3D TRACKING OF LICENSE PLATE FROM MONOCULAR CAMERA VIEW

İlhan Kubilay Yalçın[1], *Muhittin Gökmen*[2]

[1]TUBITAK MRC
Information Technologies Institute Kocaeli, Turkey
ilhan.yalcin@bte.mam.gov.tr
[2]ITU Computer Engineering Department
Istanbul Technical University Istanbul, Turkey
gokmen@cs.itu.edu.tr

ABSTRACT

In this paper, we present a method for tracking a planar object (license plate) in 3D. Given an initial estimate, we try to estimate the location, motion vector and pose of the object in 3D for the successive video frames. We utilize condensation algorithm for estimating the state of the object and filtering the measurements according to the extracted image features. Image features, namely the directional gradients on object boundaries, are calculated locally for each guess of the condensation algorithm. Local extraction of the features decreases the computational complexity of the tracking process. The algorithm tries to fit the state of the object to the image gradients. In contrast with optimization-based approaches, condensation algorithm simplifies the complexity of the system. The parameters of the algorithm are easy to understand and easy to apply even for complex structures. We demonstrate the results of the method for tracking a license plate for different scenarios. The results are promising for further research on the method.

1. INTRODUCTION

Harris et. al. [1] introduce RAPID object tracker. The RAPID tracker represents a 3D object as a set of control points, which are high contrast edges. The pose of the object is estimated and the object boundaries are tracked. The algorithm can be divided into two parts, the first dealing with making measurements in the image, and the second calculating the new pose of the object. The pose of the object is tracked over time using a Kalman filter. A constant velocity model is assumed for the object, so that there are 12 state variables (6 for pose and 6 for pose velocity) and 6 measurement variables (pose only).

Kollnig and Nagel [2] match a synthetic gradient image directly to the gray level gradient image from the video frames. The difference between the synthetic gradient image and gray value gradient of current frame is used to update the 3D pose of the model using a MAP estimator. A Kalman filter stabilizes the tracking. The edges are models with 2D Gaussians and MAP estimator utilizes Gauss-Newton method modified with Levenberg-Marquart iteration.

N. Giordana et. al. [3] introduce a 2D model based tracking. The tracking is performed in several steps. The control points on the contour are determined from the 3D model. These points form a polyhedral shape, which is assumed to correctly model the object appearance in the image. Perspective projection is applied to the model. An affine transformation is defined and the parameters are estimated. The estimation is done by the minimization of a Bayesian criterion, which is composed of two terms, one for the minimization of the differences from the model gradients and the frame gradients, one for the deformations from the model shape. For gradient matching a gradient optimization algorithm is used. For overall minimization simulated annealing is utilized.

Eric Marchand et. al. extend the work in [3], with additional paradigms [4]. The future point matching is accomplished by ME method which calculates the gradients in the direction of a line between two future points of the 3D shape model. This reduces the computational effort for the minimization step. Again the minimization step consists of two steps for affine transformation and the 3D model gradient match. For optimization an explicit discrete search algorithm is applied.

The work in [1, 2, 3, 4] relies on the matched future points, which are calculated from the 3D model. However, matching the future points can yield mismatches, which will degrade the performance of 3D pose optimization step.

In [5] condensation algorithm is applied for 3D face and eye gaze tracking. They consider the problem as nonlinear state estimation and apply a special version of Kalman filter with branching particle propagation. They consider the approach as successful but do not give a measure of computational complexity. They employ about 200 particles.

2. SYSTEM MODEL

Considering the motion of a vehicle license plate in real world, it is trivial that the motion is in three-dimensional space with many alignment possibilities. Defining a 3D location in Cartesian coordinates, we need three dimensions and for determining the alignment of a 3D object in space, we will need three angles such as

$$[\; x \quad y \quad z \quad \alpha \quad \beta \quad \gamma \;]^T \tag{1}$$

where, α is the rotation around x axis, β is the rotation around y axis and finally γ is the rotation around z axis. Beside the location and alignment of a 3D object, we need to consider its motion in 3D space, for which we will need to define a magnitude and two angles in spherical coordinates such as

$$[\; r \quad \phi \quad \theta \;]^T \tag{2}$$

where, r is the velocity, θ is the azimuthal angle in xy-plane from x-axis, ϕ is the polar angle from z-axis. So the total state vector of our system will have nine variables. On the

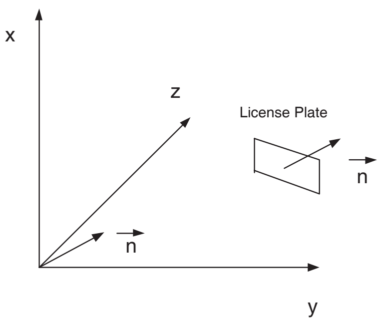

Figure 1: License plate motion model

other hand, for a planar object, which can move in the direction of its normal only as shown in Fig 1, the state vector can be represented with only seven variables because two variables for pose and motion combine. The state vector can be shown as.

$$[\ x \quad y \quad z \quad r \quad \alpha \quad \beta \quad \gamma\ ]^T \tag{3}$$

The system dynamic equation is non-linear such as

$$x_{k+1} = f(x_k) + w_k \tag{4}$$

which defines the motion in 3D. The measurement equation is also nonlinear such as,

$$y_k = h(x_k) + v_k \tag{5}$$

where w_k and v_k are Gaussian process and measurement noise. The function $f(x_k)$ contains trigonometric functions, while the function $h(x_k)$ is defined by the camera perspective transformation. The system is modeled with first order motion equation, which assumes constant velocity, spherical motion with constant $[\ \alpha \quad \beta \quad \gamma\ ]^T$ angles. In contrast with the work in [1,2,3,4], we utilize Condensation algorithm for estimating the state of the system. This work is a novel usage of Condensation algorithm for such a tracking problem.

3. PERSPECTIVE CAMERA MODEL

The camera model defines the $h(x_k)$ function. The model contains two types of parameters. The extrinsic parameters are the parameters that define the location and orientation of the camera reference frame with respect to a known world reference frame. The intrinsic parameters are the parameters necessary to link the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame. When both intrinsic and extrinsic camera parameters are known, the full camera projection matrix M is determined. The projection is than just a matrix multiplication defined as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = M. \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{6}$$

By this projection the real world coordinates $X \quad Y \quad Z$ are transformed into image coordinates $x \quad y \quad z$ and the pixel coordinates on the image p_x, p_y can be calculated by $p_x = x/z$ and $p_y = y/z$.

We utilize full perspective model, where the camera and space coordinates overlap and the 3D origin is the origin of the image plane. The intrinsic parameters are determined by the camera vendor. Depending on the utilized camera lenses, a distortion model can be necessary. Especially radial distortion has significant effects if wide angle lenses are used. We have not utilized any distortion model in our experiments.

4. CONDENSATION ALGORITHM

The standard Condensation algorithm can be summarized as follows:

1. Initialization: t = 0
   For n = 1...N generate samples from the prior in order to obtain $\{s_0^n, \pi_0^n\}$
   where s' are the samples and $\pi' s$ are the weights assigned to each sample.
2. Iterate for t=0,1,2,...
   At time step t+1 , construct the n^{th} of N samples as follows:
   (a) Propagate the samples using state transition equation to obtain $p(x_{t+1}|y_{1:t})$.
   From the sample set at time t , where the samples correspond to the location, pose and motion of the license plate:

   $$\{s_t^n, \pi_t^n\}, n = 1, 2, ..., N$$

   The new sample set $\{s_{t+1}^n, \pi_{t+1}^n\}$ is composed according to the equations:

   $$s_t^n = [\ x_t \quad y_t \quad z_t \quad r_t \quad \alpha_t \quad \beta_t \quad \gamma_t\ ] \tag{7}$$

   $$s_{t+1}^n = s_t^n + \begin{pmatrix} x_t + r_t.sin\beta_t \\ y_t - r_t.sin\alpha_t.cos\beta_t \\ z_t + r_t.cos\alpha_t.cos\gamma_t \\ r_t \\ \alpha_t \\ \beta_t \\ \gamma_t \end{pmatrix} + N(0,\sigma) \tag{8}$$

   (b) Calculate the new weights by:

   $$\pi_{t+1}^n = \pi_t^n.p(y_{t+1}^n|s_{t+1}^n)$$

   (c) Store samples $\{s_{t+1}^n, \pi_{t+1}^n, c_{t+1}^n\}$ where c_{t+1}^n are the cumulative probabilities given by:

   $$c_{t+1}^0 = 0$$

   $$c_{t+1}^n = c_{t+1}^{n-1} + \pi_{t+1}^n$$

   (d) Normalize by dividing all cumulative probabilities $c_{t+1}^n = c_{t+1}^n / c_{t+1}^N$, i.e. so that $c_{t+1}^N = 1$ and weights $\pi_{t+1}^n = \pi_{t+1}^n / c_{t+1}^N$, so that $\sum_n \pi_{t+1}^n = 1$.
   (e) Resample the samples s_{t+1}^n with probability π_{t+1}^n to obtain N samples. For this purpose, generate a random number $r \in [0,1]$, uniformly distributed. Find the smallest n for which $c_{t+1}^n \geq r$. Add this sample to the new set $\{s_{t+1}^m, \pi_{t+1}^m, c_{t+1}^m\}, m = 1, 2, ..., N$

Our approach has a probabilistic framework, where the prior and posterior probability distributions are represented by

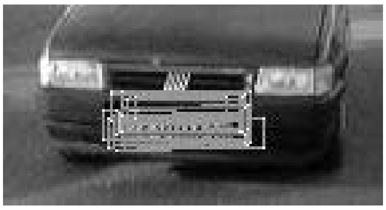

Figure 2: Samples

Figure 3: Directed gradients along the object boundaries

samples. Each sample is a vector, composed of seven variables for the location, alignment and the velocity of the object in 3D as explained in Section 2. Each sample can be considered as a guess for the license plate location, alignment and motion. The samples are projected on to the video frame in Fig 2. The samples and the associated probabilities approximate the prior and the posterior probability distributions. The probability, associated to each sample is calculated by the likelihood function.

We use the directed image gradients for determining the probabilities of each sample $\{\pi_t^n, s_t^n\}$ in Condensation algorithm. The directed gradients are calculated by taking the difference vertical to the proposed license plate borders as shown in Fig 3.

5. RESULTS

The algorithm is robust to wrong initialization; according to the initialization covariance of the particles, the algorithm can tolerate wrong initialization. There is a trade off in defining the initial covariance. If it is defined too large, the algorithm can converge to a wrong location, which is not the license plate. In Figure 4 the first sequence shows an example. The initial location is refined after ten frames. Here the number of particles is 128.

The algorithm can track severe maneuvers as shown in Figure 5. The sequence 2 shows an example. For this example the number of particles needed to be increased to 2048.

The algorithm can converge in static images; the algorithm can converge in static images. In Figure 6, the algorithm is applied on static image. The initial state is slightly wrong. In the proceeding iterations the algorithm can converge to the true boundaries.

Sequence 1 time: 0s

Sequence 1 time: 0.2s

Sequence 1 time: 1s

Figure 4: Sequence 1

REFERENCES

[1] C. J. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, Active Vision. MIT Press, Cambridge, MA, 1992.

[2] H. Kollnig H.H. Nagel 3D Pose Estimation by Fitting Image Gradients Directly to Polyhedral Models Intelligent Vehicles Symposium Proceedings of 1995

[3] N. Giordana, P. Bouthemy, F. Chaumette, F. Spindler, J.-C. Bordas, and V. Just. 2d model-based tracking of complex shapes for visual servoing tasks. In G. Hager and M. Vincze, editors, IEEE Workshop on Robust Vision for Vision-Based control of Motion, Leuven, Belgium, May 1998.

[4] Eric Marchand , Patrick Bouthemy , Franc¸ois Chaumette , Val´erie Moreau IEEE Robust real-time visual tracking using a 2D-3D model-based approach International Conference on Computer Vision, ICCV99, Volume 1, Pages 262268, Kerkira, Greece, September 1999

[5] H. Moon, R. Chellappa, A. Rosenfeld, 3D Tracking Using Shape-Encoded Particle Propagation, ICCV 2001 proceedings of 8th IEEE Int. Conference Vol 2

Sequence 2 time: 0s

Sequence 2 time: 1.5s

Sequence 2 time: 3.1s

Figure 5: Sequence 2

Sequence 3 time: 2s

Sequence 3 time: 2.2s

Sequence 3 time: 2.9s

Figure 6: Sequence 3