

FREE-FORM PLANAR CURVE TRACKING USING RELATED POINTS

Burak Yöndem, Mustafa Ünel, Aytül Erçil

Sabancı University, Faculty of Engineering and Natural Sciences
Istanbul-Turkey

burakyondem@su.sabanciuniv.edu, munel@sabanciuniv.edu, aytulercil@sabanciuniv.edu

ABSTRACT

Tracking free form objects by fitting curve models to their boundaries in real-time is not feasible due to the computational burden of fitting algorithms. In this paper, we propose to do fitting only for certain frames in an image sequence and fill in the missing ones using Kalman filtering technique. An algorithm is presented to track a free-form shaped object, moving along an unknown trajectory, within the camera's field of view (FOV). A discrete steady-state Kalman filter is used to estimate the future positions and orientation of the target object. Kalman filter uses the "related points" extracted from the decomposition of implicit polynomials of target's boundary curves and measured position of target's centroid. Related points undergo the same motion with the curve, hence could be used to estimate the orientation of the target. The resulting algorithm is verified with simulations.

1. INTRODUCTION

Implicit algebraic curves have proven very useful in many model-based applications in the past two decades. Implicit models have been widely used for important computer vision tasks such as single computation pose estimation, shape tracking, 3D surface estimation and indexing into large pictorial databases [1-6].

Tracking techniques are based on matching tokens from the image. They are extracted along the sequence and are used as measurements for the tracking algorithm. There are several tracking approaches in the literature. Most of them can be divided into four groups:

1. *3D based methods*: They use precise geometrical representation of known objects. This type of methods presents a considerable computational load that can not be justified by a real-time system most of the time. However, they have been applied for tracking individual vehicles in traffic scenes by using expensive hardware .
2. *Feature-based methods*: track individual tokens such as points, lines or curves . These methods present two main disadvantages [8]: they do not provide explicit grouping of tokens moving with coherent motion and are quite sensitive to occlusion.

3. *Deformable model-based methods*: fit models to the contours of the moving objects of the scene [9]. They exhibit initialization problems[8]. When moving objects are partially occluded in the scene, initialization fails, since models can not be adapted to the real objects.
4. *Region-based methods*: define groups of connected pixels that are detected as belonging to a single object that is moving with a different motion from its neighbouring regions [10]. Region tracking is less sensitive to occlusion due to the extensive information that regions supply. Characteristics such as size, shape, or intensity can directly be obtained from them.

In this paper, we are interested in tracking a free-form object whose boundary can be described by a planar algebraic curve. We will only consider rigid motion of the object along an unknown trajectory. We will use a unique decomposition [2,7] of algebraic curves to obtain feature points for position and orientation tracking. Decomposition represents such curves as a unique sum of products of (possibly) complex lines. The real intersection points of these lines are so called "related-points", which map to one another under affine transformations.

2. PLANAR ALGEBRAIC CURVES

2D curves can be modelled by implicit algebraic equations of the form, where $f_n(x, y)$ is a polynomial in the variables x, y , i.e. $f_n(x, y) = \sum_{ij} a_{ij} x^i y^j$ where $0 \leq i + j \leq n$ (n is finite) and the coefficients a_{ij} are real numbers [1]. Algebraic curves of degree 1, 2, 3, 4... are called lines, conics, cubics, quartics...etc. Figure 1 shows some objects with their outlines modelled by a 3L curve fitting procedure detailed in [11].

In the sequel, we will focus on the tracking of quartics and note that results can easily be extended to higher degree algebraic curves.



Figure 1 A group of 2D objects and their free-form 3L curve models.

2.1 Decomposed Quartics and Related Points

It has been shown in [2,3] that algebraic curves can be decomposed as a unique sum of line factors, the intersection of which are examples of *related-points*. Considering an accordingly decomposed monic quartic curve:

$$f_4(x, y) = \prod_4(x, y) + \gamma_2 \prod_2(x, y) + \gamma_0 \underbrace{\prod_0(x, y)}_{=1}$$

$$= \prod_{i=1}^4 \underbrace{\begin{bmatrix} 1 & l_{4i} & k_{4i} \\ x & y & 1 \end{bmatrix}}_{=L_{4i}^T} + \gamma_2 \prod_{i=1}^2 \underbrace{\begin{bmatrix} 1 & l_{2i} & k_{2i} \\ x & y & 1 \end{bmatrix}}_{=L_{2i}^T} + \gamma_0 = 0$$

where $L_{mi}^T = \begin{bmatrix} 1 & l_{mi} & k_{mi} \end{bmatrix}$ and $X^T = \begin{bmatrix} x & y & 1 \end{bmatrix}$.

The intersection point $d_p = \{x_p, y_p\}$ of any two non-parallel line factors, such as $L_{ij}^T = x + l_{ij}y + k_{ij}$ and $L_{qr}^T = x + l_{qr}y + k_{qr}$ can be defined by the matrix/vector relation:

$$\begin{bmatrix} 1 & l_{ij} & k_{ij} \\ 1 & l_{qr} & k_{qr} \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} l_{ij}k_{qr} - l_{qr}k_{ij} \\ k_{ij} - k_{qr} \end{bmatrix} \div (l_{qr} - l_{ij})$$

In the case of closed-bounded quartics, we have two pairs of complex-conjugate lines, i.e. $L_{42} = L_{41}^*$, $L_{44} = L_{43}^*$, the intersection points of which are real. For tracking, we will be using the centroid of the bounding curve and these two related points. For the robust calculation of the orientation of the free-form curve, we follow [13] and form two vectors originating from the center of mass to the two related points. The sum of these two vectors is a new vector that is quite robust against noise throughout the whole trajectory. The angle between this sum vector and the positive x-axis is defined to be the orientation of the curve.

3. KALMAN FILTERING

Kalman filters are recursive filters which provide an unbiased, minimum-variance and consistent estimate \hat{x}_k of a state vector x_k . In this section the index k represents the discrete time. Kalman filtering consists of a three-steps strategy named prediction, measurement and update. The prediction computes a first estimate of the state vector $\hat{x}_{k+1}(-)$ and of the covariance matrix defined as $P_k = E[\tilde{x}_k \tilde{x}_k^T]$, where $\tilde{x} = x_k - \hat{x}_k$ and $E[.]$ is the average operator.

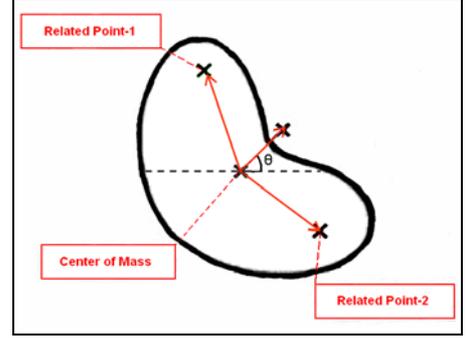


Figure 2. Tracked object with the center of mass, two related points and the corresponding point used for orientation.

$\hat{x}_k(-)$ denotes the prediction vector before measurement and $\hat{x}_k(+)$ refers to the updated vector after the measurement. Prediction equations are based on previous realizations of the updated vector $\hat{x}_k(+)$ and the updated matrix $P_k(+)$:

$$\hat{x}_{k+1}(-) = f(\hat{x}_k(+)) + w'_k$$

$$P_{k+1}(-) = P_k(+) + Q_k$$

where Q_k is the covariance matrix of the model noise w'_k . $Q_k = E[w'_k w_k'^T]$. Q_k reflects the adequacy of the model to describe the underlying physical system. The measurement step consists of the computation, through image processing routines, of visual features named the measurements: z_k . Measurements are related to the state vector through the observation equation:

$$z_k = H\hat{x}_k + v_k$$

where H is the observation matrix and v_k is a measurement error, modelled as an uncorrelated noise. The final update step modifies the state vector according to the measurement z_k , thus providing an updated estimate $\hat{x}_k(+)$. The equations describing the update step modify the state vector and the covariance through the following equations:

$$K_k = P_k(-) \cdot H_k^T \cdot [H_k \cdot P_k(-) \cdot H_k^T + R_k]^{-1}$$

$$P_k(+) = [I - K_k \cdot H_k] P_k(-)$$

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k \cdot [z_k - (H_k \cdot \hat{x}_k(-))]$$

R_k represents the covariance matrix of the measurement noise $v_k : R_k = E[v_k \cdot v_k^T]$. The matrix K_k is called the Kalman gain and has the role of modulating the update of the state vector $\hat{x}_k(-)$ into $\hat{x}_k(+)$ by appropriately weighting the measurement error v_k .

3.1 Target Model

In order to create a Kalman filter, an appropriate linear model of the target must be created. The model must describe the x and y coordinates of the target centroid and the orientation of the target. All three parameters are independent of each other. The x and y models are the same and based on Newton's second law. The orientation is based on a moment equation.

The state space representation of the model for the x and y coordinates in discrete time takes the form:

$$\begin{bmatrix} x_{k+1} \\ v_{k+1} \\ a_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \\ a_k \end{bmatrix} + \bar{w}_k$$

where T is the sampling period of the system which we have chosen to be 5 frames and w_k is the disturbance applied to the object.

The state space representation of the model for the orientation is taken as:

$$\begin{bmatrix} \theta_{k+1} \\ \omega_{k+1} \\ \alpha_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_k \\ \omega_k \\ \alpha_k \end{bmatrix} + \bar{w}_k$$

Clearly, the state matrix and input vector are identical to the translational models. Therefore it is sufficient to use one model for all three parameters. Peter Corke[12] has used the Kalman filter as solution to visual servoing problem. He used the filter to have an end-effector track an object, using a velocity based control scheme. Corke used the recursive form of the filter and his target model was second order. In this work, we used the steady-state form of the Kalman filter in order to estimate the position and orientation of the object between measurement frames. Since the system is at steady-state a single equation is used to determine the filter.

$$x_{k+1} = \underbrace{[A - KH]}_{\bar{A}} x_k + Kz_k = \bar{A}x_k + Kz_k$$

Recall that for the filter, the measurement is constant until the feature extractor's next time step. So,

$$z_k = z_{k+1} = z_{k+2} = \dots = z_{k+n-1}$$

Hence,

$$x_{k+n} = \bar{A}^n x_k + \left[\sum_{i=0}^{n-1} \bar{A}^i \right] Kz_k$$

The feature extraction algorithm, once every sample period, sends the measured values, which describe the target position and orientation to the filter. The filter will hold this value and use it as a measurement until it is updated by the feature extraction algorithm. The process of holding the measurement value has the effect of creating another input trajectory that operates on a higher frequency. The filter uses the new input measurement function to determine the incremental estimates of the object.

4. EXPERIMENTAL RESULTS

For our experiments we used a boomerang shaped object undergoing a rigid motion along a relatively complex trajectory. Object boundaries have been modelled by quartic curves. The related points of these curves are obtained from the decomposition of the curve.

A clear illustration of the filter's tracking performance can be seen from error graphs. The objective is to track the measured signal, so it is assumed that the measure is the true coordinate position. So, the error is the difference between the prediction and measured value. The error values are low and within a band of ± 3 pixels when the target performs relatively uniform motion. Figure 4 illustrates the x coordinate tracking performance. When the target makes a manoeuvre, error values shows rapid increases, however the values converge to normal error values when the manoeuvre finishes. The y-coordinate, on the other hand was exposed to higher speeds and sharp manoeuvres. Figure 5 illustrates the y coordinate tracking performance.

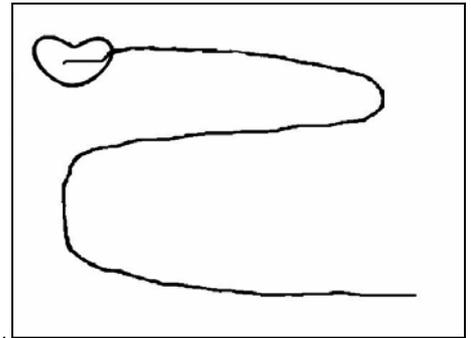


Figure 3. An example trajectory of the target

Purposely, the trajectory under consideration implies large angle variations to test robustness of the proposed orientation estimation method. As can be seen from Figure 6, orientation tracking error is within reasonable bounds.

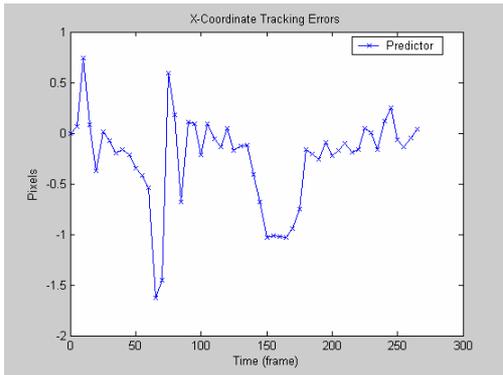


Figure 4 X-Coordinate Tracking Error of the Filter

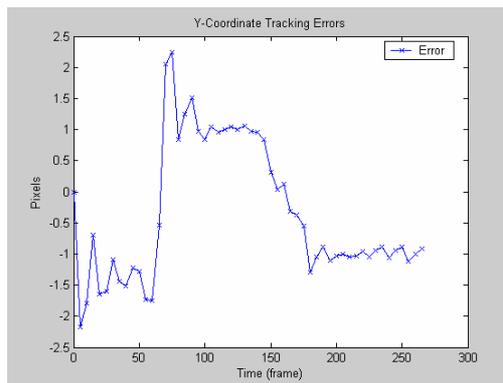


Figure 5 Y-Coordinate Tracking Error of the Filter

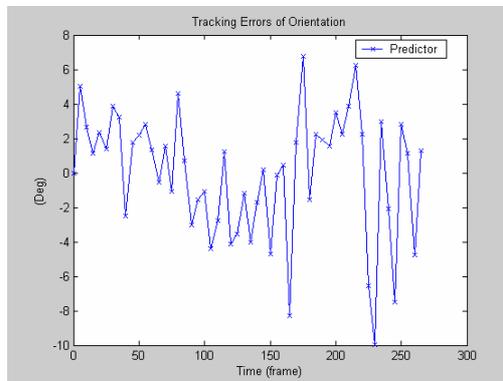


Figure 6 Orientation Tracking Error of the Filter

5. SUMMARY AND CONCLUSION

We have presented a method for tracking the position and the orientation of 2D free-form objects undergoing rigid motion. By using the fact that the related points undergo the same motion with the curve, we have employed a robust orientation measure for the curve. Tracking approach was aiming to reduce the number of computations and was quite successful.

REFERENCES

- [1] C.G. Gibson, *Elementary geometry of algebraic curves*, Cambridge University Press, Cambridge, UK, 1998.
- [2] M Unel, W. A. Wolovich, "On the construction of complete sets of geometric invariants for algebraic curves," *Advances in Applied Mathematics* Vol. 24, No. 1, pp. 65-87, January 2000.
- [3] M. Unel, W. A. Wolovich, "A new representation for quartic curves and complete sets of geometric invariants," *International Journal of Pattern Recognition and Artificial Intelligence*, December 1999.
- [4] G. Taubin, D. B. Cooper, "2D and 3D object recognition and positioning with algebraic invariants and covariants," Chapter 6 of *Symbolic and Numerical Computation for Artificial Intelligence*, Academic Press, 1992.
- [5] D. Keren et al., "Fitting curves and surfaces to data using constrained implicit polynomials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 1, January 1999.
- [6] H. Civi, C. Christopher, A. Ercil, "The Classical Theory of Invariants and Object Recognition Using Algebraic Curve and Surfaces," *Journal of Mathematical Imaging and Vision* 19: 237–253, 2003.
- [7] W. A. Wolovich, Mustafa Unel, "The determination of implicit polynomial canonical curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 10, pp. 1080-1089, October 1998.
- [8] A. Mitiche and P. Bouthemy, "Computation and analysis of image motion: a synopsis of current problems and methods", *International Journal of Computer Vision*, 19(1):29-55, 1996
- [9] A. Blake, R. Curwen and A. Zisserman, "A framework for spatio-temporal control in the tracking of visual contours", *International Journal of Computer Vision*, 11(2): 127-145, 1993.
- [10] D.P. Kottke and Y. Sun, "Motion estimation via cluster matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1128-1132, 1994
- [11] Z. Lei, M. M. Blane and D. B. Cooper, "3L Fitting of Higher Degree Implicit Polynomials," In proceedings of 3rd IEEE Workshop on Applications of Computer Vision, pp. 148-153, Florida 1996
- [12] Peter I. Corke and Malcolm C. Good, "Dynamic effects in high performance visual servoing", *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1838-1843, 1992
- [13] Mustafa Unel. "Polynomial Decompositions for Shape Modeling, Object Recognition and Alignment" PhD Thesis, Brown University, Providence, RI 02912, May 1999.