Integrating Object Detection with 3D Tracking Towards a Better Driver Assistance System

Victor Adrian Prisacariu[1], Radu Timofte[2], Karel Zimmermann[2], Ian Reid[1], Luc Van Gool[2]
[1]*Active Vision Laboratory, University of Oxford, United Kingdom*
[2]*ESAT-PSI / IBBT, Katholieke Universiteit Leuven, Belgium*
{*victor, ian*}*@robots.ox.ac.uk*,{*rtimofte, kzimmerm, vangool*}*@esat.kuleuven.be*

Abstract

Driver assistance helps save lives. Accurate 3D pose is required to establish if a traffic sign is relevant to the driver. We propose a real-time system that integrates single view detection with region-based 3D tracking of road signs. The optimal set of candidate detections is found, followed by AdaBoost cascades and SVMs. The 2D detections are then employed in simultaneous 2D segmentation and 3D pose tracking, using the known 3D model of the recognised traffic sign. We demonstrate the abilities of our system by tracking multiple road signs in real world scenarios.

1 Introduction

Traffic signs are designed to help drivers reach their destination safely, by providing them with usefull information. However, when road signs are missed or missunderstood accidents happen. A recent statistic shows that over 98% of car accidents happen because the driver was distracted. By attracting the driver's attention to the traffic signs on the road many accidents would be averted. As a result there has been much work towards a fast and reliable traffic sign detection and tracking system.

Most current work involves combining a detector with a Kalman filter, like in [3, 8], or with a particle filter, like in [5, 4]. These methods rely on a predictable car motion model or reliable feature detectors. For example in [6] and [3] the car is assumed to move in a straight line and with constant velocity. As feature descriptors, trackers usually use edges or some kind of information extracted from the traffic sign shape. In [3] the authors explicitly model geometric properties of the traffic sign shape (i.e. a triangle shaped sign has to be equilateral). This leads to a lack of robustness when subjected to occlusions, deformations or motion blur. In [5] the authors track circular signs and assume these have a coloured border on a white interior and that clear edges can be extracted. This approach would not scale to differently shaped signs and is again vulnerable to motion blur. A solution to some of these problems is region based tracking. Regions are more stable than edges so tracking is more robust, and they are less affected by occlusions or motion blur.

To our knowledge, with notable exceptions like [5] and [10], most of previous road sign work was *2D*. That is the position of the traffic sign was tracked in the *image* rather than in *3D space*. In [5] the authors use inertial sensors mounted on the car to help them obtain an approximation of the 3D pose of the signs, with respect to the car. Unfortunately this approach would fail when the traffic sign does not point towards the car, like in the case shown in Figure 1. Here the no right turn sign does not have the same rotation as the inertial sensor mounted on the car. An alternative method is presented in [10], where multiple views, 3D reconstruction, and an Minimum Description Length based method are used. Although a 3D pose is recovered, processing time is very long - it is an offline method for 3D mobile mapping purposes.

Our approach [1] integrates a single view detection and recognition step with a multiobject, model and region based, 3D tracker. This has several advantages: (i) we are able to obtain the full 3D pose of the traffic signs in the image, accounting for the case in Figure 1, (ii) the tracking is region based, making it robust to motion blur and occlusions, (iii) because our tracker processes only a small region in and around the detection we are able to achieve real time performance.

The remainder of this paper is structured as follows: we begin by presenting an overview of our algorithm in

[1]This work was supported by the Flemish IBBT-URBAN project and by EPSRC through a DTA grant. The authors thank GeoAutomation for providing the images.

Figure 1. Importance of determining the traffic sign orientation. The no right turn sign does not point towards the car.

Section 2. In Section 3 we detail our single view detection and recognition step while in Section 4 we present our 3D tracker. In Section 5 we include the results of applying our system to several images and videos. We conclude in Section 6.

2 Algorithm Overview

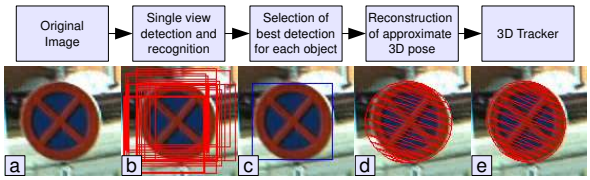

Figure 2. Algorithm overview

An outline of our algorithm is shown in Figure 2. It consists of two phases: first the single view detector is ran on the image and the best detection for each object is selected. Second, the 3D pose at the current frame is predicted, based on the 2D detection and a constant velocity motion model. The 2D detection bounding box is converted to a 3D pose using a 4 point planar pose recovery algorithm. The 3D tracker is then used to refine the 3D pose, for each object in the image. If the detection corresponds to a traffic sign the approximate 3D pose will be its initialisation.

3 Object Detection and Recognition

When a new frame is available we first use the object detection and recognition algorithm of [10]. The single-view detection phase consists of the following steps:

1) Candidate extraction - very fast preprocessing step, where the optimal combination of simple (i.e. computationally cheap), adjustable extraction methods select bounding boxes with possible traffic signs. This step requires an automatic offline learning stage, where an optimal subset of those extraction methods is learnt to yield very few false negatives, while keeping the number of false positives in check.

2) Detection - Extracted candidates are verified further by a binary classifier which filters out remaining background regions. It is based on the Viola and Jones Discrete AdaBoost classifier [11]. Detection is performed by cascades of AdaBoost classifiers, followed by an SVM operating on normalised RGB channels, pyramids of HOGs [2] and AdaBoost-selected Haar-like features.

3) Recognition - A hierarchy of SVM classifiers splits the detections in six basic traffic sign subclasses (triangle-up, triangle-down, circle-blue, circle-red, rectangle and diamond) and then another hierarchy of SVM classifiers (for each subclass) assigns the traffic sign type for the different candidate detections.

At this stage several detections might be available for each object, as shown in Figure 2b. We perform the non-maximum suppression using the energy function of [1, 7] as the colour segmentation score. With b as the bounding box we write:

$$P(b) = \prod_{x \in \Omega} \Big(H_e(a(x)) P_f + \big(1 - H_e(a(x))\big) P_b \Big) \quad (1)$$

Here Ω is the image domain, H_e is the smooth Heaviside function, x is the pixel in the image, a(x) equals 1 inside the bounding box and -1 outside and:

$$P_f = \frac{P(y_i|M_f)}{\eta_f P(y_i|M_f) + \eta_b P(y_i|M_b)} \quad (2)$$

$$P_b = \frac{P(y_i|M_b)}{\eta_f P(y_i|M_f) + \eta_b P(y_i|M_b)} \quad (3)$$

with η_f the number of foreground pixels, η_b the number of background pixels, y_i the colour of the i-th pixel, $P(y_i|M_f)$ the foreground model over pixel values y and $P(y_i|M_b)$ the background model. We use RGB images and our models are histograms with 32 bin for each channel, which are updated online, allowing for variations in illumination. When an object is first detected we chose the detection with the highest SVM score, and initialise $P(y_i|M_f)$ and $P(y_i|M_b)$.

4 3D Initialisation and Tracking

The core of our tracking is the PWP3D algorithm [7]. It assumes a known 3D model, a calibrated camera

and known foreground/background region statistics. A level set embedding function is built from the contour of the projection of the model and the posterior per-pixel foreground/background probability is maximised as a function of pose. The energy function that is minimised is similar to the log of Equation 1:

$$E(\Phi) = -\sum_{x\in\Omega} \log\Big(H_e(\Phi)P_f + \big(1-H_e(\Phi)P_b\big)\Big) \quad (4)$$

with the level set embedding function Φ replacing $a(x)$.

The actual minimisation is done by computing the derivatives of this energy function with respect to the pose parameters and using gradient descent, as in [7]. This does have the disadvantage that convergence is not guaranteed within the permitted number of iterations i.e. more iterations would almost always lead to a better solution. In our testing we noticed that an average of 15 iterations is enough for a good enough result.

The PWP3D tracker needs an initial 3D pose and values for the foreground / background membership probabilities. Also, at each new frame, the 2D detections need to be converted to (approximate) 3D poses to be combined with the 3D tracker. In our case the objects can be approximated with their planar counterparts, which means we can use any one of several planar pose recovery algorithms currently available to convert the 2D bounding box to a 3D pose. We use a current state-of-the-art algorithm introduced in [9]. This algorithm requires (at least) 4 3D-2D point correspondences. We use the 4 corners of the detection bounding box as the 2D points and relate them to the 4 corners of the bounding box enclosing the 3D model of object. An example result is depicted in Figure 2d.

At each new frame, for all previously known objects, the new 3D poses (obtained from the 2D bounding boxes) must be integrated with the tracker. To do this we begin by defining a constant velocity motion model:

$$v^i_{t_k} = t_{k-1} - t_{k-2} \quad v^i_{r_k} = r_{k-1}r^{-1}_{k-2} \quad (5)$$

where k is the current frame, $k-1$ is the previous frame, t is the translation and r is the rotation quaternion from the tracker. The velocity given by the 2D detections is:

$$v^{ii}_{t_k} = u_k - u_{k-1} \quad v^{ii}_{r_k} = p_k p^{-1}_{k-1} \quad (6)$$

where u is the translation and p is the rotation quaternion, obtained from the detector by using the 4 point planar pose recovery algorithm. The predicted pose for the current frame becomes:

$$t_k = t_{k-1} + \alpha v^i_{t_k} + \beta v^{ii}_{t_k} \quad r_k = r_{k-1} q_\alpha v^i_{r_k} q_\beta v^{ii}_{r_k} \quad (7)$$

where k is the current frame, $k-1$ is the previous frame, t is the translation from the tracker and u is the translation from the detector. The variables α, β, q_α and q_β are dependant on the distance between the object and the camera. α and q_α are inverse proportional to this distance while β and q_β are proportional to it. Thus we are able to give more importance to the motion model when the object is closer to the car and vice versa.

Finally the predicted pose (t_k, r_k) is refined using the tracker. The tracker could have been used alone to obtain the pose changes from consecutive frames, but this would have led to longer processing times and would have increased the chance of loosing tracking.

We could have used a Kalman filter for a purely statistical fusion of the tracking data with the approximate poses from the detector. A Kalman filter represents all measurements, system state and noise as multivariate Gaussian distributions. By iterating the tracker rather than doing a purely statistical data fusion we make no pretence on the type of these probability distributions.

5 Results

We tested our algorithm with a multitude of traffic signs types and shapes. In Figure 3 (top) we show our system tracking a pedestrian crossing sign and obtaining a reasonably accurate pose even when the object is far from the camera. It is possible to calculate the distance between the sign and the car, which could then be used to automatically adjust the speed of the car.

In Figure 3 (bottom) we show our system tracking multiple objects. At present the pose for each sign is estimated independently, though of course relative to the camera each sign undergoes the same rigid transformation. We might expect improved performance were we to introduce this coupling.

In Figure 4 we compare the performance of our system, tracking a single sign over a distance of 70m (or 70 frames at 36km/s), with and without the 3D tracker. At higher distances the object becomes only tens of pixels big so detection and tracking are prone to errors. Higher resolution images would allow for the sign to be detected and tracked earlier. The car was moving in a straight line, with constant speed. Translation should therefore change linearly, while rotation should remain constant. Though there is little choice between using the 3D tracker and using just the 4 point planar pose recovery algorithm with regard to translation the tracker must be used to reliably obtain rotation.

A video processed by our system is available at **http://homes.esat.kuleuven.be/˜rtimofte** . It shows our system tracking multiple objects of different types, orientations and colours, with or without motion blur and occlusions. Our CPU implementation of the detection phase runs at around 20fps on 640x480 images

Figure 3. Filmstrip showing 5 frames from a video tracking a pedestrian crossing sign (top) and tracking multiple traffic signs (bottom)

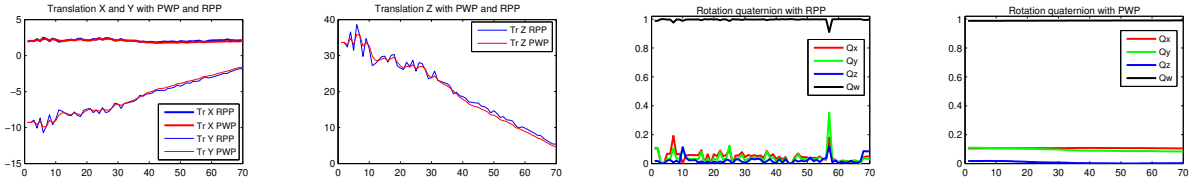

Figure 4. System performance while tracking a sign over 70m, with just the 4 point pose recovery (RPP) and with the tracker (PWP)

while the GPU based tracker needs up to 20ms per object (on a 640x480 image).

6 Conclusions

In this work we proposed a system that can track multiple traffic signs in 3D, from a single view. By integrating accurate detections with 3D region based tracking our system is robust to motion blur and occlusions, while still running in real time. Future work could add person and car 3D detection and tracking to create a complete driver assistance system.

References

[1] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV 2008*, pages 831–844.

[2] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR 2007*, pages 401–408.

[3] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh. Road-sign detection and tracking. *Vehicular Technology, IEEE Transactions on*, 52:1329–1341, 2003.

[4] L. D. Lopez and O. Fuentes. Color-based road sign detection and tracking. In *ICIAR 2007*, pages 1138–1147.

[5] M. Meuter, A. Kummert, and S. Muller-Schneiders. 3d traffic sign tracking using a particle filter. In *ITSC 2008*, pages 168–173.

[6] G. Piccioli, E. De Micheli, P. Parodi, and M. Campani. Robust road sign detection and recognition from image sequences. In *Intelligent Vehicles '94 Symposium*, pages 278–283.

[7] V. Prisacariu and I. Reid. Pwp3d: Real-time segmentation and tracking of 3d objects. In *BMVC 2009*.

[8] A. Ruta, Y. Li, and X. Liu. Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recognition*, 43:416–430, 2010.

[9] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *PAMI*, 28:2024–2030, 2006.

[10] R. Timofte, K. Zimmermann, and L. van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. In *WACV 2009*, pages 69–76.

[11] P. Viola and M. Jones. Robust real-time face detection. In *IJCV 2001*, volume 2, pages 747–757.