TP4: Objects tracking in video sequences

Loic Simon Chaohui Wang Panagiotis Koutsourakis

22 May 2008

1 Theory

1.1 Cross-correlation

The normalized cross-correlation is a measure of the similarity between patches (i.e sub-windows) of different images. Its value is a real number between -1 and 1. Two similar windows have a high cross-correlation. Given two patches $\mathbf{u}$ and $\mathbf{v}$, of equal size $n \times n$, their cross-correlation is :

$$C(\mathbf{u}, \mathbf{v}) = \frac{\langle \mathbf{u} - \bar{\mathbf{u}}, \mathbf{v} - \bar{\mathbf{v}} \rangle}{\|\mathbf{u} - \bar{\mathbf{u}}\| \|\mathbf{v} - \bar{\mathbf{v}}\|} \tag{1}$$

where $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ are the mean values of the respective patches.

Intuitively, the cross-correlation can be thought as a normalized dot product, in other words it corresponds to the cosine between the vectors $\mathbf{u} - \bar{\mathbf{u}}$ and $\mathbf{v} - \bar{\mathbf{v}}$.

Based on this simple similarity measure, we can define a patch tracking approach for a sequence of images. Assuming that the object to track does not move too quickly, one can look in the next frame for the patch that best fits the one in the current frame. Here "best fits" means the one with the highest cross-correlation.

1.2 Kalman filter

The kalman filter is a linear technique to filter discrete data. It is well fitted to the study of dynamic systems, for which the evolution models are known. Let $x_t \in \mathbb{R}^n$ the vector encoding the state of the system we are focusing on (at time t). Let's assume that this system evolves following a linear and discrete process:

$$x_t = A x_{t-1} + w_{t-1} \tag{2}$$

In the previous equation, w_{t-1} is a 0 mean Gaussian noise with covariance matrix Q. A is a $n \times n$ matrix that encodes the transition from the previous to the current state.

Apart from that, a partial measurement $z_t \in \mathbb{R}^m$ ($m < n$ in general) is made on the state:

$$z_t = Hx_t + v_t \tag{3}$$

Commonly, the noise v_t is assumed Gaussian, with 0 mean and independent from w_t. Let's denote R its covariance matrix.

The Kalman filter takes both the evolution law and the measurement into account, in order to evaluate the system's state. It proceeds iteratively in two steps:

1. **Predicting :** A first guess $\hat{x}_t^-$ is estimated as well as an error quantity (the covariance matrix P_t^-).

$$\hat{x}_t^- = A\hat{x}_{t-1} \tag{4}$$
$$P_t^- = AP_{t-1}A^T + Q \tag{5}$$

2. **Correcting** This step add a correction based on the measurement z_t. We call x_t *a posteriori* estimation:

$$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1} \tag{6}$$
$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-) \tag{7}$$
$$P_t = (I - K_t H)P_t^- \tag{8}$$

   The matrix K_t called filter gain can be thought of as an inverse of H. If interested in further details, one can have a look at the following introduction to Kalman filtering: http://www.cs.unc.edu/%7Ewelch/media/pdf/kalman_intro.pdf

The idea here is to use the kalman filter to combine the information from a prior knowledge on the motion (eg constant speed,...) and the given measurements (an estimation of the position based here on the cross-correlation).

2 Computation

2.1 Tracking based on cross-correlation

Complete the function **measureCrossCorrelation** which arguments are the current frame, the patch representing the selected target, the position of the target in the previous frame, and the half-width (**searchdx**) and half-height (**searchdy**) of the search window. This function aims at computing the position of the window in the current frame that gives the maximum cross-correlation.

2.2 Kalman tracking

1. Explain how you can define a state vector to encode a constant velocity constraint ; how many components would have the state vector? What would be their meaning? Give details on how you would define the matrix A.

2. Fill in the function **twoStepKalmanFiltering** to implement both steps of the Kalman filter. This function takes as arguments the previous state estimator **previousState** ($\hat{x}_{t-1}$), the estimated error at previous state **PkPrevious** (P_{t-1}), the **measurement** (z_t) as well as the matrices A ,H, Q and R. It computes the new state estimator ($\hat{x}_t$) **currentState** and the error (P_t) **PkCurrent**.