Kernel-Based Method for Tracking Objects with Rotation and Translation

Haihong Zhang, Zhiyong Huang
School of Computing
National University of Singapore
{zhangha1,huangzy}@comp.nus.edu.sg

Weimin Huang, Liyuan Li
Insititute for Infocomm Research
Singapore 119613
{wmhuang,lyli}@i2r.a-star.edu.sg

Abstract

This paper addresses the issue of tracking translation and rotation simultaneously. Starting with a kernel-based spatial-spectral model for object representation, we define an l_2-norm similarity measure between the target object and the observation, and derive a new formulation to the tracking of translational and rotational object. Based on the tracking formulation, an iterative procedure is proposed. We also develop an adaptive kernel model to cope with varying appearance. Experimental results are presented for both synthetic data and real-world traffic video.

1. Introduction

Object tracking is a fundamental problem in machine vision [1], and it means to estimate the state of one or multiple objects as a set of observations (image sequences) become available on-line. In visual tracking, a key component is object representation which could describe the correlation between the appearance and the state of the object.

Many parametric statistical techniques have been applied to object representation, by exploiting essential statistics of the appearance of the object. In [2] a unimodal Gaussian was engaged to model a blob region of similar color. Furthermore, multimodal Gaussian using EM algorithm provides a way to model a blob with a mixture of colors [3, 4], but practically it may face the problem in choosing the right number of Gaussians.

Unlike parametric techniques, non-parametric techniques do not necessarily depend on a presumed distribution model of the object, thus they are more widely applicable. In the machine vision community, color histogram is a non-parametric approach that has been extensively exploited [5, p.47]. Another class of non-parametric technique called kernel density estimation has also become a very important data analysis tool [5, p.125]. Much research has been done on the theoretical properties of kernel estimators and the superiority over other estimators such as histograms is well-established. In [6], Comaniciu took advantages of spatial kernels, together with color histograms, to represent blob-alike color objects, and the representation led to an efficient mean-shift approach to tracking. Moreover, Collins extended the mean-shift tracker by using scale kernels to accurately capture the target's variation in scale [7]. Besides, Elgammal also proposed a color-blob model based on kernel density estimation for object segmentation [8].

However, another type of motion – rotation – has not yet been intensively studied in previous research using kernel density estimation techniques. In fact, rotation estimation is an important problem since rotation information can be very useful for both follow-up processing and video understanding. For example, an automatic face recognition system would require geometrically normalized face images, and such a geometric operation would depend on the face's accurate orientation.

This paper proposes a new method for tracking translation and rotation simultaneously. The method employs a kernel-based model for object representation, which offers more accurate spatial-spectral description than previous blob models. And by defining a l_2-norm similarity measure between the target model and the observation, we derive a new formulation to the tracking of translational and rotational object. From the tracking formulation, an iterative procedure is derived to estimate the object's state. To cope with inconsistent appearance of the target, this paper also presents an adaptive kernel model using an online maintenance mechanism.

We conduct a few experiments to examine the proposed tracking method. The results attest to the effectiveness of the proposed method.

2. Tracking Formulation

Generally, an visual object appears as a region consisting of N pixels $\{\mathbf{x}_i\}$ with colors $\{\mathbf{u}_i = \mathbf{u}(\mathbf{x}_i)\}$. Given a collection of samples, we represent the object by a joint spatial-

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

color kernel density

$$p(\mathbf{x},\mathbf{u}) = \alpha \sum_{i=1}^{N} k_s(||\mathbf{x}-\mathbf{x}_i||^2) k_u(||\mathbf{u}-\mathbf{u}_i)||^2) \quad (1)$$

where α is a normalization constant, k_s and k_u are kernel functions with bandwidth h_s and h_u.

Let's the object rotate around a reference point $\mathbf{x}_c$, usually the centroid of the object, by an angle θ. All the points on the object would be subject to the following transform T

$$\begin{aligned} x_i^{(r)} &= \hat{x}_i cos\theta - \hat{y}_i sin\theta + x_c \\ y_i^{(r)} &= \hat{x}_i sin\theta + \hat{y}_i cos\theta + y_c \end{aligned} \quad (2)$$

or written in vector form as: $\mathbf{x}_i^{(r)} = M_\theta \hat{\mathbf{x}}_i + \mathbf{x}_c$, where $\hat{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x}_c$ is the relative position of a pixel $\mathbf{x}_i$. M_θ is the rotation matrix determined by θ.

In tracking, we have a target model $p(\mathbf{x}^{(p)}, \mathbf{u}^{(p)})$ centered at origin ($\mathbf{x}_c^{(p)} = 0$), and an observation (usually a candidate region around the presumed position) $q(\mathbf{x}^{(q)}, \mathbf{u}^{(q)})$ in the current frame. Tracking a rotating object amounts to estimating the position $\mathbf{x}_0$ and the pose θ that maximize the similarity between p_θ and q, where p_θ is the kernel density representation of the target after translation and rotation by $\{\mathbf{x}_0, \theta\}$. Here the similarity distance between representations is defined in l_2 metric space

$$\begin{aligned} D(p_\theta, q) &= \int ||p_\theta - q||^2 d\mathbf{u} d\mathbf{x} \quad (3) \\ &= \int p_\theta p_\theta d\mathbf{u} d\mathbf{x} + \int qq d\mathbf{u} d\mathbf{x} - 2\int p_\theta q d\mathbf{u} d\mathbf{x} \end{aligned}$$

It can be shown that for given p and q, the first two terms on the right hand are constants. With the general relation

$$\begin{aligned} & \exp(-\frac{1}{2\sigma^2}(\xi-\xi_1)^2)\exp(-\frac{1}{2\sigma^2}(\xi-\xi_2)^2) \quad (4) \\ = \; & \exp(-\frac{2}{2\sigma^2}(\xi - \frac{\xi_1+\xi_2}{2})^2)\exp(-\frac{1}{2\sigma^2}\frac{(\xi_1-\xi_2)^2}{2}) \end{aligned}$$

we can calculate the integral for the last item and rewrite the distance measure as

$$D(p_\theta, q) \propto \sum_{i,j} k_s(\frac{||M_\theta \mathbf{x}_i^{(p)} + \mathbf{x}_0 - \mathbf{x}_j^{(q)}||^2}{2}) k_u^{(p_i q_j)} \quad (5)$$

where i or j denotes a sample from object O_p or candidate O_q, and $k_u^{(p_i q_j)}$ represents $k_u(\frac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(q)}||^2}{2})$. Thus, the tracking can be formulated as searching for

$$\{\hat{\mathbf{x}}_0, \hat{\theta}\} = \text{argmin}_{\{\mathbf{x}_0,\theta\}} D(p_\theta, q) \quad (6)$$

3. Iterative Tracking Algorithm

Due to the continuity of kernel functions in Eq. 5, the goal of tracking (Eq. 6) means $\nabla D(p,q) = 0$, which implies both $\nabla_{\mathbf{x}_0} D(p,q) = 0$ and $\nabla_\theta D(p,q) = 0$. After some manipulations, there is

$$\begin{aligned} \nabla_{\mathbf{x}_0} D(p,q) &\propto \nabla_{\mathbf{x}_0} \textstyle\sum_{i,j} k_s(\frac{||M_\theta \mathbf{x}_i^{(p)} + \mathbf{x}_0 - \mathbf{x}_j^{(q)}||^2}{2}) k_u^{(p_i q_j)} \\ &\propto \sum_{i,j} w_{ij}(\mathbf{x}_0 + M_\theta \mathbf{x}_i^{(p)} - \mathbf{x}_j^{(q)}) \quad (7) \end{aligned}$$

where the weight w_{ij} is given by

$$w_{ij} = -2\; g_s(\frac{||M_\theta \mathbf{x}_i^{(p)} + \mathbf{x}_0 - \mathbf{x}_j^{(q)}||^2}{2}) k_u^{(p_i q_j)} \quad (8)$$

with g_s the derivative of k_s.

Set $\nabla_{\mathbf{x}_0} D(p,q) = 0$ and we obtain the following solution

$$\mathbf{x}_0 = \frac{\sum_{i,j} w_{ij}(\mathbf{x}_j^{(q)} - M_\theta \mathbf{x}_i^{(p)})}{\sum_{i,j} w_{ij}} \quad (9)$$

Since $\mathbf{x}_0$ is embedded in weights $\{w_{ij}\}$ on the right side, the equation implies an iterative solution.

To study the partial derivative of $D(p,q)$ with respect to θ, we denote

$$\hat{\mathbf{x}}_j^{(q)} = \mathbf{x}_j^{(q)} - \mathbf{x}_0, \;\; \mathbf{x}_i^{(r)} = M_\theta \mathbf{x}_i^{(p)} \quad (10)$$

then we have

$$\begin{aligned} f_{ij} &\triangleq ||\mathbf{x}_i^{(r)} - \hat{\mathbf{x}}_j^{(q)}||^2 \\ &= (x_i^{(r)} - \hat{x}_j^{(q)})^2 + (y_i^{(r)} - \hat{y}_j^{(q)})^2 \\ &= sin\theta(2y_i^{(p)}\hat{x}_j^{(q)} - 2x_i^{(p)}\hat{y}_j^{(q)}) + \\ &\quad cos\theta(-2x_i^{(p)}\hat{x}_j^{(q)} - 2y_i^{(p)}\hat{y}_j^{(q)}) + c_{ij} \\ &\triangleq a_{ij} sin\theta + b_{ij} cos\theta + c_{ij} \quad (11) \end{aligned}$$

where c_{ij} a variable independent upon θ.

After some manipulations, we derive

$$\begin{aligned} \nabla_\theta D(p,q) &\propto \sum_{ij} w_{ij} \frac{\partial f_{ij}}{\partial \theta} \\ &\propto \beta_1 cos\theta + \beta_2 sin\theta \quad (12) \end{aligned}$$

where

$$\beta_1 = \sum_{i,j} w_{ij} a_{ij} \;,\; \beta_2 = \sum_{i,j} -w_{ij} b_{ij} \quad (13)$$

and w_{ij} is as given in Eq. 8. We see that setting $\nabla_\theta D(p,q) = 0$ amounts to setting $\beta_1 cos\theta + \beta_2 sin\theta = 0$. This can be easily solved by

$$\theta = -sin^{-1}(\frac{\beta_1}{\sqrt{\beta_1^2 + \beta_2^2}}) \quad (14)$$

Hence, we can use Equations (9,14) together with Eq. 5 to simultaneously estimate the position and the pose angle of the object by an iterative program. In a simplified form, the tracking algorithm is presented below.

1. Given the target model $p(\mathbf{x}, \mathbf{u})$ with $\{\mathbf{x}_i^{(p)}, \mathbf{u}_i^{(p)}\}$, as well as its location $\mathbf{x}_0$ and pose θ in the previous frame.
2. Initialize the location and pose of the target in the current frame. At present we just use their previous state;
3. Extract a candidate region ($\{\mathbf{x}_j^{(q)}, \mathbf{u}_j^{(q)}\}$) containing the object at the presumed location and pose;
4. Update $\mathbf{x}_0$ according to Eq. 9, let the change be $d(\mathbf{x}_0)$;
5. Update θ according to Eq. 14, let the change be $d(\theta_0)$;
6. If $d(\mathbf{x}_0) > \epsilon_x$ or $d(\theta_0) > \epsilon_\theta$, go to Step 3;
7. Proceed to next frame, go to Step 2.

4. Adaptive Kernel Density Model

In many real world videos, the object of interest may show continuous variations in appearance caused by e.g. out-of-plane rotation or non-rigid deformation. This would pose a problem to target modeling ellaborated earlier. To solve the problem, this section introduces an adaptive target model based on an on-line maintenance mechanism.

First, we define a probabilistic transient appearance model that consists of three elements: R_t, $l_t(\mathbf{x})$ and $C_t(\mathbf{x})$. R_t is the region of the pixels $\mathbf{x}$ that might belong to the target, where l_t is the likelihood of a pixel being from the target, and C_t is the color of a pixel. Then, according to the object's transformation T at frame t, the appearance of each pixel in the target model would be $\{T(\mathbf{x}), \mathbf{u}_t(T(\mathbf{x})\}$. Then the appearance model would be updated by

$$\begin{aligned} l_t(\mathbf{x}) &= l_{t-1}(\mathbf{x}) + p_t(T(\mathbf{x}), \mathbf{u}_t(T(\mathbf{x}))) \\ C_t(\mathbf{x}) &= (1-\alpha)C_{t-1}(\mathbf{x}) + \alpha \mathbf{u}_t(T(\mathbf{x})) \end{aligned} \quad (15)$$

where p_t is the kernel density (Eq. 1), α a parameter controlling the speed of color adaptation.

5. Evaluation with Synthetic Data

In this evaluation, we used a computer-generated diamond-shaped color object and a few 20-frame long test sequences, each sequence with a particular level of additive Gaussian noise. The object was moving and rotating stochastically at the speed of up to 10 pixels and 1/12 rad per frame. The Gaussian noise had a standard derivative σ that ranged from 50 to 200. Some cropped image samples are shown in Figure 1, where the corresponding noise level σ are 50, 80, 110, 140, 170 or 200 from left to right (pixel value in the range [0 255]).

Here we also examined two variants of our method (the proposed method is referred to as "Rotational Spatial-Color KD"). In one case, the variant tracker called "Spatial-Color KD" did not cope with rotation by setting $\theta \equiv 0$; in the other case, the tracker "Color KD" discarded spatial information in the object representation model (Eq. 1) by setting $k_s \equiv 1$, thus it may be comparable to blob trackers [6]. Figure 5 shows the comparative results.

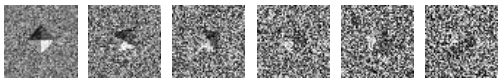


Figure 1. Synthetic images under various levels of noise corruption.

The results show that the proposed method could track both the object's position and pose angle accurately under high level (up to 170) of noise. It's also indicated that the method was more robust and could produce significantly smaller tracking error than "Color KD" or "Spatial-Color KD".

6. Experiment with Real World Video

The traffic video (Figure 3) were obtained from KOGS/IAKS Universität Karlsruhe [1]. The video was captured on a snowy day, where the target objects were cars turning at a cross. The low quality of images and the small size (up to 36×36) of cars pose a problem to accurate tracking of position as well as pose angle.

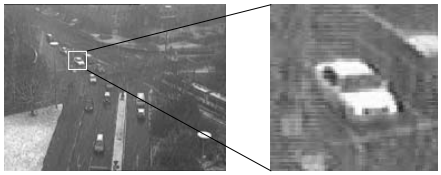
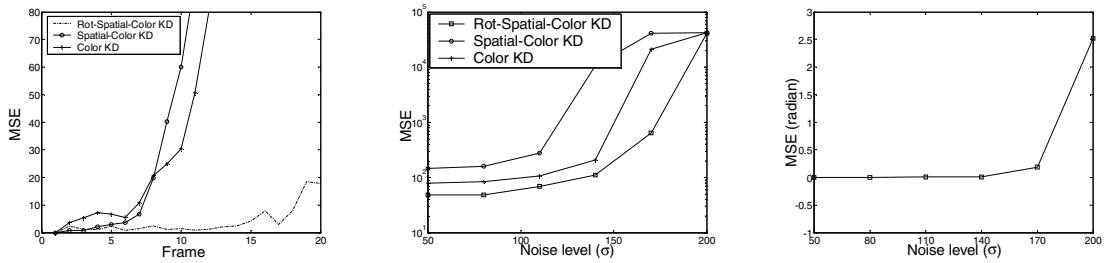


Figure 3. Real world car tracking image

The proposed tracking algorithm with adaptive kernel density model was applied to tracking the cars. Some results are shown in Figure 4, where the right panel plots the estimated trajectory with changing pose angle (denoted by arrows). It can be seen that, despite the out-of-plane rotation and low image quality, the model could well adapt to the changing appearance of the object, yielding accurate estimation of both translation and rotation.

At present, the initial target are set manually. To achieve full automation, a detection and segmentation algorithm could be used to locate and initialize the target model. Favourably, there has been much research on detection

1 http://i21www.ira.uka.de/image_sequences/



(a) Tracking translation at $\sigma = 170$ (b) Tracking translation over σ (c) Tracking pose angle over σ

Figure 2. Comparative results of tracking artificial objects

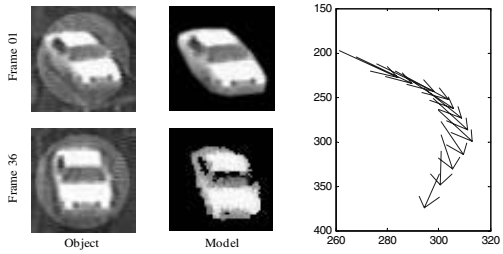


Figure 4. Results of Car Tracking

[9]. Recent work in [10] also offers an efficient detection-segmentation approach, which will hopefully be incorporated into our tracking system.

7. Computational Complexity

Because the algorithm is iterative and it usually takes a few iterations (usually less than 5) to converge, the overall computational complexity would mainly depend on the cost of each iteration. In particular, computing translation vector by Eq. 9 requires $O(N_p N_q)$ computational time ($N_p(N_q)$ is the number of target(candidate) pixels). And, for computing rotation angle according to Eq. 14, the cost is also approximately given by $O(N_p N_q)$. Hence, we conclude that the overall computational cost for each iteration is given by $O(N_p N_q)$.

Besides, an effective method to reduce the computational complexity is to use an alternative kernel function such as Epanechnikov kernel which is much less computationally expensive than Gaussian functions.

In a real implementation (with Matlab and Visual C++, unoptimized codes) on a conventional 1.3GHz Pentium-M PC, the system achieved a frame rate of 4fps for tracking a target object of 645 pixels (RGB color space).

8. Conclusion

The purpose of this work was to propose a new method for tracking translational and rotational objects. We have derived a formulation to the translation-rotation tracking, by using a kernel-based spatial-color model for object representation. With this formulation, we have developed an iterative procedure to tracking. Furthermore, an adaptive kernel model has been proposed to cope with varying target appearance during tracking. Experimental results attest to the effectiveness of the proposed method.

References

[1] Y. Bar-Shalom and T. Fortmann, Tracking and Data Association, Aademic Press, 1988.

[2] C.R. Wern, A. Azarbayejani, T. Darrell, and A.P. Pentland, Pfinder: Real-time tracking of human body, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, pp. 780–785, July 1997.

[3] Y. Raja, S.J. Mckenna, and S. Gong, Colour model selection and adaptation in dynamic scenes, in European Conference of Computer Vision, Vol. 1, pp. 460-474, 1998.

[4] Y. Raja, S.J. Mckenna, and S. Gong, Segmentation and Tracking Using Color Mixture Models, ACCV, pp. 607-614, 1998.

[5] D. W. Scott. Multivariate Density Estimation. Wiley, New York, 1992.

[6] D. Comaniciu, V. Ramesh and P. Meer. Real-Time Tracking of Non-Rigid Objects using Mean Shift, IEEE Proc. CVPR, Vol 2, pp.142-149, 2000.

[7] R. T. Collins. Mean-shift Blob Tracking through Scale Space. in IEEE CVPR, vol.2 pp. 234–40, 2003.

[8] A. Elgammal and R. Duraiswami and L. Davis. in IEEE ICCV, Vol 2, pp. 145–152, 2001.

[9] M.-H. Yang and N. Ahuja, Detecting Faces in Images: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, Jan. 2002.

[10] L.-Y. Li, W.-M. Huang, I.Y.H. Gu, Q. Tian, Foreground object detection in changing background based on color co-occurrence statistics, in Proc. 6th IEEE Workshop on Applications of Computer Vision, pp. 269 - 274, 2002.