Probabilistic Tracking on Riemannian Manifolds

Yi Wu[1], Bo Wu[2], Jia Liu[2], Hanqing Lu[1]
[1]*Institute of Automation, Chinese Academy of Sciences, Beijing*
[2]*Institute of Electronics, Chinese Academy of Sciences, Beijing*
ywu@nlpr.ia.ac.cn

Abstract

The covariance region descriptor recently proposed in [1] has been proved robust and versatile for a modest computational cost. The covariance matrix enables efficient fusion of different types of features, where the spatial and statistical properties as well as their correlation are characterized. The similarity of two covariance descriptor is measured on Riemannian manifolds. Relying on the same metric, but within a probabilistic framework, we propose a novel tracking approach on Riemannian manifolds. The particle filtering technique allows us to better handle background clutter, as well as the temporary occlusions of the target. Furthermore, we extend the fast covariance computation to the tracking problem, which makes the tracking procedure more efficient. The proposed approach is robust to noises and much faster than the original search-based covariance tracker [2]. Extensive experimental results demonstrate greatly improved performance over classical color-based Bayesian tracker.

1. Introduction

Object tracking is a critical task in many computer vision applications such as surveillance, augmented reality and human-computer interfaces. Target representation is one of major components for a typical visual tracker. Extensive researches have been done on this topic.

Histograms have been proved to be a powerful representation for an image region. Discarding the spatial information, the color histogram is robust to the change of object pose and shape. Several successful tracking systems have been developed using color histograms [3, 4]. Recently, Stanley et al. [5] proposed a novel histogram named spatiogram in which each bin is spatially weighted by the mean and covariance of the locations of the pixels that contribute to that bin. Spatiogram captures not only the values of the pixels but their spatial relationships as well. To calculate the histogram efficiently, Fatih [6] proposed a fast way to extract histograms called integral histogram. When the integral histogram has been constructed, the histogram of any rectangular region can be computed efficiently independent of the region size.

The covariance region descriptor recently proposed in [1] has been proved robust and versatile for a modest computational cost [2, 7]. The covariance matrix enables efficient fusion of different types of features and its dimensionality is small. An object window is represented as the covariance matrix of features; the spatial and statistical properties as well as their correlation are characterized within the same representation. The similarity of two covariance descriptor is measured on Riemannian manifolds. Fatih [2] generalized the covariance descriptor to tracking problem by simply exhaustive searching in the whole image the region that best matches the model descriptor. This maximal likelihood estimation easily runs into problems by the background clutter and is very time-consuming.

Improvement for such situations is one of the benefits of our proposed probabilistic tracking approach. Relying on the same metric to comparing two covariance descriptors, we embed it within a sequential Monte Carlo framework. This requires the building of a local likelihood on Riemannian manifolds, the coupling of this observation model with a dynamical state space model, and the sequential approximation of the posterior distribution with a particle filter. This sample-based filtering technique enables to track multiple posterior modes, which is the key to escape from background distraction and to recover after temporary occlusions. Furthermore, we extend the fast covariance computation to tracking problem which makes the tracking procedure more efficient. The proposed approach is robust to noises and much faster than the original search-based covariance tracker [2].

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

2. Probabilistic tracking

2.1. Sequential Monte Carlo tracking

In the Bayesian perspective, object tracking can be viewed as a state estimation problem. Denote the state of target in time *t* and the corresponding observation as x_t and y_t, respectively. The state set up to time *t* is $x_{0:t}$, where x_0 is the initial state, and the observation set up to time *t* is $y_{0:t}$.

The purpose of tracking is to estimate the state given all the observations or equivalently to construct the filtering distribution $p(x_t|y_{0:t})$. Using the conditional independence properties, we can formulate the density propagation for the tracker as follows:

$$p(x_t|y_{0:t}) \propto p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_{0:t-1})\, dx_{t-1}$$

For visual tracking problems, the recursion can be used within a sequential Monte Carlo framework where the posterior $p(x_t|y_{0:t})$ is approximated by a weighted sample set $\{x_t^n, w_t^n\}_{n=1}^{N_s}$, where $\sum_{n=1}^{N_s} w_t^n = 1$. All the particles are sampled from a proposal density $q(x_t^n|x_{t-1}^n, y_t)$. The weight associated with each particle is formulated as follows:

$$w_t^n \propto \frac{p(y_t|x_t^n) p(x_t^n|x_{t-1}^n)}{q(x_t^n|x_{t-1}^n, y_t)} w_{t-1}^n$$

To avoid degeneracy of the weights, it is necessary to resample the particles from time to time. After resampling, all weights are equal to $\frac{1}{\mathrm{N_s}}$.

The common choice of proposal density is by taking $q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1})$. As a result, the weights become the local likelihood associated with each state $w_t^n \propto p(y_t|x_t^n)$. The Monte Carlo approximation of the expectation $\hat{x}_t = \frac{1}{N_s} \sum_{n=1}^{N_s} x_t^n \approx E(x_t|y_{0:t})$ is used as the state estimation at time *t*.

2.2. State dynamics

Our aim is to track a region of interest in the image plane. The shape of this region is defined by a rectangle. The state is defined as $\mathrm{x} = (d, s, v)$, where $d = (x, y)$ is the location, $\mathrm{s} = (w, h)$ represents the object size and $v = (v_x, v_y)$ is the velocity.

Commonly a first-order (B=0) or second-order auto-regressive dynamics is chosen to modal the state transition:

$$x_t = Ax_{t-1} + Bx_{t-2} + C\mathcal{N}(0, \Sigma)$$

Matrices *A*, *B*, *C* and *Σ* defining this dynamics could be learned or be set manually. Because these parameters are difficult to learn, the *AR* dynamics is not suitable for our case. We propose the following scheme to predict the state.

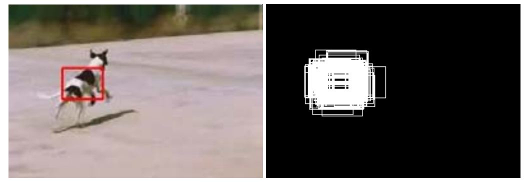


Figure 1: The state of target is shown by a red rectangle, and the corresponding particles are illustrated by white rectangles (totally 100 particles).

$$d_t^n = d_{t-1}, v_t^n = 0 \qquad u < u_0$$
$$d_t^n = d_{t-1}^n + v_{t-1}^n \qquad u \geq u_0$$

where u is a random number from distribution $U(0,1)$, $u_0 \in [0,1]$, d_t^n is the location of the *n*th particle in time *t*, d_{t-1} is the tracked target location in time *t-1*, v_t^n, v_{t-1}^n are the velocity of the *n*th particle in time *t* and *t-1*, respectively.

After that, we add small random disturbance to the predicted state x_t^n. The benefit of this prediction technique is that it avoids the difficult estimation of the matrices in the *AR* dynamics and puts no constraint on the motion of targets.

3. Covariance descriptor

The covariance region descriptor proposed in [1] enables efficient fusion of different types of features and its dimensionality is small. In this descriptor an object window is represented as the covariance matrix of features. The spatial and statistical properties as well as their correlation are characterized within the same representation.

Let *I* be the observed image, and *F* be the $W \times H \times d$ dimensional feature image extracted from *I*

$$F(x, y) = \Phi(I, x, y)$$

where *Φ* can be any mapping such as color, gradients, filter responses, etc. Let $\{z_i\}_{i=1}^N$ be the *d*-dimensional feature points inside a given rectangular region *R* of *F*. The region *R* is represented by the $d \times d$ covariance matrix of the feature points

$$Cov_R = \frac{1}{N-1} \sum_{n=1}^{N} (z_i - \mu)(z_i - \mu)^T$$

where *N* is the number of pixels in the region *R*. μ is the mean of the feature points.

The element (i, j) of Cov_R represents the correlation between feature *i* and feature *j*. When the extracted *d*-dimensional feature includes the pixel's coordinate, the covariance descriptor encodes the spatial information of features.

3.1. Fast Covariance Computation

With the help of integral images, the covariance descriptor can be calculated efficiently. When

frame 40 frame 55 frame 67

Figure 2: Tracking results for *Jogging* sequence

d(d+1)/2 integral images are constructed, the covariance descriptor of any rectangular region can be computed independent of the region size.

In the visual tracking problems, the tracked object only occupies small part in the image, as shown in Fig.1. If we compute the integral images for the whole image, many computation resources would be wasted. Observed from our experiments, more than 60% of the computation time is used to construct the integral images. Therefore, we only compute the integral images in the region which is occupied by all the particles. This technique makes the tracking procedure more efficient.

4. Metric on Riemannian manifolds

Supposing no features in the feature vector would be exactly identical, the covariance matrix is positive definite. Thus the nonsingular covariance matrix can be formulated as a connected Riemannian manifold. A manifold is locally similar to a Euclidean space. For differentiable manifolds, the derivative at a point X lies in a vector space T_X, the tangent space at that point. Each tangent space has an inner product $\langle , \rangle_X$ and the norm for a tangent vector is defined by $\|y\|_X^2 = \langle y, y \rangle_X$.

An invariant Riemannian metric on the tangent space is defined as

$$\langle y, z \rangle_X = tr\left(X^{-\frac{1}{2}} y X^{-1} z X^{-\frac{1}{2}}\right)$$

The exponential map associated to the Riemannian metric is given by

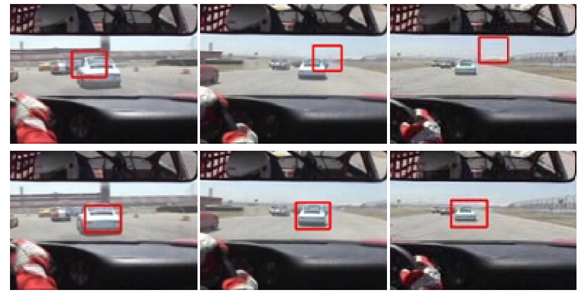
$$exp_X(y) = X^{\frac{1}{2}} exp\left(X^{-\frac{1}{2}} y X^{-\frac{1}{2}}\right) X^{\frac{1}{2}}$$

The logarithm uniquely defined at all the points on the manifold is

$$log_X(Y) = X^{\frac{1}{2}} log\left(X^{-\frac{1}{2}} Y X^{-\frac{1}{2}}\right) X^{\frac{1}{2}}$$

For a symmetric matrix, the exponential is given by

$$exp(\Sigma) = \sum_{k=0}^{\infty} \frac{\Sigma^k}{k!} = U exp(D) U^T$$



frame 106 frame 266 frame 300

Figure 3: Tracking results for *Race* sequence

Similarly, the logarithm series is

$$log(\Sigma) = \sum_{k=0}^{\infty} \frac{(-1)^{k-1}(\Sigma - I)^k}{k!} = U log(D) U^T$$

where $\Sigma = UDU^T$ is the eigenvalue decomposition of the symmetric matrix Σ. $exp(D)$ and $log(D)$ are the diagonal matrix of the eigenvalue exponentials and logarithms respectively.

The distance between symmetric positive definite matrices is measured by

$$d^2(X, Y) = \langle log_X(Y), log_X(Y) \rangle_X = tr\left(log^2\left(X^{-\frac{1}{2}} Y X^{-\frac{1}{2}}\right)\right)$$

To measure the similarity between the covariance matrices respectively corresponding to the target model and the candidate, we use the same metric on Riemannian manifolds. An exponential function of the distance is adopted as the local likelihood:

$$p(y_t | x_t^n) \propto exp\{-\lambda \cdot d^2[C^*, C(x_t^n)]\}$$

In all the experiments reported in this paper, we fixed the parameter λ to the same value $\lambda = 0.5$.

5. Experimental results

The experiments are carried out on 8 sequences totally more than 2000 frames. Due to the space limitation, here we only display tracking results on three sequences.

We extract 7-dimentional feature for each pixel:

$$F(x, y) = \left(x, y, R(x, y), G(x, y), B(x, y), I_x(x, y), I_y(x, y)\right)$$

where x and y are pixel location, R, G, B are the RGB color values and I_x, I_y are the intensity derivatives. For comparison, we choose HSV color histogram to calculate the local likelihood for conventional color-based particle filter tracker.

The first rows in Fig.2, Fig.3 and Fig.4 demonstrate the tracking results for color-based particle filter tracker and the second rows illustrate the results for covariance-based particle filter tracker. We can see that our proposed tracker is obviously better than traditional one. In the *Jogging* sequence illustrated in Fig.2, the

frame 35 frame 68 frame 91

Figure 3: Tracking results for *dog* sequence

target is fully occluded for a few frames and the color of clutters is similar to the target. As a result, the color-based probabilistic tracker fails to track the target, while our proposed tracking approach handles this occlusion case successfully. In the *Race* sequence, the background is changing from time to time; as a result the traditional approach is not stable and loses the target due to the influence of the sky.

To test sensitivity against noise, we contaminated the color values with additive zero mean Gaussian noise with standard deviation $\sigma = 0.5$, where sample results are shown in Fig.4. We can see that the performance of the color-based tracker significantly degrades, while our proposed approach tracks the target successfully. This benefits from the average filter during covariance computation which has also been pointed out in [2].

To quantify the performance of tracking algorithm, we adopt the criteria proposed in [8]:

$$S = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{|L|}\sum_{t\in L}\frac{2A_{O,t}^{(m)}}{A_{R,t}+A_{\hat{x},t}^{(m)}}\right)\in[0,1]$$

Refer to [8] for more details of this criteria.

The average performance on all the test sequences is shown in Table 1. We can see that the covariance-based probabilistic tracking approach significantly outperforms the other two techniques for a comparative computational effort.

	A	B	C
Score	0.39	0.59	0.67
Speed (ms/frame)	16	150	18

Table 1: A: color-based particle filter tracker. B: covariance-based tracker [2]. C: covariance-based probabilistic tracker.

Our proposed covariance-based probabilistic tracking approach is much faster than the conventional covariance-based approach which uses the exhaustive searching scheme to detect the target. When the number of particle is set to 100, the computational time for each frame is about 18 ms on a P4 3.2GHz machine, while for the conventional covariance-based approach, the search takes about 600 ms for a 320×240 image and runs at 150 ms/frame when using a hierarchical search. We can see that our proposed tracker at least eight times faster than the conventional one.

6. Conclusions

Embedding the covariance-based tracker introduced in [2] within a probabilistic framework, we further improve the tracking robustness and speed. We achieve the robust tracking of target under temporary occlusions, as well as the tracking of object with background distraction.

The proposed probabilistic tracker is much more suitable for multi-target tracking which is our ongoing work. Due to the integral images used for fast calculation of covariance matrix, when tracking multi-object, the computational cost grows less than the linear of the tracked target number. When Covariance-based object detector [7] is used to initialize the targets, the computational cost would lower than the independent detector and tracker. This is because the detector shares the same base features (integral images) with the tracker. Furthermore, the boosted particle filter [9] can be used to improve the multi-object tracking performance.

ACKNOWLEDGEMENTS

This work is partially supported by the National Natural Science Foundation of China (Grant No. 60675003) and the National High Technology Research and Development Program of China (Grant No. 2006AA01Z315).

References

[1] O. Tuzel, F. Porikli, and P. Meer. *Region covariance: A fast descriptor for detection and classification*. In ECCV 2006.

[2] F. Porikli, O. Tuzel, and P. Meer. *Covariance tracking using model update based on Lie algebra*. In CVPR 2006.

[3] D. Comaniciu, V. Ramesh, and P. Meer. *Kernel-based object tracking*. In PAMI 2003.

[4] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. *Color-Based Probabilistic Tracking*. In ECCV 2002.

[5] S. T. Birchfield, and S. Rangarajan. *Spatiograms Versus Histograms for Region-Based Tracking*. In CVPR 2005.

[6] F. Porikli. *Integral histogram: A fast way to extract histograms in Cartesian spaces*. In CVPR 2005.

[7] O. Tuzel, F. Porikli, and P. Meer. *Human Detection via Classification on Riemannian Manifolds*. In CVPR 2007.

[8] J. Vermaak, N. Lawrence, and P. Perez. *Variational inference for visual tracking*. In CVPR 2003.

[9] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. *A boosted particle filter: Multitarget detection and tracking*. In ECCV 2004.