanishing line ℓ_v.

Initially, we assume that the two reference points are known in the scene and that these points are on a line perpendicular to the ground plane. As we see later, only p_1 is needed and we take this point to be the camera position on the ground plane. We also assume that the y-axis of the camera is parallel to the lines going through the reference points and the ground plane and the top of the vehicle. These assumptions allow us to simplify the computations that will follow. Since the camera coordinate system is such that the x-coordinates of the reference points and the camera are the same, we can reduce the number of degrees of freedom by setting $x_1 = x_6$ and $x_3 = x_4$. It is worth stressing that, since our aims of computation are the slope and intersect of the vanishing line, this choice of camera coordinate system is

arbitrary and can be modelled using a rotation, which is an affine transformation on the image plane.

Here, we make use of perspective geometry and, more specifically, of Desargues theorem [16]. Recall that Desargues theorem states that, in a perspective space, if three straight lines joining the vertices of two triangles all meet in a point, i.e. the vanishing point, then the three intersections of pairs of corresponding sides lie on a straight line. As a result, points which are perspective from a point are also perspective from a line and, therefore, the cross ratio α between them should remain constant. The cross ratio α for the configuration in Figure 1 is given by

$$\alpha = \frac{\mid p_{v\perp} - p_6 \mid\mid p_1 - p_5 \mid}{\beta \mid p_{v\perp} - p_5 \mid\mid p_1 - p_6 \mid} \tag{2}$$

where $\beta = \mid p_1 - p_2 \mid$ is the distance between the two reference points p_1 and p_2. But, since the point $p_{v\perp}$ is a point at infinity, it can be shown that $\frac{|p_{v\perp}-p_6|}{|p_{v\perp}-p_5|} = 1$ and, hence, Equation 2 becomes

$$\alpha = \frac{\mid p_1 - p_5 \mid}{\beta \mid p_1 - p_6 \mid} \tag{3}$$

From Figure 1, it is clear that the lines ℓ_1 and ℓ_2, together with the vanishing line ℓ_v form a pencil whose intersect is at the vanishing point p_v. In the right-hand panel of Figure 1, we show the geometry of the pencil. Assuming p_1 and p_3 are known, i.e. their image locations have been determined (e.g. road lines, curbs), we can express the points p_5, p_6 and p_v in homogeneous coordinates as follows

$$\begin{aligned} p_v &= [x_v, y_v, 1]^T &&= \left[\frac{b_v - b_1}{m_1 - m_v}, m_1\frac{b_v - b_1}{m_1 - m_v} + b_1, 1\right]^T \\ p_6 &= [x_6, y_6, 1]^T &&= [x_1, m_v x_1 + b_v, 1]^T \\ p_5 &= [x_5, y_5, 1]^T &&= [x_1, m_2 x_1 + b_2, 1]^T \end{aligned} \tag{4}$$

where the intersect and slope for lines ℓ_1 and ℓ_2 are given by

$$\begin{aligned} m_1 &= \frac{y_1 - y_3}{x_3 - x_1}; & b_1 &= y_3 - m_1 x_3 \\ m_2 &= \frac{y_4 - y_v}{x_v - x_4}; & b_2 &= y_4 - m_2 x_4 \end{aligned} \tag{5}$$

From the equations above, it is clear that the cross ratio α only depends on the slope m_v and intersect b_v of the vanishing line ℓ_v. To exploit this to our advantage, we substitute the equations above into Equation 3 and write

$$\alpha = \frac{\mid a_1 \mid}{\mid a_2 b_v + m_v a_3 + \gamma \mid} \tag{6}$$

where $a_1 = p_3 \times p_1 + p_1 \times p_4 + p_4 \times p_3$, $a_2 = x_1 - x_3$, $a_3 = x_4(x_1 - x_3)$ and $\gamma = y_1(x_3 - x_4) + y_3(x_1 + x_4)$.

We can remove the absolute value by working with the square of α. Further, by using the shorthands $\varphi = \frac{1}{\alpha}$ and $\hat{a}_1 = \frac{1}{a_1}$, we can rearrange the equation above as $(\varphi\hat{a}_1)^2 = (a_2 b_v + m_v a_3 + \gamma)^2$, which is a quadratic whose parameters can be recovered using Bayesian theory by employing a least-squares estimator.

To do this, we make use of the normal linear model, which can be expressed, in compact form, as $\mathbf{y} = \mathbf{X}\theta + \epsilon$, where ϵ is a vector of random errors, θ is the vector of parameters that govern the model, $\mathbf{X}$ is a matrix of known coefficients and $\mathbf{y}$ is the vector of independent terms γ^2 for each of the n frames of the video sequence under study.

Thus, we can view the independent terms γ^2 as samples and assume that the random errors are uncorrelated, have zero mean, common variance σ^2 and that the conditional distribution of $\mathbf{y}$, given θ, σ^2, is governed by a multivariate normal distribution. As a result, the likelihood becomes

$$L(\mathbf{y} \mid \theta, \sigma^2) = \left(\frac{1}{2\pi\sigma^2}\right)^{-\frac{1}{n}} \exp\left\{-\frac{(\mathbf{y} - \mathbf{X}\theta)^T(\mathbf{y} - \mathbf{X}\theta)}{2\sigma^2}\right\} \tag{7}$$

Note that, if the matrix $(\mathbf{y} - (X)\theta)^T(\mathbf{y} - (X)\theta)$ is not singular, we can complete the square in the exponent of Equation 7 as follows $(\mathbf{y} - \mathbf{X}\theta)^T(\mathbf{y} - \mathbf{X}\theta) = (\theta - \hat{\theta})^T\mathbf{X}^T\mathbf{X}(\theta - \hat{\theta}) + \mathbf{S}$, where $\mathbf{S} = (\mathbf{y} - \mathbf{X}\hat{\theta})^T(\mathbf{y} - \mathbf{X}\hat{\theta})$ is the residual for the estimator and $\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ is the maximum likelihood for the vector of estimated parameters $\hat{\theta}$. Thus, $\hat{\theta}$ is the vector of estimated α^2, slope m_v and intersect b_v that we aim to recover.

IV. On-Road Object Detection

It is worthwhile to note that the work presented in this paper is not concerned with the classification of detected on-road objects into classes like cars, pedestrians, cyclists, and so on.

As a first step, we attempt to quickly eliminate image pixels that correspond to the road surface because the performance can be improved if such pixels are excluded from further processing. The important issue here is that this approach is only useful, if such pixels can be found faster than the computation would take, if they were not included. As outlined earlier, we believe that only a multiple cue system can deliver the robustness required for practical applications, as single cues tend to fail in one condition or another. In our system, we currently use two cues: a colour background model for the road and the entropy of image patches which can both be computed in real-time. The multi-cue framework is extendible and other cues could be included. In the following, we will present these individual cues in more detail, before turning our attention to the issue of how to integrate the results of the individual cues.

A. Using a Colour Road Background Model

First, a statistical model of the colour of the road surface is built from a training set of images. It would be possible to build a model from known road surface pixel locations, but we have found that it suffices to build a Gaussian model for each pixel by accumulating the colour information over the entire training set, under the assumption that the video sequence was not recorded in heavy traffic, i.e. we assume that a reasonable amount of road surface is visible. A training set as small as 100 images gave good results in our experimental evaluation. Colour information is gathered by first transforming the image colour space to HSI colour

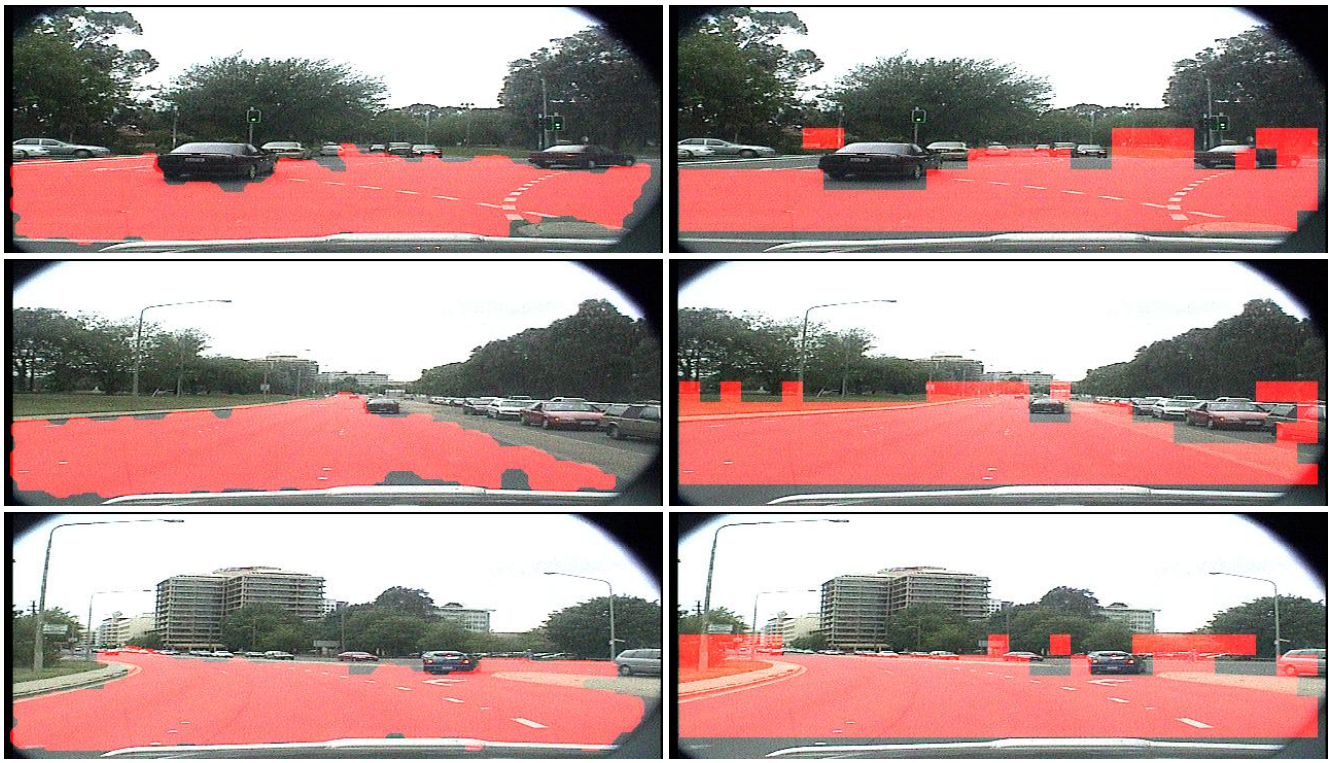

Fig. 2. Road detection in three example frames. Left: Using a colour road background model. Right: Using the entropy of image patches.

space [17], where the effects of intensity are separated from hue and saturation. Our road surface model is based on the information in the latter two.

Once such a model is available, the current input frame is first converted to HSI colour space. Then, the following inequalities are applied to decide whether a pixel belongs to the road surface

$$\text{Road} = \begin{cases} 1 & \text{if} \quad |H(x,y) - \bar{H}| \leq 3\sigma_H \\ & \text{AND} \quad |S(x,y) - \bar{S}| \leq 3\sigma_S \\ 0 & \text{else} \end{cases} \tag{8}$$

where H and S are the current hue and saturation values, $\bar{H}$ and $\bar{S}$ are the mean of hue and saturation, and σ_H and σ_S are the standard deviation of hue and saturation. If hue and saturation are within three standard deviations from the current mean, the pixel (x, y) is deemed to be on the road surface. After all pixels have been processed, the Gaussian models for hue and saturation are updated so that the algorithm is able to adapt to changes in road surface colour. At any one time, we keep a maximum of 1000 samples for each model, in order to have a good, yet computationally not expensive model.

In addition, it is possible to build better masks of road surface areas by taking advantage of the spatial relationships between road surface pixels. In other words, if a pixel has been classified as belonging to the road surface class, then the likelihood is quite high that its neighbouring pixels are also members of that class. We can take advantage of this property by applying a symmetric 2D Gaussian filters to the binary mask image

$$f(x,y) = exp\left(-\frac{1}{2}\frac{(x^2+y^2)}{s^2}\right) \tag{9}$$

where s represents the scale which can be used to change the size of the 'footprint' of the Gaussian. A cascade of 2D Gaussian filters allows to find connected regions and to remove noisy and falsely detected pixels. We use a cascade of three Gaussian filters with decreasing s, i.e. increasing footprint. Each filter is moved across all pixels in the region of interest and the convolution is computed. The first filter removes spurious, noisy pixels that are classified as road surface but are not connected to or close to any other road surface pixels. The second filter, with a footprint larger than the first Gaussian filter, then attempts to connect any remaining road surface pixels so as to create connected regions. Finally, the third Gaussian filter, again with a larger footprint than the previous filter, performs a similar operation to the first filter and removes mostly pixels along the rim of any road surface area found. Doing so pixels that were wrongly added to the road surface area by the second Gaussian filter. In an approximation, the cascade of Gaussian filters can also be implemented as morphological operations, which can be done in real-time.

B. Using Entropy

As another cue in our system, we employ the entropy measure (in the information theory sense). Specifically, we

use the Shannon entropy [18]

$$H(I_k) = -\sum_{i=1}^{n} p(i)\, log_2 p(i) \tag{10}$$

and apply it to rectangular patches $I_k[x - w...x + w, y - w...y + w]$ of the k^{th} video frame I_k where w denotes half the window size in each dimension. The $p(i)$ then become the entries of the co-occurrence matrix. The pixel positions (x, y) of the image patches are chosen in such a way that all pixels are converted once and only once, i.e. the updates are $x = x + 2w$ and $y = y + 2w$, respectively (obviously assuming that we move to the start of a line or row once the end of the previous line or row of image patches has been reached). The entropy $H(I_k)$ is computed for each image patch.

If $H(I_k) > d_{\text{thresh}}$, then the image patch is taken as containing (a substantial amount of) foreground objects. Otherwise, the image patch is considered as belonging to the background scene. A suitable value for d_{thresh} can be learnt from labelled training images. It should be noted that this approach does not give as detailed information as the road background model described in Section IV-A but is faster to compute. Smaller patch sizes would enable more detailed results but would also give rise to more falsely classified patches. Since we are using all cues in a joint fashion, we found empirically that $w = 10$ provides a good compromise between level of detail and computational speed.

C. Multiple Cue Integration

A central issue in any multiple cue or multiple sensor system is the issue of integrating the results from the individual cues. In the work presented here, we follow a voting approach in which each of the individual cues votes for a pixel to belong to the road surface or not. Our system requires both cues to be in favour of a pixel belonging to the road surface, before that pixel is marked as such. Figure 3 shows results for three example video frames.

With the methods described here, the number of pixels to be processed further with more computationally expensive methods is reduced by up to 65% on average. The reduction in overall computation is not quite as big as that because the background model and entropy methods themselves have a certain computational cost but this cost is much lower than, for example, performing a boosted classification trained to detect vehicles on the entire image.

D. Response Binning for Object Detection

As described in Section II, the approach we use for the actual detection of vehicles in the remaining image areas is that of *Response Binning* [11], which is an extension of the AdaBoost and RealBoost approaches introduced by Viola and Jones [10], [12].

To date, our work has focussed on detecting other cars in the scene, which have a reasonable amount of similarity in their appearance. Separate classifiers would need to be learnt for other vehicles, e.g. trucks. We do so by training a classifier from 2037 hand-labelled training examples of cars

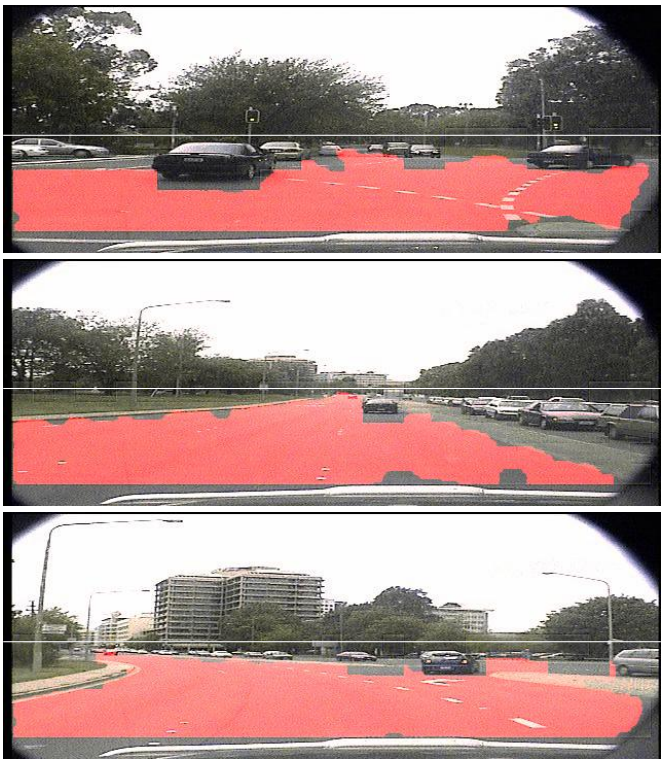

Fig. 3. Road detection (red areas) and horizon estimation (white line) in three example frames using multiple cues.

(positive set) and 5000 training examples not containing cars (negative set). Once the classifier had been learnt, it was applied to the remaining image parts to detect cars. Figure 4 shows an example result of the detection phase.

V. On-Road Object Tracking

Once candidate objects have been found, the adaptable template matching method described in Section II is used to track candidates over a sequence of video frames. While tracking, rather than just relying on the adaptable template matching approach, the position of vehicle candidates is also cross-checked with suggested candidate locations from the *Response Binning* method. However, doing both at all times would be computationally too expensive and so we only perform the cross-check once per second, while in the mean time we use the adaptable template matching approach to quickly track the vehicle candidates within a small window around the previous location. In the experiments, a search window of ± 20 pixels in x-direction and ± 10 in y-direction was used, based on the image resolution and the fact that larger apparent movements occurred in the horizontal direction. Figure 4 shows an example result of the detection and tracking.

VI. Results and Discussion

Experiments have been performed with an image sequence of over 2500 images taken by a camera system mounted in a car while driving through an urban environment. The resolution of the images is 640×240, the frame rate 60Hz.

Fig. 4. Example of vehicle detection and tracking.

As can be seen from Figure 4, clearly visible vehicles are detected and tracked well. However, partially occluded vehicles are not always detected correctly with the current template approach. Improving this will be one aspect of future work. The current unoptimised implementation of our approach can achieve a frame rate of up to 10Hz, depending on the size of the remaining image area to be processed after the pre-processing as well as on the number of vehicle candidates to be tracked.

VII. Conclusions

We have presented a single-camera system for detecting and tracking on-road objects, in particular other vehicles around the driver's car. Much of the work to date has been on finding ways to reduce the computational amount required of the overall algorithm by quickly excluding pixels above the horizon and on the road surface. A novel horizon estimation method has been proposed to this end. Our system employs currently two different cues to separate on-road objects from the road background, but further cues could be added. A voting mechanism is used to integrate the results of the individual cues to deliver a combined result. On-road objects are tracked using adaptable templates which allow for changes to the shape and appearance of objects over time. Promising results for some test video sequences have been shown.

In future work, we will investigate further cues to the detection system to further improve the results. Such cues will include optical flow and 2D wavelet analysis. Furthermore, we plan to use 2D active appearance models [19] or 3D morphable models to improve the vehicle tracking process to be more robust to illumination changes. These methods build statistical models of the appearance of objects which is expected to help with the difference in shape and texture of vehicles. We will also work on improving the detection and tracking of partially occluded vehicles which is a common occurrence in on-road object tracking.

VIII. ACKNOWLEDGMENTS

The authors would like to thank Antonio Robles-Kelly of the Vision Science, Technology and Applications program at the NICTA Canberra Research Laboratory for his help with the vanishing line estimation.

National ICT Australia is funded by the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

References

[1] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, July 1997.

[2] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR1999*, vol. 2. Fort Collins (CO), USA: IEEE, June 1999, pp. 246–252.

[3] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive Background Estimation and Foreground Detection using Kalman-Filtering," in *Proceedings of the International Conference on Recent Advances in Mechatronics ICRAM95*. Istanbul, Turkey: UNESCO Chair on Mechatronics, Aug. 1995, pp. 193–199.

[4] N. Ohta, "A Statistical Approach to Background Subtraction for Surveillance Systems," in *Proceedings of the Eighth IEEE International Conference on Computer Vision ICCV2001*, vol. 2. Vancouver, Canada: IEEE, July 2001, pp. 481–486.

[5] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric Model for Background Subtraction," in *Proceedings of the 6th European Conference on Computer Vision ECCV2000*, ser. Lecture Notes in Computer Science LNCS 1843, vol. II. Dublin, Ireland: Springer, June 2000, pp. 751–767.

[6] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background Modeling and Subtraction of Dynamic Scenes," in *Proceedings of the Ninth IEEE International Conference on Computer Vision ICCV2003*. Nice, France: IEEE Computer Society, Oct. 2003, pp. 1305–1312.

[7] F. Kahl, R. Hartley, and V. Hilsensetin, "Novelty Detection in Image Sequences with Dynamic Background," in *Statistical Methods in Video Processing: ECCV 2004 Workshop SMVP 2004*, ser. Lecture Notes in Computer Science, D. Comaniciu, R. Mester, K. Kanatani, and D. Suter, Eds., vol. 3247. Prague, Czech Republic: Springer, May 2004, pp. 117–128.

[8] H. Nakatani, S. Kimura, O. Saito, and T. Kitahashi, "Extraction of vanishing point and its application to scene analysis based on image sequence," in *Proc. 5th ICPR*, 1980, pp. 370–372.

[9] C. McLean and D. Koyyuri, "Vanishing Point Detection by Line Clustering," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1090–1095, Nov. 1995.

[10] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR2001*, vol. 1. Kauai, USA: IEEE, Dec. 2001, pp. 511–518.

[11] B. Rasolzadeh, L. Petersson, and N. Pettersson, "Response Binning: Improved Weak Classifiers for Boosting," in *Proceedings of the 2006 IEEE Intelligent Vehicles Symposium IV2006*. Tokyo, Japan: IEEE, June 2006, pp. 344–349.

[12] P. Viola and M. Jones, "Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade," in *Proceedings of Advances in Neural Information Processing Systems 14 (NIPS2001)*. Vancouver, Canada: MIT Press, Dec. 2001, pp. 1311–1318.

[13] G. Loy, R. Goecke, S. Rougeaux, and A. Zelinsky, "Stereo 3D Lip Tracking," in *Proceedings of the Sixth International Conference on Control, Automation, Robotics and Computer Vision ICARCV2000*, Singapore, Dec. 2000, on CD-ROM.

[14] G. Loy, E. Holden, and R. Owens, "A 3D Head Tracker for an Automatic Lipreading System," in *Proceedings of the Australian Conference on Robotics and Automation ACRA2000*, Melbourne, Australia, Aug. 2000, pp. 37–42.

[15] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge University Press, June 2003.

[16] H. S. M. Coxeter and S. L. Greitzer, "Perspective triangles; desargues's theorem," *Geometry Revisited*, pp. 70–72, 1967.

[17] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics - Principles and Practice*. Reading (MA), USA: Addison-Wesley, 1996.

[18] C. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[19] T. Cootes, G. Edwards, and C. Taylor, "Active Appearance Models," in *Proceedings of the European Conference on Computer Vision ECCV'98*, ser. Lecture Notes in Computer Science 1406, H. Burkhardt and B. Neumann, Eds., vol. 2. Freiburg, Germany: Springer, June 1998, pp. 484–498.