Visual Tracking of Buses in a Parking Lot

T. Castanheira, P. Silva, R. Ferreira, A. Bernardino, J. Gaspar
http://www.isr.ist.utl.pt

Institute for Systems and Robotics / IST
Lisbon, Portugal

Abstract

In this paper we propose an automatic system for the visual-tracking of buses in a parking lot, by using a set of Pan-Tilt-Zoom (PTZ) cameras. It is assumed that the parking lot has specific entry points, so that the places from where the buses come in are known beforehand. To detect the buses a background subtraction method is used, being then made an initial estimation of the bus position through backprojection. This estimate is then refined by an algorithm that also provides the bus orientation. The final estimate is then used in a EKF (Extended Kalman Filter) filter to provide a estimation of the bus next position, which allows to decide the camera movement. To test the system a simple simulator was developed using Matlab and Virtual Reality Modelling Language (VRML).

1 Introduction

Task automation has been increasing in a variety of industries and services, due to its capability of improving either efficiency and effectiveness. In this context the work presented in this paper is proposed as a method to automatically track buses moving in a parking lot. The system consists of a set of multiple PTZ cameras placed at certain locations of the parking lot, that begin to track and detect the buses from the moment they enter it.

In [7] a number of motion modalities for one pan-tilt camera are assessed with respect to omni-awareness or, more precisely, maximizing the percentage of events found. Starzyk and Qureshi [6] consider multiple PTZ cameras to track pedestrians (moving events). They design a behavior based architecture which handover tracking inter-cameras and maximize the zoom of each camera while not loosing tracking of all pedestrians. In our work the events to find and track are moving buses. Contrarily to the motion of people, the motion of buses can be predicted. In this work we take the predictions of the motion into account in the design of PTZ control.

2 Camera Model and Bus Detection

The pin-hole camera model [4, 5] is used in this work to represent the relationship between world, M, and image, m, coordinates:

$$m \approx PM = K[R \mid t]\, M \qquad (1)$$

where P is called the projection matrix, K and R the intrinsic and extrinsic parameters matrices respectively and t is the translation vector.

It is assumed that when a bus enters the parking lot a signal is sent to notify the system, which then moves the field of view of an available camera to the adequate entry point. From this time onwards the bus detection is made through a background subtraction algorithm, and a measure of the bus position is generated by using backprojection. Then an optimization algorithm uses a bus 3D model to improve this estimation and to calculate the bus orientation.

2.1 Background Subtraction and Bus Pose Estimation

In order to detect a bus, we start by building a model of the background for each of the PTZ cameras [3]. This background model allows acquiring a background image at any pan-tilt-zoom configuration. The current (real) image is then subtracted to the background image, resulting a logical mask whose pixels indicate the bus pixels.

The center of mass of the detected bus pixels allows estimating coarsely the bus location. Backprojection [4] is applied to the mass center (pixel location) subject to setting the Z coordinate to be in the ground plane. In practice this corresponds to solving a matrix equation, $M = C + \alpha D$, where M is a point in world coordinates (as in equation 1), C the projection center, D a point in infinity and α is a scaling factor.

2.2 Fine Tuning the Pose Estimation

To fine tune the (coarsely) estimated pose it is used a minimization algorithm that finds the local minimum of a cost function of several variables:

$$(X,Y,\theta)^* = \arg_{(X,Y,\theta)} \min F(X,Y,\theta)\,, \quad F(X,Y,\theta) = 1 - \frac{\#(A \cap B)}{\#B} \quad (2)$$

where A and B are both binary masks (see figure 1 for an example). In the case of A it is obtained from the background subtraction algorithm of section 2.1. Relatively to B it is a synthetic mask, generated by placing a 3D model of the bus in an image of the background and then using it in the background subtraction algorithm mentioned in 2.1. Initially the 3D model is placed at the position X, Y of the world referential frame, computed from the backprojection mentioned in 2.1 and with the orientation θ obtained from the EKF (see section 3.2) prediction step. With each iteration of the minimization algorithm, a Levenberg-Marquardt like algorithm, the X, Y and θ values are updated so that the value of the cost function $F(X,Y,\theta)$ approaches zero. When the search ends, the masks A and B will ideally be overlapping perfectly.
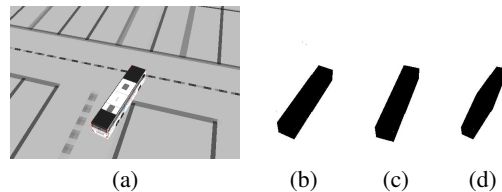

Figure 1: (a) Camera view (b) Mask A (c) Mask B (d) Intersection of A and B

3 Bus Tracking Algorithm

For each bus being tracked there is an associated EKF which is used to predict the bus position in the next sampling time. The measurements correspond to the estimation obtained from the bus detection algorithm mentioned in section 2. The camera pan and tilt angles are then adjusted in order to position the center of view at the EKF estimation. This guarantees that in the next sampling time the bus will be kept visible and close to the image center.

3.1 Bus kinematics model

The bus kinematics model used assumes that: (i) the front wheels can spin around their axis but have no traction, (ii) the back wheels have traction but do not spin or slide, (iii) the bus body and wheels are assumed to be rigid bodies [1, 2]. From these assumptions and the geometric relations shown in figure 2(a) it was then generated the following discrete kinematics model, that is used by the Extended Kalman filter in the prediction step:

$$\begin{cases} x_{k+1} = x_k + T \cdot v_k \cos(\theta_k + \Phi_k + \frac{T}{2} \cdot \frac{v_k}{L} \sin \Phi_k) \\ y_{k+1} = y_k + T \cdot v_k \sin(\theta_k + \Phi_k + \frac{T}{2} \cdot \frac{v_k}{L} \sin \Phi_k) \\ \theta_{k+1} = \theta_k + T \cdot \frac{v_k}{L} \sin \Phi_k \\ v_{k+1} = v_k \\ \Phi_{k+1} = \Phi_k \end{cases} \quad (3)$$

The state variables are the position x, y of the bus front axel shaft center(x_f and y_f coordinates of figure 3) and the bus body orientation angle θ. The model inputs are the bus linear velocity v and the front wheels angle Φ relatively to θ, both assumed to be constant. In what concerns the constants T and L they are the sampling interval and the front wheel distance respectively.

3.2 Extended Kalman Filter

The Kalman filter is a linear quadratic estimation method to compute estimates of unknown variables, from measurements corrupted with zero
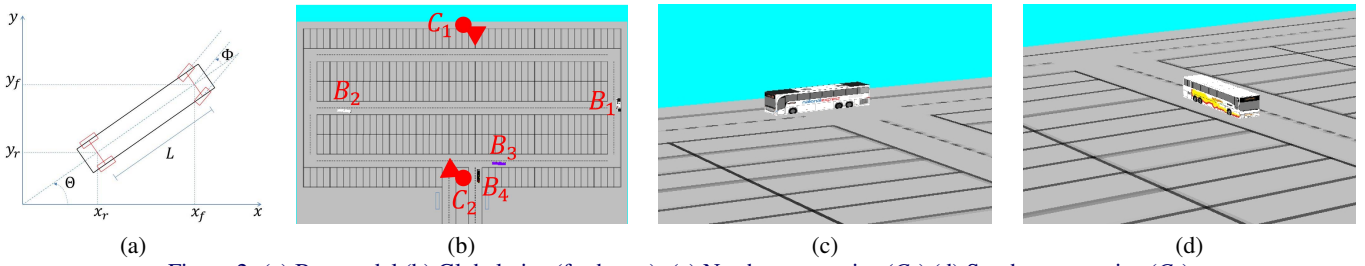
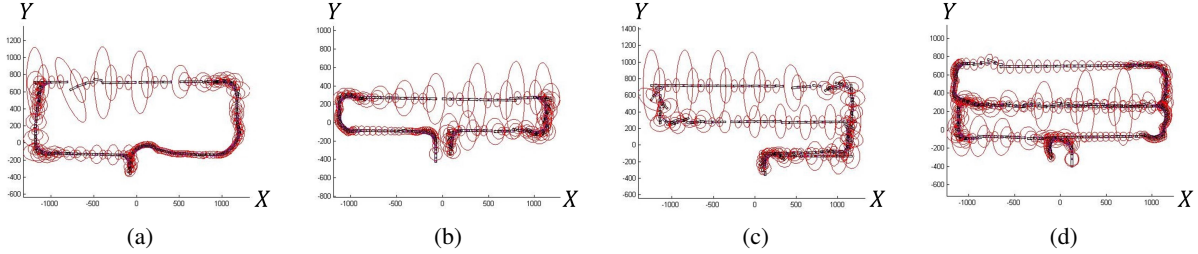Figure 2: (a) Bus model (b) Global view(for buses). (c) North camera view(C_1) (d) South camera view(C_2)

Figure 3: Trajectories estimated for each bus and associated uncertainty ellipses: (a) B_1,(b) B_2,(c) B_3,(d) B_4

mean Gaussian noise. As the original Kalman filter could only be used on linear systems the Extended Kalman filter was used instead. It assumes a system of the form:

$$\begin{cases} X_{k+1} = f(X_k, \xi_k) \\ z_k = h(X_{k+1}, \eta_k) \end{cases} \quad (4)$$

The variables X_k, z_k are the state variables and the measurements respectively. In what concerns ξ_k and η_k they are zero mean multivariate system and observation Gaussian noises. The functions f and h correspond to non-linear functions which are linearized. Then the Extended Kalman filter equations are applied to obtain the *a priori* and the *a posteriori* estimations.

4 Complete System

The complete system takes into account three main components: buses, cameras, and the 3D world. Buses have autonomous motions (are driven by on board drivers). The PTZ cameras are mounted at known positions and orientations of the 3D world. PTZ cameras are controlled automatically to detect and track the buses. The 3D world, the parking-lot floor, is described as a plane.

At every second of time elapsed, the following tasks are run: (i) Predict the buses position and orientation with the EKF and compute corresponding uncertainty ellipses. (ii) Assign a bus to a camera taking into consideration a FIFO multitasking algorithm which takes into account the uncertainty ellipses' areas and distances of the cameras to the buses in *pan*, *tilt* and *zoom* units. (iii) Move each camera to image the coordinate predicted for its assigned bus (X, Y). (iv) For every image acquired by the cameras, observe the buses position using the algorithm described in section 2. (vi) Update the EKF filers using the observations of the buses.

5 Experimental Results

In order to test the complete system a 3D simulator built was developed in MATLAB using VRML. The buses have predefined trajectories. The PTZ cameras are controlled automatically.

The simulated system encompasses two cameras (C_1 and C_2) and four buses (B_1, B_2, B_3 and B_4). Figure 2(b) shows an aerial view of the parking lot at a simulation iteration where the four buses were all in the scene. The center of the world referential frame is at the center of the floor plane of figure 2(b). The cameras were positioned at the red dots location and oriented in the direction of the "red triangles" base associated to each red dot. Typical images acquired by the camera are shown in figures 2(c) and 2(d).

At the end of the experiment the plots presented in figure 3 were obtained. In these graphs the rectangles position and orientation represent the estimates obtained from the EKF update step for each bus, at a given simulation moment. The elliptical lines are the limits of the uncertainty area associated with the EKF estimation. As can be noted in these graphs

the program was able to keep track of the positions of the buses during all the simulation. There were however certain positions where the estimation was not accurate. However the system was always able to recover from the errors and continue tracking the buses.

6 Conclusion and Future Work

In this paper an algorithm to detect and track multiple buses was developed as well as a simulator to test it. The approach used showed encouraging simulation results.

There are however some improvements that need to be applied to make this project feasible in reality. For example the background subtraction algorithm has to be improved so that it can handle the lighting changes that would occur in the real world. Another example, a camera controller needs to be developed to deal with the dynamics of the motors moving the camera.

The simulator itself also has much space for improvement. As an example, a number of light changes that occur in real world can be simulated by an artificial sun moving according to the time of the day.

References

[1] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer, 2011.

[2] Mario Filipe Florêncio and Pedro Emanuel Agostinho. *Parqueamento Automóvel Automático*. Instituto Superior Técnico, 2005.

[3] Ricardo Galego, Alexandre Bernardino, and José Gaspar. Vignetting correction for pan-tilt surveillance cameras. In *VISAPP 2011. International Conference on Computer Vision Theory and Applications*, 2011.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[5] E. Hayman L. de Agapito and I. Reid. Self-calibration of a rotating camera with varying intrinsic parameters. *Proc 9th British Machine Vision Conference, Southampton*, pages 883–893, 1998.

[6] W. Starzyk. Multi-tasking smart cameras for intelligent video surveillance systems. *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011.

[7] D. A. P. Vicente. Event detection with pan-tilt cameras. *Dissertação para a obtenção do Grau de Mestre em Engenharia Electrotécnica e de Computadores*, 2009.