Kernel-Based Object Tracking

Ekaterina Tikhoncheva

16.07.2014

Agenda

1. Introduction
2. Target Representation
3. Similarity Function
4. Target Localization
5. Calculation Results

Introduction

Tracking consists of two major components:

- Target Representation and Localization
- Filtering and Data Association

We introduce an Approach toward target representation and localization based on Mean-Shift Algorithm[1].

Target Representation

1. Target in the image
   is represented by an ellipsoidal region, which then will be normalized to a unit circle.
   $\Rightarrow$ Normalized pixel locations $\{x_i^*\}_{i=1\dots n}$ centered in 0.
2. Feature Space
   RGB color space quantized into $16 \times 16 \times 16$ bins.
   $\Rightarrow m = 16^3 = 4096$-bins histogram

3. Target Model
   is represented by its probability density function (pdf) in the feature space.

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|x_i^*\|^2)\delta[b(x_i^*) - u]$$

- $k : [0, \infty) \to \mathbb{R}$ is convex and monotonic decreasing profile of an isotropic kernel: $K(x) = k(\|x\|^2)$
- $b : \mathbb{R}^2 \to \{1 \dots m\}$ return bin of the pixel in the quantized feature space
- $C = \frac{1}{\sum_{i=1}^{n} k(\|x_i^*\|^2)}$ is a constant, which ensures $\sum_{u=1}^{m} \hat{q}_u = 1$

$\{x_i\}_{i=1\ldots n_h}$ are normalized pixel locations of target candidate, centered in y in the current frame.

3. Target Candidate
   is also represented by its probability density function (pdf) in the feature space.

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta[b(x_i) - u]$$

- $k : [0, \infty) \to \mathbb{R}$ is convex and monotonic decreasing profile of an isotropic kernel: $K(x) = k(\|x\|^2)$
- h: defines the scale of the target candidate
- $b : \mathbb{R}^2 \to \{1 \ldots m\}$ return bin of the pixel in the quantized feature space
- $C_h = \frac{1}{\sum_{i=1}^{n_h} k(\|\frac{y-x_i}{h}\|^2)}$ is a constant, which ensures $\sum_{u=1}^{m} \hat{p}_u = 1$

Similarity Function

To calculate distance between target and candidates we need to define a distance function between two distributions.

Similarity Function

To calculate distance between target and candidates we need to define a distance function between two distributions.

$$d(y) = \sqrt{1 - \hat{\rho}(y)}$$

where $\hat{\rho}(y) \equiv \hat{\rho}[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u}$ is the sample estimate of Bhattacharrya coefficient.

Similarity Function

To calculate distance between target and candidates we need to define a distance function between two distributions.

$$d(y) = \sqrt{1 - \hat{\rho}(y)}$$

where $\hat{\rho}(y) \equiv \hat{\rho}[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u}$ is the sample estimate of Bhattacharrya coefficient.
Note : $\hat{\rho}(y)$ is the cosine of the angle between the m-dimensional unite vectors $(\sqrt{\hat{p}_1(y)}, \dots \sqrt{\hat{p}_m(y)})$ and $(\sqrt{\hat{q}_1}, \dots \sqrt{\hat{q}_m})$.

Similarity Function

To calculate distance between target and candidates we need to define a distance function between two distributions.

$$d(y) = \sqrt{1 - \hat{\rho}(y)}$$

where $\hat{\rho}(y) \equiv \hat{\rho}[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u}$ is the sample estimate of Bhattacharrya coefficient.

Note : $\hat{\rho}(y)$ is the cosine of the angle between the m-dimensional unite vectors $(\sqrt{\hat{p}_1(y)}, \ldots \sqrt{\hat{p}_m(y)})$ and $(\sqrt{\hat{q}_1}, \ldots \sqrt{\hat{q}_m})$.

Note : $d(y)$ is a metric.

Similarity Function

To calculate distance between target and candidates we need to define a distance function between two distributions.

$$d(y) = \sqrt{1 - \hat{\rho}(y)}$$

where $\hat{\rho}(y) \equiv \hat{\rho}[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u}$ is the sample estimate of Bhattacharrya coefficient.
Note : $\hat{\rho}(y)$ is the cosine of the angle between the m-dimensional unite vectors $(\sqrt{\hat{p}_1(y)}, \ldots \sqrt{\hat{p}_m(y)})$ and $(\sqrt{\hat{q}_1}, \ldots \sqrt{\hat{q}_m})$.
Note : $d(y)$ is a metric.
Note : $d(y)$ is smooth $\Rightarrow$ we can apply gradient-based optimization!

Target Localization

The Problem of Target Localization is equal to Minimization of $d(y)$ OR Maximization of Bhattacharrya coefficient $\hat{\rho}(y)$.

Let $\hat{y}_0$ be a target location in the previous frame.

$$\hat{\rho}[\hat{p}(y), \hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{y}_0)\hat{q}_u} + \frac{C_h}{2}\sum_{u=1}^{m} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)$$

where $w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}}\delta[b(x_i)-u]$.
Maximize the second term, because the first term is independent on y:

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}$$

where $g(x) = -k'(x)$, assumed that $k'(x)$ exist almost everywhere.

Epanechnikov Kernel

In d-dimensional case

$$k(x) = \begin{cases} \frac{1}{2} {c_d}^{-1}(d+2)(1-\|x\|) & \text{if } \|x\| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

where c_d is the volume of the unit d-dimensional sphere.
In one dimensional case: $d = 1$, $c_d = 2\pi$.

For this Kernel $g(x)$ is a constant. If we use it for $\hat{y}_1$ we become:

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)} \Rightarrow \hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i}$$

Algorithm

Input: Target model $\{q_u\}_{u=1...m}$
Location $\hat{y}_0$ of target in previous frame

1. Initialize location of the target in the current frame with $\hat{y}_0$, compute $\{\hat{p}(\hat{y}_0)_u\}_{u=1...m}$
2. Compute weights $w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(x_i) - u]$
3. Compute the next location $\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i}$
4. If $\|\hat{y}_1 - \hat{y}_0\| < \epsilon$ Stop
   Else $\hat{y}_0 = \hat{y}_0$ and go to 2

Calculation Results

(ResultRedcup2.avi)

References I

[1] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Computer Society*, 25:564–577, 2003.

The End

Thank you for your Attention!