Robust Density Comparison for Visual Tracking

Omar Arif
omararif@gatech.edu
Patricio Antonio Vela
pvela@ece.gatech.edu

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta GA, 30332, USA

Many problems in computer vision require measuring the distance between two distributions. For example, in visual tracking, the object to be tracked is presumed to be characterized by a probability distribution. To track the object, each image of the sequence is searched to find the region whose sample distribution closely matches the model distribution. This paper presents a technique to robustly compare two distributions represented by samples, without explicitly estimating the density. The method is based on mapping the distributions into a reproducing kernel Hilbert space, where eigenvalue decomposition is performed. Retention of only the top M eigenvectors minimizes the effect of noise on density comparison. A sample application of the technique is visual tracking, where an object is tracked by minimizing the distance between a model distribution and candidate distributions.

Density Comparison: Let $\{u_i\}_{i=1}^n$, with $u_i \in \mathbb{R}^d$, be a set of n observations. A probability density at a point u can be estimated by the construction of a finite series of orthogonal functions [1],

$$p(u) = \sum_{k=1}^{M} \omega^k \Psi^k(u) \,. \tag{1}$$

where $\{\Psi^k\}_{k=1}^M$ are M orthonormal functions with coefficients given by ω^k. The orthonormal functions and the coefficients can be computed using kernel principal component analysis (KPCA) [2] and are given by

$$\begin{aligned} \Psi^k(u) &= \langle V_k, \phi(u)\rangle = \textstyle\sum_{i=1}^n w_i^k \boldsymbol{k}(u,u_i), \\ \omega^k &= \tfrac{1}{n}\textstyle\sum_{i=1}^n \Psi^k(u_i). \end{aligned} \tag{2}$$

Using Equations (2) in Equation (1), the probability density estimate at a test point u has the form,

$$p(u) = \sum_{k=1}^{M} \omega^k \Psi^k(u) = \sum_{k=1}^{M} \omega^k \left\langle V^k, \phi(u) \right\rangle \equiv \langle \mu_r[P_u], \phi(u)\rangle \,, \tag{3}$$

where the final equality defines the proposed robust mean map $\mu_r : P_u \to \mu_r[P_u]$, with $\mu_r[P_u] := \sum_{k=1}^M \omega^k V^k$. The robust distance measure between the two distributions P_u and P_v is defined using the robust mean map μ_r, and we call it the robust Maximum Mean Discrepancy (**rMMD**) (MMD measure has been defined in [3], where KPCA is not carried out in the kernel space),

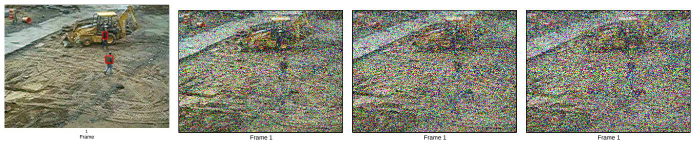
$$\begin{aligned} D_r(P_u, P_v) &:= ||\mu_r[P_u] - \mu_r[P_v]|| \,, & (4) \\ &= ||\boldsymbol{\omega_u} - \boldsymbol{\omega_v}|| \,, & (5) \end{aligned}$$

where $\boldsymbol{\omega_u} = [\omega_u^1, \ldots, \omega_u^M]^T$ and $\boldsymbol{\omega_v} = [\omega_v^1, \ldots, \omega_v^M]^T$. Since both mean maps live in the same eigenspace, the eigenvectors V^k have been dropped in Equation (5).

Visual Tracking: To apply the robust density comparison method to visual tracking, assume that the target object undergoes a geometric transformation T from a region R to a region $\tilde{R}$, such that $R = T(\tilde{R}, a)$, where $a = [a_1, \ldots, a_g]$ is a vector containing the parameters of transformation and g is the total number of transformation parameters. The objective is to estimate the transformation parameters a. Let $\{u_i\}_{i=1}^n$ and $\{v_i\}_{i=1}^m$ be the pixel vectors extracted from region R and $\tilde{R}$. Let the pixel vectors extracted from the region R are given by $u_i = [I(x), x]$, where $I(x)$ be the p-dimensional appearance vector extracted from image I at the spatial location x, and let $v_i = [I(\tilde{x}_i), T(\tilde{x}_i, a)]^T = [I(\tilde{x}_i), x_i]^T$. The rMMD measure between the distributions of the regions R and $\tilde{R}$ is given by the Equation (4), and with the L_2 norm is

$$D_r = \sum_{k=1}^{M} \left(\omega_u^k - \omega_v^k\right)^2 , \tag{6}$$

where the M-dimensional robust mean maps for the two regions are $\omega_u^k = \frac{1}{n}\sum_{i=1}^n \Psi^k(u_i)$ and $\omega_v^k = \frac{1}{m}\sum_{i=1}^m \Psi^k(v_i)$. Gradient descent can be used to



(a) Original (b) Noise $\sigma = .1$ (c) Noise $\sigma = .2$ (d) Noise $\sigma = .3$

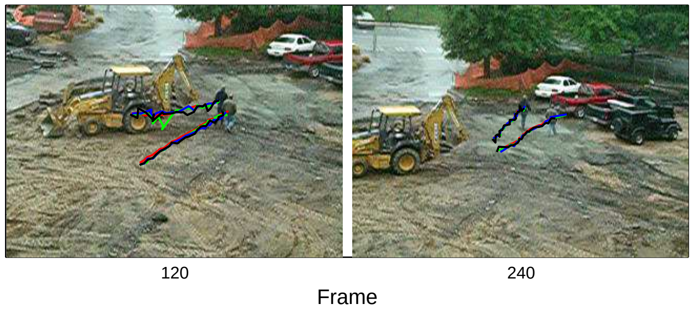


Figure 1: Construction Sequence. Trajectories of the track points are shown. Red: No noise added, Green: $\sigma = .1$, Blue: $\sigma = .2$,Black: $\sigma = .3$. The tracker tracked in all the cases.

minimize the distance with respect to the transformation parameter a. The gradient of Equation (6) with respect to the transformation parameters a is

$$\nabla_a D_r = -2 \sum_{k=1}^{M} \left(\omega_u^k - \omega_v^k\right) \nabla_a \omega_v^k , \tag{7}$$

where $\nabla_a \omega_v^k = \frac{1}{m}\sum_{i=1}^m \nabla_a \Psi^k(v_i)$. The gradient of $\Psi^k(v_i)$ with respect to a is,

$$\nabla_a \Psi^k(v_i) = \nabla_x \Psi^k(v_i) \cdot \nabla_a T(\tilde{x}, a), \tag{8}$$

where $\nabla_a T(\tilde{x}, a)$ is a $g \times 2$ Jacobian matrix of T and is given by $\nabla_a T = [\frac{\partial T}{\partial a_1}, \ldots, \frac{\partial T}{\partial a_g}]^T$. The gradient $\nabla_x \Psi^k(v_i)$ is computed as,

$$\nabla_x \Psi^k(v_i) = \frac{1}{\sigma_s^2} \sum_{j=1}^{n} w_j^k \boldsymbol{k}(u_j, v_i)(\pi_s(u_j) - x_i), \tag{9}$$

where π_s is a projection from d-dimensional pixel vector to its spatial coordinates, such that $\pi_s(u) = x$ and σ_s is the spatial bandwidth parameter used in kernel $\boldsymbol{k}$. The transformation parameters are updated using the following equation,

$$a(t+1) = a(t) - \delta t \nabla_a D_r, \tag{10}$$

where δt is the time step.

Figure 1 shows results of tracking two people under different level of Gaussian noise. Matlab command `imnoise` was used to add zero mean Gaussian noise of $\sigma = [.1, .2, .3]$. The sample frames are shown in Figure 1(b), 1(c) and 1(e). The trajectories of the track points are also shown. The tracker was able to track in all cases.

[1] M. Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, pages 669–688, 2002.

[2] B. Scholköpf, A. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.

[3] A. Smola, A. Gretton, L. Song, and B. Scholkopf. A Hilbert space embedding for distributions. *Lecture Notes in Computer Science*, 2007.