Object Tracking in a Video Sequence

CS 229 Final Project Report
Young Min Kim
ymkim@stanford.edu

Abstract

Object tracking has been a hot topic in the area of computer vision. A lot of research has been undergoing ranging from applications to noble algorithms. However, most works are focused on a specific application, such as tracking human, car, or pre-learned objects. In this project, objects randomly chosen by a user are tracked using SIFT features and a Kalman filter. After sufficient information about the objects are accumulated, we can exploit the learning to successfully track objects even when the objects come into the view after it had been disappeared for a few frames.

Key Words- Object tracking, SIFT, Kalman filter

1. Introduction

Object tracking is useful in a wide range of applications: surveillance cameras, vehicle navigation, perceptual user interface, and augmented reality [1]. However, most of the research on tracking an object outperforms using selective algorithms that are applicable for fixed settings.

The focus of this project is tracking a general object selected in a real time. The object to be tracked in a frame is chosen by a user. Scale Invariant Feature Transform (SIFT) features [2], point features that are highly distinguishable for an object, are used as a reliable feature to track with lack of initial training data. The motion of a selected object is learned assuming a Gaussian model by Kalman filter [3][5]. While tracking the object, more features are accumulated and the prediction made by Kalman filter becomes more reliable as more frames are passed.

The rest of paper is organized as follow: Section 2 presents the theoretical background about SIFT features and Kalman filter, the two most important ideas used in the tracking algorithm. The tracking algorithm is explained in Section 3 including the usage of SIFT features and Kalman filter in detail. Section 4 concludes the paper with possible future extensions of the project.

2. Background

2.1. SIFT Features

SIFT [2] is an efficient way to find distinctive local features that are invariant to rotation, scale, and possible occlusion. To find SIFT features, you produce images in different scales. Each image is convolved with a Gaussian kernel, and the differences between adjacent scales of convolved images are calculated. Candidate keypoints are local maxima and minima of the difference. From the candidates, keypoints are selected based on measures of their stability. One or more orientations are assigned to each keypoint location based on local image gradient directions. The gradients at the selected scale in the region will represent the keypoints. The full description on calculating SIFT points and usage of them for matching images can be found at [2].

Since we do not have any prior knowledge of the objects, point features are used to represent and detect an object rather than texture, color or structure.

2.2. Kalman Filter

Kalman filter assumes Gaussian distribution of states and noise. Suppose x is the state, z is the measurement, w is process noise, v is measurement noise, and they are all Gaussian. The noises w and v are independent to states and measurements. Then we have [3][4]

$$x_{k+1} = Ax_k + w_k$$
$$z_k = Hx_k + v_k$$
$$w_k \sim N(0, Q_k)$$
$$v_k \sim N(0, R_k)$$
$$\overline{P_k} = E[\overline{e_k}\overline{e_k}^T], \overline{e_k} = x_k - \overline{X_k}$$
$$P_k = E[e_k e_k^T], e_k = x_k - X_k$$

where P denotes the error covariance.

Then, the Kalman filter estimates the state x of time k+1 and correct the prediction using measurement z of that time using the following equations,

Time update (prediction):

$$\overline{X_{k+1}} = A_k X_k,$$
$$\overline{P_{k+1}} = A_k P_k A_k^T + Q_k.$$

Measurement update (correction):

$$K_k = \overline{P_k} H_k^T (H_k \overline{P_k} H_k^T + R_k)^{-1}$$
$$X_k = \overline{X_k} + K_k (z_k - H_k \overline{X_k})$$
$$P_k = (I - K_k H_k) R_k$$

The values with bar on the top are predicted value and K is Kalman gain. The full derivation of above equations is shown in [4].

2.2.1. Object tracking using Kalman filter To use Kalman filter for object tracking we assume that the motion of the object is almost constant over frames. The state variables, dynamic matrix and measurement matrix commonly used for 2D tracking can be found in [5].

$$\begin{pmatrix} x_{obs} \\ y_{obs} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} + N(0, R), \quad R = \begin{pmatrix} r^2 & 0 \\ 0 & r^2 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ \dot{x}' \\ \dot{y}' \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} + N(0, Q), \quad Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q^2 & 0 \\ 0 & 0 & 0 & q^2 \end{pmatrix}$$

3. Tracking Algorithm

Figure 1 briefly depicts the basic steps of algorithm in connection with SIFT features and a Kalman filter of the object. As shown on the right side of Figure 1, we store a collection of SIFT features found and a Kalman filter that is used to predict the next location for each object. The information is kept even when the object is disappeared from frame, so that it can be reused when the object comes into sight in the future.

The tracking algorithm begins when a user selects the object the object to track. The SIFT features found in the location of the object are stored. In the next frame, a Kalman filter makes prediction for a possible location of the object. The algorithm looks into either the location predicted by the Kalman filter or the identical location as the previous frame depending on how reliable the Kalman filter is. We use the prediction of Kalman filter when the prediction error is smaller than the pre-set threshold value. In the beginning of the algorithm, where we do not have enough information of the motion of the object, the identical location of the object as the previous frame is considered. The following step matches the keypoints between the candidate area of object and the stored SIFT features. The true location of the object is found from the location of matched keypoints and the measurement value is used to correct Kalman filter. From the location found, the algorithm continues on to the next frame repeating the same process. Figure 2 shows the screen shot while running the tracking algorithm.

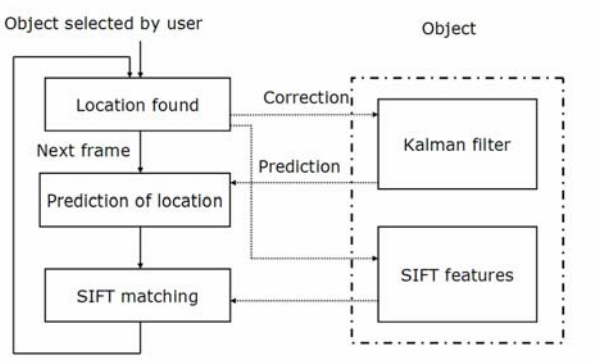

Figure 1 Algorithm flowchart Each step of algorithm interacts with the Kalman filter and the stored SIFT features of the object, shown on the right side. When the error of prediction is large, prediction is set to be the location of the object in the previous frame.

3.1 The State Vector

$$\begin{pmatrix} x_{obs} \\ y_{obs} \\ w_{obs} \\ h_{obs} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ w \\ h \\ \dot{w} \\ \dot{h} \end{pmatrix} + N(0, R)$$

$$\begin{pmatrix} x' \\ y' \\ \dot{x}' \\ \dot{y}' \\ w' \\ h' \\ \dot{w}' \\ \dot{h}' \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ w \\ h \\ \dot{w} \\ \dot{h} \end{pmatrix} + N(0, Q)$$

In the tracking algorithm, not only location but also size of the tracked object is estimated. As an extension from section 2.2.1, the width and height of the rectangular

selection, and the velocity of change for the width and height are added as components of state vector.

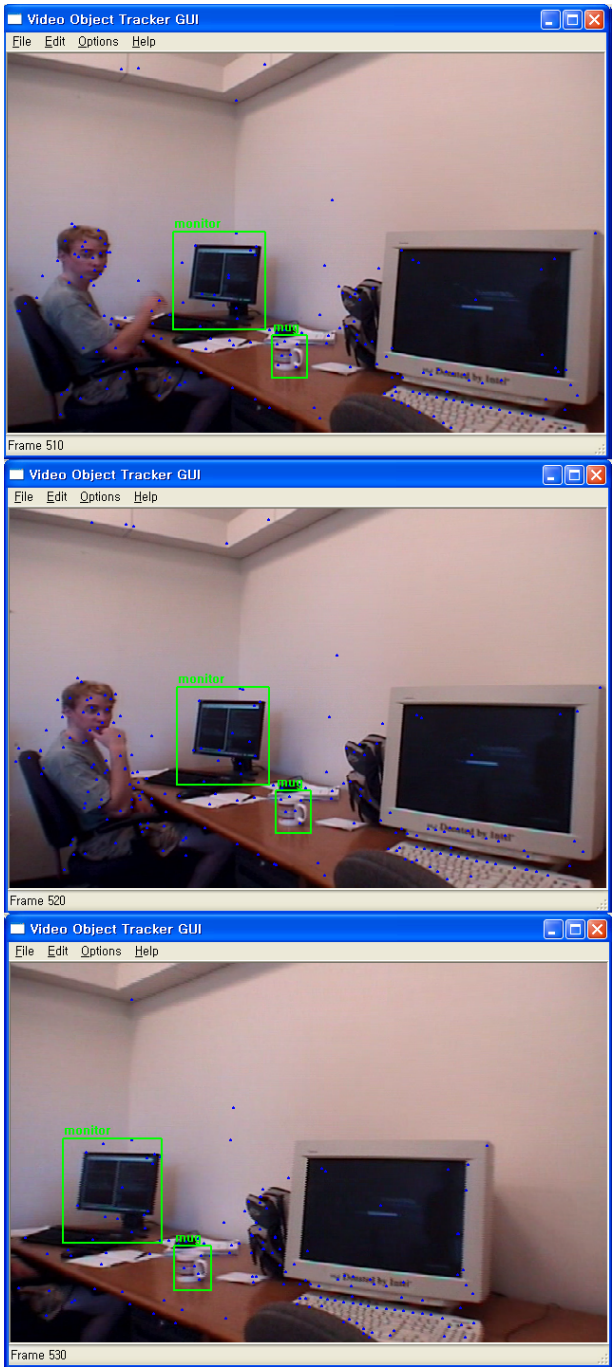

Figure 2 Screen shot of every 10 frames The objects are shown in green boxes, and SIFT features are shown in blue dots. A monitor and a mug are being tracked.

The Kalman filter used for the tracking algorithm is a simple extension from 2.2.1 assuming the location (x, y) and the size (w, h) are independent. The assumption is reasonable in the sense that the direction of which the object is moving does not have a linear relationship with the width or height of the object.

3.2 Measurement Using SIFT Features

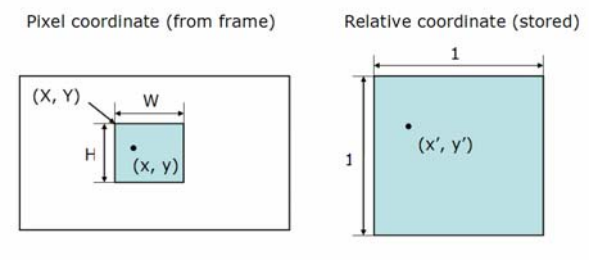

Figure 3 Transform of feature location from pixel coordinate to relative coordinate

The coordinates of SIFT features are transformed into relative location of the feature to be used as means of finding location and size of selected object. As seen in Figure 3, we rescale the selection rectangle into square with length 1. The relationship between the stored coordinate (x', y') and the pixel coordinate (x, y) can be easily written as:

$$x = X + x' W$$
$$y = Y + y' H$$

Suppose we have a new frame, and we found matched feature with relative coordinate (x', y') from pixel location of (x, y) in the frame. If there are more than one matched SIFT features for the object, we can calculate X, Y, H, W by solving the least-square solution of following matrix equation.

$$\begin{pmatrix} 1 & x_{1'} \\ 1 & x_{2'} \\ . & . \\ 1 & x_{n'} \end{pmatrix} \begin{pmatrix} X \\ W \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ .. \\ x_n \end{pmatrix}, \quad \begin{pmatrix} 1 & y_{1'} \\ 1 & y_{2'} \\ . & . \\ 1 & y_{n'} \end{pmatrix} \begin{pmatrix} Y \\ H \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ .. \\ y_n \end{pmatrix}$$

3.3. Change of Noise Model

Although SIFT features are distinctive and result in reliable matching in most of times, SIFT feature can rarely pick a matching point that is similar (usually points within the same object) but not at the exactly same location. The predictions are not very reliable after the single mistake. To reduce the effect of the wrong matching point onto the Kalman filter, we will design a different noise model for measurement update:

$$v_k \sim N(0, R_1) \quad with probability \quad \alpha$$
$$\sim N(0, R_2) \quad with probability \quad (1-\alpha)$$

When α is close to 1 and R_1 is small and R_2 is large, the rare error can be dissolved into case of noise model N (0, R_2). That is, the ordinary correct matching between SIFT features in two pictures corresponds to the noise model with low error (small R_1 and α close to 1) while the rare mismatch case corresponds to the noise model with higher error (large R_2), but low probability (1- α). After modifying the Kalman filter by the new noise model, the prediction is robust to wrong measurements.

Full derivation of the modified Kalman filter equations with the new noise model (density filtering) is available in the Appendix A.

4. Experiment

As a standard to compare, I manually leveled tracking objects at each frame. The performance of the proposed algorithm and simple optical flow method are compared in the sense of relative error from the manual standard. Please note that the optical flow algorithm compared is rather naïve algorithm calculated only on the four corners of the selected region. There are more sophisticated approaches that we were not able to compare against due to time constraints.

The optical flow works relatively well in the beginning, but the error blows up once it lost track of the object. The average performance measure is (error of proposed algorithm)/ (error of optical flow) = 0.4906. The plots comparing the two algorithms with different objects are shown in Figure 4. The large jump at the last frame for monitor (the second plot) and the mug (the third plot) are due to the fact that the objects moved out of the view.

5. Conclusion and Future Works

With SIFT features and Kalman filters to learn the motion, you can follow a general object that user selects. The nobility of the proposed algorithm is robustness in the cases when it loses track of the object. With higher resolution and more motion of camera involve, this work can further extended into finding the location of stationary objects as well as the odometer of camera.

6. Acknowledgement

I would like to thank to Steve Gould for his help setting up video labeler and providing data set to run and test the tracking algorithm. This project has been possible with his invaluable advice. Siddharth Batra kindly provided a library to find SIFT features modifying David Lowe's code. Professor Andrew Ng gave advices and guidelines. I would like to appreciate their contribution on the project.

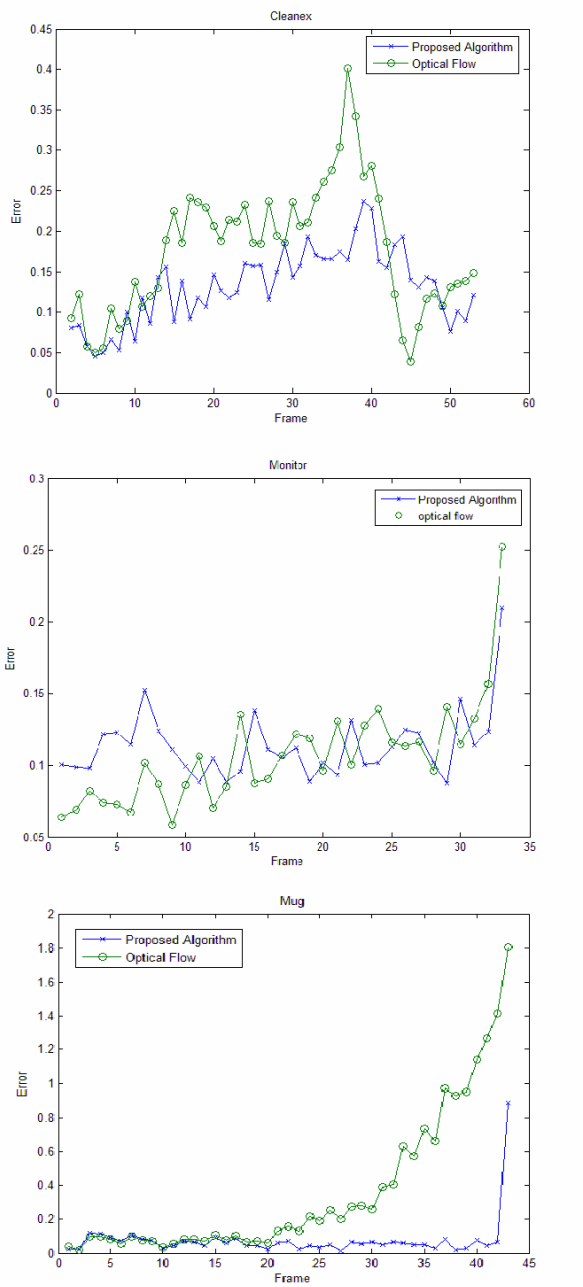

Figure 4 Plots comparing performance of proposed tracking algorithm against optical flow

7. References

[1] Yilmaz, A., Javed, O., and Shah, M., "Object tracking: A survey", *ACM Computing Surveys*, 38, 4, Article 13, Dec. 2006, 45 pages.

[2] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, 60, 2, 2004, pp. 91-110.

[3] Weissman, T., "EE378 Handout: Kalman Filter", *Lecture notes on EE378 Statistical Signal Processing, http://eeclass.stanford.edu/ee378/.*

[4] Intel Coporation, *OpenCV Reference Manual, http://www.cs.unc.edu/Research/stc/FAQs/OpenCV/OpenCVReferenceManual.pdf*, 1999-2001.

[5] Thrun, S., and Kosecka, J., "Lecture 12 Tracking Motion", *Lecture notes on CS223b, http://cs223b.stanford.edu/notes/CS223B-L12-Tracking.ppt*

Appendix A. Density Filtering

Suppose H is a deterministic matrix and U and N are independently Gaussian vectors. Then, the probability distribution of U given V is also Gaussian when V, U, and N are related as below:

$$V = HU + N$$
$$U|V = v \sim N(\mu_U G(v - H\mu_U), (I - GH)\Lambda_U)$$
$$G = \Lambda_U H^T (H\Lambda_U H^T + \Lambda_N)^{-1}$$

Now, suppose our noise model N is changed in accordance with random variable Z:

$$V|Z=1,\ V = HU + N_1,\ P(Z=1) = \alpha$$
$$V|Z=2,\ V = HU + N_2,\ P(Z=2) = 1 - \alpha$$

The distribution of U given V is still Gaussian but the mean and the variance is changed. The mean is easily calculated:

$$E[U|V] = E_Z[E[U|V,Z]]$$
$$\mu_U + \alpha G_1(V - H\mu_U) + (1-\alpha)G_2(V - H\mu_U)$$

To calculate variance, we can use the law of total variance.

$$VAR(U|V) = E_Z[VAR(U|V,Z)] + VAR_Z(E(U|V,Z))$$

The first term:

$$VAR(U|Z=1) = (I - G_1H)\Lambda_U$$
$$VAR(U|Z=2) = (I - G_2H)\Lambda_U$$
$$E_Z[VAR(U|V,Z)] = \Lambda_U + (\alpha G_1 + (1-\alpha)G_2)H\Lambda_U$$

Second term:

$$E[U|Z] = \mu_U + G_1(V - H\mu_U) = A_1, \quad w.p. \quad \alpha$$
$$E[U|Z] = \mu_U + G_2(V - H\mu_U) = A_2, \quad w.p. \quad 1-\alpha$$
$$VAR_Z(E(U|V,Z)) = \alpha A_1 A_1^T + (1-\alpha)A_2 A_2^T$$
$$- (\alpha A_1 + (1-\alpha)A_2)(\alpha A_1 + (1-\alpha)A_2)^T$$
$$= \alpha(1-\alpha)(A_1 - A_2)(A_1 - A_2)^T$$

The equations for modified Kalman filter that uses the new model for measurement update can be found by adequate substitutions of noise into the mean and variance found above. N is R, G is K, and P is the covariance.

$$v_k \sim N(0, R_1) \quad w.p. \quad \alpha$$
$$\sim N(0, R_2) \quad w.p. \quad (1-\alpha)$$
$$\hat{K}_k = \alpha \overline{P}_k H_k^T (H_k \overline{P}_k H_k^T + R_{k,1})^{-1}$$
$$+ (1-\alpha)\overline{P}_k H_k^T (H_k \overline{P}_k H_k^T + R_{k,2})^{-1}$$
$$\hat{X}_k = \overline{X}_k + \hat{K}_k(z_k - H_k \overline{X}_k)$$
$$\hat{P}_k = (I - \hat{K}_k H_k)\overline{P}_k + \alpha(1-\alpha)BB^T$$
$$where\ B = (K_1 - K_2)(z_k - H_k \overline{X}_k)$$

The hat means it is the value (Kalman gain, corrected state, posterior error covariance) of new noise model for measurement update.