KERNEL SELECTION BY MUTUAL INFORMATION FOR NONPARAMETRIC OBJECT TRACKING

J. M. Berthommé, T. Chateau and M. Dhome
LASMEA - UMR 6602, Université Blaise Pascal, 24 Avenue des Landais, 63177 Aubière Cedex, France

Keywords: Kernel Selection, Information Theory, Nonparametric Tracking.

Abstract: This paper presents a method to select kernels for the subsampling of nonparametric models used in real-time object tracking in video streams. We propose a method based on mutual information, inspired by the CMIM algorithm (Fleuret, 2004) for the selection of binary features. This builds, incrementally, a model of appearance of the object to follow, based on representative and independant kernels taken from points of that object. Experiments show gains, in terms of accuracy, compared to other sampling strategies.

1 INTRODUCTION

Many methods of object tracking are based on the appearance of the object in the first frame of the video. The aim is to find an area whose appearance is closer to the previous one in the next frames.

This modeling is called nonparametric because it makes no assumption on the data distribution. It is largely based on the use of kernel functions whose bandwidth is critical. To fit to local densities, Sylvain Boltz (Boltz et al., 2009) used the k-nearest neighbors (k-NN) instead of the Parzen windows (Parzen, 1962) and showed how to estimate the Kullback-Leibler divergence within this framework. He made the connection with Mean-Shift described by (Fukunaga and Hostetler, 1975) and revived by (Comaniciu et al., 2000).

However k-NN sin by their computational complexity. For two populations of n and m points in dimension d, we have to make $O(nmd)$ calculations to get the distances and $O(nm\,log\,m)$ to sort them. Garcia et al. (Garcia et al., 2008) showed that GPU parallelization pushed the limit and was better than approximate k-NN. This technical resolution is not challenged here, we only seek to reduce "upstream" the number of points of the model.

For this we propose an algorithm based on information theory to select representative and independant kernels. Once the model is built, the tracking is performed by a sequential particle filter whose observation function uses an approximation of the Kullback-Leibler divergence by the k-nearest neighbors.

2 METHOD

The initialization takes place in the first frame. The method is assumed to access to the *Region of Interest* (ROI) containing the cropped portion of the object to follow. The labelling of the pixels is binary (0/1) and is the *a priori* to start. The image labels are noted Y. Figure 1 shows an example of appearance to track and the map of the associated labels.

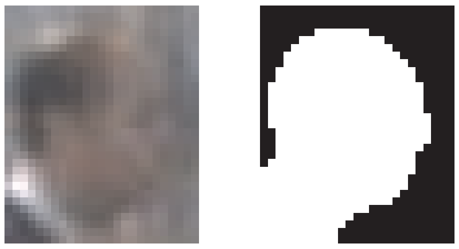
Figure 1: Labelling of the object of interest. To the left: image of the object to track. To the right: labels image.

2.1 Kernel Generation and Optimization

The aim is to select special points and kernels associated with the object to track. We first choose M random pixels $\mathbf{x_m}(u_m, v_m)$ in the ROI. Each one is coded by U,V,R,G,B and is the support of nine kernels in the D following dimensions: R, G, B, UVR, UVG, UVB, UV, RGB and $UVRGB$. Each dimension is normalized by its highest amplitude, from 0 to 255 for the 8-bit colors, 1 to w for U and 1 to h for V. Color C includes the channels R, G, B. Then, for each

$\mathbf{x_m}$, nine maps of normalized Euclidean distances are computed for all the pixels $\mathbf{x}(u,v)$ of the ROI:

$$\begin{aligned}
d_C(\mathbf{x},\mathbf{x_m}) &= \left(\frac{C(u,v)-C(u_m,v_m)}{255}\right)^2\\
d_{UV}(\mathbf{x},\mathbf{x_m}) &= 1/2\left\{\left(\frac{u-u_m}{w-1}\right)^2+\left(\frac{v-v_m}{h-1}\right)^2\right\}\\
d_{UVC}(\mathbf{x},\mathbf{x_m}) &= 1/2\left\{d_{UV}(\mathbf{x},\mathbf{x_m})+d_C(\mathbf{x},\mathbf{x_m})\right\}\\
d_{RGB}(\mathbf{x},\mathbf{x_m}) &= 1/3\{d_R(\mathbf{x},\mathbf{x_m})+d_G(\mathbf{x},\mathbf{x_m})+\ldots\\
&\qquad d_B(\mathbf{x},\mathbf{x_m})\}\\
d_{UVRGB}(\mathbf{x},\mathbf{x_m}) &= 1/2\left\{d_{UV}(\mathbf{x},\mathbf{x_m})+d_{RGB}(\mathbf{x},\mathbf{x_m})\right\}
\end{aligned}$$

For each distance, we define a probability to belong to the object or not by taking a Gaussian kernel $\mathtt{K}$. This step introduces a parameter λ, related to the kernel bandwidth, which spreads more or less the distribution around $\mathbf{x_m}$. At this stage, no normalization is performed to remain homogeneous with the Boolean labels. However, round to a normalization, the notions of kernel and distribution remain valid. Thus:

$$P[\mathbf{x}\in Object] = \mathtt{K}_{\mathtt{D},\lambda,\mathbf{x_m}}(\mathbf{x}) \text{ with :}$$

$$\mathtt{K}_{\mathtt{D},\lambda,\mathbf{x_m}}(\mathbf{x}) = \begin{cases} e^{-\lambda.d_D(\mathbf{x},\mathbf{x_m})} & \text{if } Y(\mathbf{x_m})=1\\ 1-e^{-\lambda.d_D(\mathbf{x},\mathbf{x_m})} & \text{if } Y(\mathbf{x_m})=0\end{cases}$$

Subsequently, we call X the map of probabilities of the pixels associated with the kernel $\mathtt{K}_{\mathtt{D},\lambda,\mathbf{x_m}}$. The parameter λ is very important. We optimize it by maximizing the mutual information I between the probability map X and the labels Y (MacKay, 2003):

$$I(X;Y) = H(X)+H(Y)-H(X,Y)$$

This optimization via, for example, the Levenberg-Marquardt method quickly converges to a solution λ^*. We do not detail it and simply present a possible evolution of I depending on λ in Figure 2.

Figure 3 shows the evolution of X from white to black depending on the kernel bandwidth λ. When λ becomes small the map of probabilities becomes all white ($\lim\limits_{\lambda\to 0^+} e^{-\lambda.d}=1$) and the information exchanged with the labels null. When λ becomes large the probability map is all black ($\lim\limits_{\lambda\to+\infty} e^{-\lambda.d}=0$) and mutual information null again. Between the two there is a λ^* solution as $0\le I(\lambda^*)\le H(Y)$.

2.2 Kernel Selection

$N=9M$ kernels, optimal in the sense of λ, were generated but only K with $K\ll N$ are selected. The goal is to keep the set the most representative of the labels

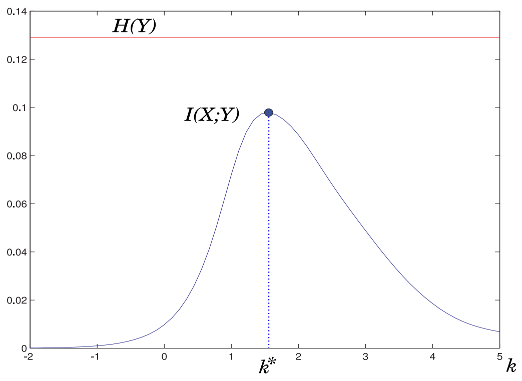

Figure 2: Kernel bandwidth optimization by maximizing the mutual information with the labels Y. $I=f(\lambda)$ with $\lambda=10^k$ where k = [-2, 5].

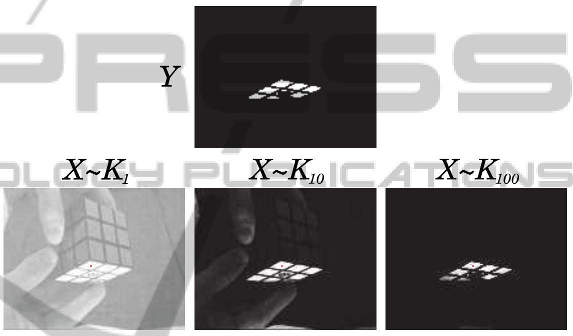

Figure 3: Evolution of the map of probabilities X based on the kernel bandwidth. Y *vs* X for λ = {1, 10, 100}.

Y of the ROI of the first image. This was performed by the CMIM algorithm (*Conditional Mutual Information Maximization*) (Fleuret, 2004) with a novelty to compare booleans to probabilities:

```
for n=1...N do
    s[n] ← $I(Y;X_n)$
end
for k=1...K do
    $\nu[k]$ = arg max$_n$ s[n]
    for n=1...N do
        s[n] ← min{s[n], $I(Y;X_n|X_{\nu[k]})$}
    end
end
```

Algorithm 1: CMIM algorithm.

$$\begin{aligned}I(Y;X_n|X_m) =& H(Y,X_m)-H(X_m)\\ &-H(Y,X_n,X_m)+H(X_n,X_m)\end{aligned}$$

For the calculations of $H(X_n,X_m)$ and $H(Y,X_n,X_m)$ the labels Y and the probability maps X_n and X_m are assumed to get as much information as possible together. For instance, for one pixel, if $p_n=0.8$ and $p_m=0.7$ then

$p_{(X_n=1)\cap(X_m=1)} = min(p_n, p_m) = 0.7$. This exchange of information can be represented by a noisy channel with $H(X_n) \leq H(X_m)$.

Figure 4 shows an example of the selection of five kernels by CMIM based on Figure 1:

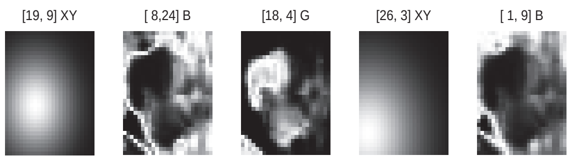

Figure 4: Selection of 5 kernels by CMIM.

2.3 k-PPV MCMC Tracking

The goal is to find in each new image the closest ROI to the original. Information is normalized by dimension and weighted on UVRGB by the weights $[1/4, 1/4, 1/6, 1/6, 1/6]$. Each image can be viewed as a probability density function (PDF) and compared with another one by a similarity measure, namely the Kullback-Leibler divergence. (Boltz et al., 2009) showed how to estimate it in a k-NN framework and why it is well aware of the local densities in high dimensions.

For a reference population R of n_R points in dimension d and a target population T of n_T points in the same dimension ($n_R \neq n_T$ a priori), with $\rho_k(U,s)$, the Euclidean distance between the point s and its k^{th} nearest neighbor in U ($U \equiv R$ or T), the Kullback-Leibler divergence $\mathfrak{D}_{KL}(T,R)$ can be estimated, in an unbiased way, by:

$$\mathfrak{D}_{\mathrm{KL}}(T,R) \overset{\mathrm{kPPV}}{=} \log\frac{n_R}{n_T - 1} + \frac{d}{n_T}\sum_{s\in T}\log\frac{\rho_k(R,s)}{\rho_k(T,s)}$$

An ideal estimation would consider all the pixels of R and T. However, to avoid the combinatorial explosion or just speed up the calculations, we try to downsample these regions. The question is how. We have compared three types of sub-samples: two regular, one random and one from the kernels selected by CMIM.

Trackings use a sequential particle filter (Arulampalam et al., 2002) with a Markov chain (Khan et al., 2005). Each particle represents a region of the image. The upper left corner of the reference ROI R indicates the first position. From the second we apply N_p random transformations φ_i to R. It generates N_p particles or ROI targets T_i whose weight is: $w_i = e^{-\mu.\mathfrak{D}_{\mathrm{KL}}(T_i,R)}$ with μ a fixed constant. The particle with the maximum weight takes on the new position. The others are not forgotten. They will generate new particles in the subsequent iterations. The reproduction rate is governed by the Metropolis-Hastings rule: $min(1, w_{new}/w^*) < rand()$. Few particles of low weight are also "burned" at each iteration.

We have also several important assumptions: that the reference R doesn't not change, that the object is far from the camera and therefore that the ROI undergoes only translations and remains fixed.

3 EXPERIMENTS

A manual tracking provided the ground truth GT. The particle filter algorithm truth AT was then compared to GT following different configurations.

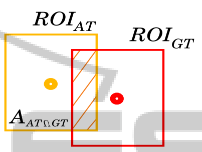

Figure 5: Intersection area $\mathcal{A}_{AT\cap GT}$ of the ROI of the algorithm AT and of the ground truth GT in the same image.

To determine the quality η of a tracking, AT and GT were compared image by image. We calculated their reports of intersection area and union area and check whether for a given tolerance τ it makes "fit" the tracked ROIs to the ground truth or not. This state, good or not, is rated β. For N_i images:

$$\beta_i(\tau) = \begin{cases} 1 & \text{if } \frac{\mathcal{A}^i_{AT\cap GT}}{\mathcal{A}^i_{AT\cup GT}} \geq \tau \\ 0 & \text{else} \end{cases}$$

$$\eta(\tau) = \frac{1}{N_i}\sum_{i=1}^{N_i}\beta_i(\tau)$$

Four significant subsamplings are presented: regular on raw data ("Raw"), regular on data smoothed by a Gaussian kernel ("Gauss"), random on data averaged per Voronoi cell ("RandVor"), and finally from the convolution of kernels selected by CMIM with their optimized bandwidth ("KerOpt").

Results are based on the sequence "WalkByShop1cor" of CAVIAR between images 192 and 309. We tried to follow the head of a man on a window of size 31×25 pixels. The curves show the variation of good tracking η depending on the tolerance τ on the percentage of area common with the ground truth. The four curves in green, khaki, brown and red in Figure 6 represent subsamplings of a pixel of 1, 4, 7 and 10 *i.e.*: 775, 56, 20 and 12 points for the configurations "Raw", "Gauss" and "RandVor", from top to bottom. Down on the same figure, the three curves gray, violet and cherry compile the "KerOpt" configuration for respectively: 55, 30 and 5 points filtered from the selected kernels.

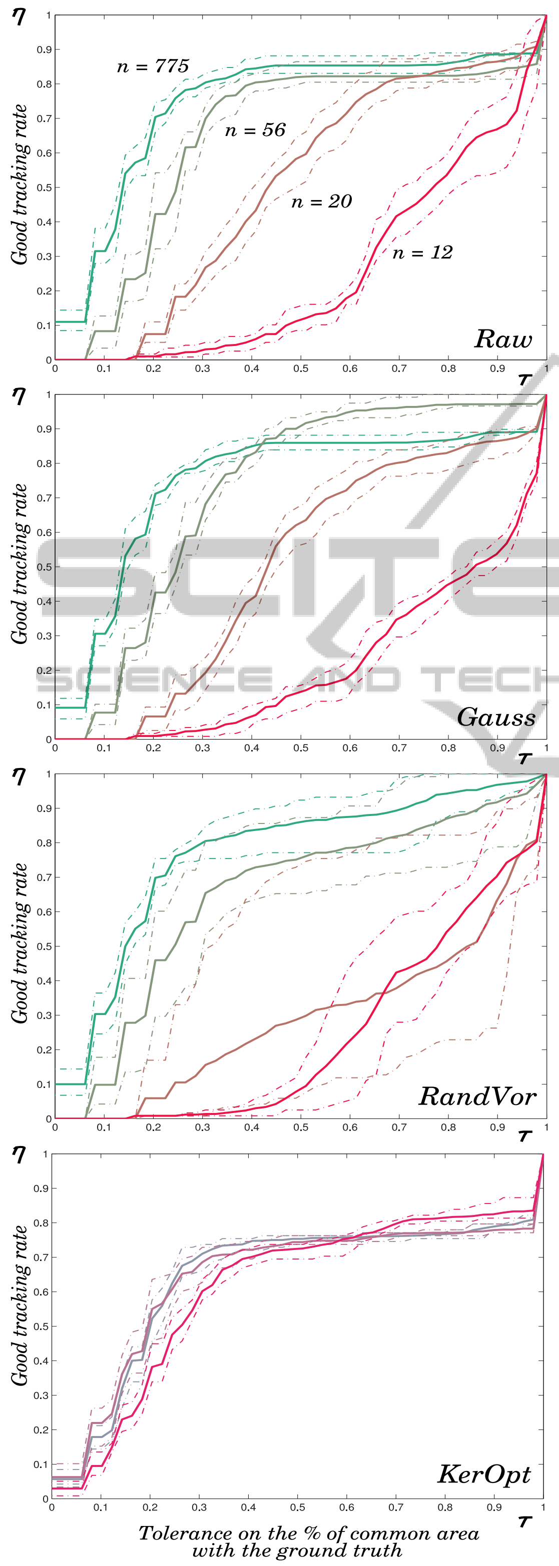

Figure 6: Tracking performances η(τ) of the subsamplings ”Raw”, ”Gauss” and ”RandVor” for K = {775, 56, 20, 12} points and ”KerOpt” for K = {55, 30, 5} points.

4 CONCLUSIONS

Our goal was to track a target with appearance model at low cost, *i.e.* based on few points. We have shown how to build a light model made up of independent and representative kernels of the prime appearance. Trackings made with this filtering were compared to other more traditional and showed equivalent performance for a number of points lower.

However, many things has to be improved. Here's a partial list: 1) integrate the temporal coherence in addition to the spatial coherence by introducing explicit time T in a new representation, for instance: $UVRGBT$, 2) make evolve the reference ROI R along the tracking by an on-line learning of new likely labels, 3) add new parameters to the transformation φ to consider rotations or homotheties or even why not see φ in a nonparametric way by considering each point of the model as a single control point connected to other by a consistent deformable mesh. All these points seem difficult but not unattainable.

REFERENCES

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188.

Boltz, S., Debreuve, E., and Barlaud, M. (2009). High-dimensional statistical measure for region-of-interest tracking. *IEEE Transactions on Image Processing*, 18(6):1266–1283.

Comaniciu, D., Ramesh, V., and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *CVPR*, volume 2, pages 142–149. Published by the IEEE Computer Society.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *The Journal of Machine Learning Research*, 5:1531–1555.

Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40.

Garcia, V., Debreuve, E., and Barlaud, M. (2008). Fast k nearest neighbor search using GPU. In *CVPR Workshop on Computer Vision on GPU (CVGPU)*, Anchorage, Alaska, USA.

Khan, Z., Balch, T., and Dellaert, F. (2005). Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1805–1918.

MacKay, D. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press.

Parzen, E. (1962). On the estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.