Image cues fusion for contour tracking based on particle filter

P. Li, François Chaumette

Image Cues Fusion for Object Tracking Based on Particle Filter

Peihua Li and François Chaumette

IRISA/INRIA Rennes, Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
pli@irisa.fr

Abstract. Particle filter is a powerful algorithm to deal with non-linear and non-Gaussian tracking problems. However the algorithm relying only upon one image cue often fails in challenging scenarios. To overcome this, the paper first presents a color likelihood to capture color distribution of the object based on Bhattacharry coefficient, and a structure likelihood representing high level knowledge regarding the object. Together with the widely used edge likelihood, the paper further proposes a straightforward image cues fusion for object tracking in the framework of particle filter, under assumption that the visual measurement of each image cue is independent of each other. The experiments on real image sequences have shown that the method is effective, robust to illumination changes, pose variations and complex background.

1   Introduction

Probabilistic object tracking in image sequences has widespread applications in human-computer interaction, surveillance, visual servoing and biomedical image analysis. It has therefore been an active research topic in computer vision for over a decade.

Among probabilistic object tracking algorithms, particle filter has attracted considerable attention in recent years, because of its powerful ability to deal with general non-linear and non-Gaussian problems [1, 12]. In the framework of particle filter, one of the most important parts is the likelihood function (i.e., the measurement model). Some researchers are devoted to present different likelihoods for effective tracking, including those based on edge [1], or color [2, 3, 4]. Although particle filter has proven successful in dealing with object tracking, visual measurement dependent only on one image cue is not sufficient, and tracking failures often occur in complex scenarios [5, 6, 7]. Several factors can result in this consequence, such as significant illumination changes in the environment, pose variations of the object and non-linear deformations of shapes, in addition to noise and dense clutters in complex background.

Some researchers have recently made efforts to try to solve the problem. Wu et al. [6] present a novel particle filter, as an approximation of a factorized graphical model, in which shape and color samples are interactively drawn from each others' measurements based on importance sampling. The algorithm is

novel and tested on many real image sequences, but the effects are not so much satisfying. Pérez et al. [7] combine color information and motion information into tracker, in addition, they also consider the problem of multi-modality fusion, such as that of sound and image. While combining multiple information into particle filter, what one should consider carefully is how to integrate them in an effective way. Both Wu et al. and Pérez et al. employ methods similar to importance sampling introduced in [8]. Wu et al. first draw samples from color information based on importance sampling and evaluate shape probability. After that they make importance sampling on shape information and evaluate the color likelihood. Pérez et al. adopt similar method, either first drawing samples from color cue and then evaluating motion cue, or sampling from motion cue and then performing evaluation of color cues. This way, different image cues are applied to the algorithm sequentially, instead of simultaneously. While it may help improve efficiency, the inaccurateness, or, the most worst of all, failure of one cue, will heavily affect the others. In general, it is not desirable that evaluation of one cue will affect that of others.

We present in the paper a multiple cues fusion method in the framework of particle filter, in which all of the image cues are evaluated simultaneously on each discrete sample (particle). Three different likelihood functions, representing respectively three different image cues, are considered in the paper. We assume that the measurement of each image cue is independent of each other, and they are integrated to contribute to the overall measurement density. This way, different image cues are fused simultaneously and the failure of one image cue will not affect the evaluation of the other cues. The image cues we are interested in are edge information, color distribution and/or structural information of a specific class of objects, e.g., human faces. Note that our method of image cues integration is similar to that introduced in [9]. There exist, however, great differences between theirs and ours. In [9], Spengler et. directly extend the approach introduced in [10] from single hypothesis to multiple hypotheses, in which the image cues are evaluated on the whole image map (so they will have to confine their algorithm in the small image, say, 90*72, just like in [10]), and then different cues are weighted and added. In contrast, our image cues are only evaluated on particles. In addition, the modelling of image cues are also quite different.

The remainder of the paper is structured as follows. Section 2 introduces the generative model for object tracking, and then presents the three likelihood functions involved in the algorithm. Section 3 describes the particle filter based tracking algorithm. Section 4 makes experiments to validate the algorithm. Section 5 contains concluding marks.

2   Generative Model for Object Tracking

2.1   Shape Model

Following that described in [11], the contour is parameterized as a B-spline curve, for a set of B-spline basis is general and flexible in representing different

(complex) shapes, and in controlling the degree of continuity. Specifically, the tracked objects are modelled as follows

$$\mathbf{r}(s,t) = \begin{bmatrix} x(s,t) \\ y(s,t) \end{bmatrix} = \begin{bmatrix} \mathbf{B}(s)^{\mathrm{T}} & 0 \\ 0 & \mathbf{B}(s)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{Q}^x(t) \\ \mathbf{Q}^y(t) \end{bmatrix} \tag{1}$$

where $\mathbf{B}(s) = [b_0(s) \quad \cdots \quad b_{J-1}]^T$, for $0 \leq s \leq S$, $b_i(s)(0 \leq i \leq J-1)$ is the ith B-spline basis function, $\mathbf{Q}^x$ is a column vector whose unit consists of x coordinates of all the control points and similarly with $\mathbf{Q}^y$ (the time index t is omitted hereafter for simplicity), and L is the number of spans. The configuration of the spline is restricted to a shape-space of vectors $\mathbf{X}$ defined by

$$\begin{bmatrix} \mathbf{Q}^x \\ \mathbf{Q}^y \end{bmatrix} = \mathbf{W}\mathbf{X} + \begin{bmatrix} \bar{\mathbf{Q}}^x \\ \bar{\mathbf{Q}}^y \end{bmatrix} \tag{2}$$

where $\mathbf{W}$ is a shape matrix whose rank is less than $2J$, and $\bar{\mathbf{Q}} = \begin{bmatrix} \bar{\mathbf{Q}}^x & \bar{\mathbf{Q}}^y \end{bmatrix}^T$ is a template of the object. Below are two kinds of shape spaces used in the paper, the first for head tracking and the second for person tracking

$$\mathbf{W} = \begin{bmatrix} \mathbf{1} \, \mathbf{0} \, \bar{\mathbf{Q}}^x \\ \mathbf{0} \, \mathbf{1} \, \bar{\mathbf{Q}}^y \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \mathbf{1} \, \mathbf{0} \, \bar{\mathbf{Q}}^x \, -\bar{\mathbf{Q}}^y \\ \mathbf{0} \, \mathbf{1} \, \bar{\mathbf{Q}}^y \, \bar{\mathbf{Q}}^x \end{bmatrix} \tag{3}$$

2.2   Dynamical Model

The motion equation of the state in the shape space is modelled as the multi-dimensional second order auto-regression (AR) process, which generally can be seen as the discretized form of a continuous stochastic second order dynamic system [11]. This multi-dimensional AR process may be regarded as the direct extension of a 1D case. Define

$$a_1 = -\exp(-2\beta\tau)$$
$$a_0 = 2\exp(-\beta\tau)\cos(\omega\tau)$$
$$b_0 = \sqrt{1 - a_1^2 - a_0^2 - 2\frac{a_1 a_0^2}{1 - a_1}}$$

Define also the damping coefficient β, the oscillation period ω, and the sampling period of the system τ. Then the 1D AR process has the following form:

$$x_k = a_1 x_{k-1} + a_0 x_{k-2} + b_0 \nu \tag{4}$$

where ν is one dimensional Gaussian i.i.d. noise. It is desirable, in practice, to model the translation and the shape variations of the contour separately, so the 1D AR process is extended respectively to two complementary subspaces of the shape space: translation subspace and deformation subspace. Then the multi-dimensional motion model can be represented as below

$$\mathbf{X}_k = \mathbf{A}_1 \mathbf{X}_{k-1} + \mathbf{A}_0 \mathbf{X}_{k-2} + \mathbf{B}_0 \mathcal{V} \tag{5}$$

2.3   Observation Model

The observation model $p(\mathbf{Y}_k|\mathbf{X}_k)$ concerns in the paper three kinds of image cues: edge information of the contour, weighted color histogram and structural information outputted from boosted detector, which are represented respectively, $p_c(\mathbf{Y}_k|\mathbf{X}_k)$, $p_s(\mathbf{Y}_k|\mathbf{X}_k)$ and $p_e(\mathbf{Y}_k|\mathbf{X}_k)$. Given the predicted target state, we assume that the observation procedure of the three cues are independent of one another, so the overall measurement density has the following form

$$p(\mathbf{Y}_k|\mathbf{X}_k) = p_c(\mathbf{Y}_k|\mathbf{X}_k)p_s(\mathbf{Y}_k|\mathbf{X}_k)p_e(\mathbf{Y}_k|\mathbf{X}_k) \tag{6}$$

In practice, the log-likelihood of Equ. (6) is evaluated, and the multiplications thus become sum on the right side of the above equation. It can be seen that if one cue fails, its contribution to the overall measurement density becomes negligible.

Color Likelihood We define a likelihood to measure the confidence of the color similarity of a candidate to the target, which is based on the metric introduced in [13]. In the likelihood function, both target and candidate distribution are represented by weighted multi-channel color histogram: $\hat{\mathbf{q}} = \{\hat{q}_u\}$ with $\sum_{u=1}^{N_c} \hat{q}_u = 1$ for target, and $\hat{\mathbf{p}}(\mathbf{x}) = \{\hat{p}_u(\mathbf{x})\}$ with $\sum_{u=1}^{N_c} p_u = 1$ for the candidate, where $\mathbf{x}$ is the center of the candidate ellipse, and $u = 1, \ldots, N_c$ denote the bins of the histogram. Denote $\mathbf{z}_i \; i = 1, \ldots, M$ the pixel locations of one candidate (the ellipse which best fits the discrete sample), and $\mathcal{K}(\cdot)$ the weighting function that has the following form

$$\mathcal{K}(\mathbf{z}) = \begin{cases} c(1 - \|\mathbf{z}\|^2) & \text{if } \|\mathbf{z}\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The value of each weighted histogram bin u for the candidate distribution can be expressed as

$$\hat{p}_u(\mathbf{x}) = \frac{\sum_{i=1}^{M} \mathcal{K}(\|\frac{\mathbf{x}-\mathbf{z}_i}{\mathbf{h}}\|^2)\delta(b(\mathbf{z}_i) - u)}{\sum_{i=1}^{M} \mathcal{K}(\|\frac{\mathbf{x}-\mathbf{z}_i}{\mathbf{h}}\|^2)} \tag{8}$$

where $\mathbf{h}$ is the radius of a candidate region, $b(\mathbf{z}_i)$ is a function which associates to the pixel at location $\mathbf{z}_i$ the index $b(\mathbf{z}_i)$ of the histogram, and $\delta(\cdot)$ is the Kronecker delta function. The metric to measure the similarity of the target and candidate is

$$d(\hat{\mathbf{q}}, \hat{\mathbf{p}}(\mathbf{x})) = \sqrt{1 - \rho(\hat{\mathbf{q}}, \hat{\mathbf{p}}(\mathbf{x}))} \tag{9}$$

where $\rho(\hat{\mathbf{q}}, \hat{\mathbf{p}}(\mathbf{x}))$ is the Bhattacharyya coefficient that has the following form

$$\rho(\hat{\mathbf{q}}, \hat{\mathbf{p}}(\mathbf{x})) = \Sigma_{u=1}^{N_c} \sqrt{\hat{p}_u(\mathbf{x})}\sqrt{\hat{q}_u} \tag{10}$$

Upon this, we define the corresponding likelihood as follows

$$p_c(\mathbf{Y}_k|\mathbf{X}_k) = \frac{1}{\sqrt{2\pi}\sigma_c} \exp -\frac{1 - \rho(\hat{\mathbf{q}}, \hat{\mathbf{p}}(\mathbf{x}_k))}{2\sigma_c^2} \tag{11}$$

In practice we get the ellipse that best fits the contour of each particle, on which the likelihood is evaluated. The target distribution is achieved via a face detector when for head tracking. Otherwise, it is manually set.

Structure Likelihood The structure likelihood is aimed at representing the high level knowledge of the object and is achieved via machine learning algorithm. In the paper we are interested in boosted multi-view face detector [14, 15]. A total of five different views are considered in the paper: frontal view, left and half-left profiles, and right and half-right profiles. For efficiency, two levels are involved: the first level concerns a cascaded face detector trained on all training examples, which contains non-face examples and face examples in five different views. The test image region which only pass the first level will continue to try to pass the second level. Five different cascaded detectors are in the second level which are responsible for detections of faces which may be in different views.

A cascaded detector implicitly assumes a certain form for the underlying probability distribution [16]. Define N_s the total number of layers in the detector, and $1, \cdots, n_s$ the layers the detection process passed, in which the output is above the relevant threshold for the input from the test rectangle corresponding to the particle. In our implementation, we assume for simplicity that the likelihood of the particle is related to n_s/N_s. More precisely, we define the structure likelihood as

$$p_s(\mathbf{Y}_k|\mathbf{X}_k) = \frac{1}{\sqrt{2\pi}\sigma_s} \exp -\frac{1 - n_s/N_s}{2\sigma_s^2} \qquad (12)$$

The face detection is performed within the circumscribed rectangle of the minimal area for each particle. Because of the level and the cascade structure, the fast computation of features used in the detector, and the search being constrained in a small image rectangle, the evaluation of the likelihood is efficient. Furthermore, when the face is not present, most of regions will fail to pass the first level. This further reduces the computational load.

Edge Likelihood The model introduced by MacCormick is adopted for the observation of edges [12]. The measurement as regards edge is made at a finite number of points along the contours modelled as B-spline curve, and the normals to the contour at these sample points are searched for features. These normals have fixed length L and are termed measurement lines. A Canny edge detector is applied to each measurement line and the points of local maximum adopted as features. In general, a set of n_l features are detected on the measurement line indexed by $l = 1, \ldots, N_e$. The distances of these features from the contour constitute a set $z_j^{(l)}, j = 1, \ldots, n_l$. Each feature at distance $z_j^{(l)}$ from the contour could correspond to the true boundary of the object (in which case it is called an edge feature) or random visual clutter (in which case it is called a clutter feature).

We assume that only one edge feature can be detected on the measurement line. To model the observation density, some further assumptions are made

– The number of clutter features on the measurement lines of length L obeys a Poisson law with density λ.
– The density of the clutter features is uniform on the measurement line.
– The probability that the edge feature is not detected is P_0 and the probability that it is detected is $P_1 = 1 - P_0$.
– The distribution of the distance between the edge feature and the true contour location is Gaussian, with zero mean and variance σ_e^2.

From these assumptions, we obtain the following equation for the likelihood of the observation at a sample point, given the state $\mathbf{X}_k$:

$$p(l|\mathbf{X}_k) \propto P_0 + \frac{1 - P_0}{\lambda} \sum_{j=1}^{n_l} \frac{1}{\sqrt{2\pi}\sigma_e} \exp - \frac{(z_j^l)^2}{2\sigma_e^2} \tag{13}$$

Assuming that the feature outputs on distinct normal line are statistically independent, the overall edge likelihood becomes

$$p_e(\mathbf{Y}_k|\mathbf{X}_k) = \prod_{l=1}^{N_e} p(l|\mathbf{X}_k) \tag{14}$$

3   Image Cues Fusion for Contour Tracking Based on Particle Filter

Target tracking can be characterized as the problem of estimating the state $\mathbf{X}_k$ of a system at (discrete) time k, as a set of observations $\mathbf{Y}_k$ become available over time. The Bayesian filtering framework is based on the densities $p(\mathbf{X}_k|\mathbf{X}_{k-1})$ and $p(\mathbf{Y}_k|\mathbf{X}_k)$. The transition prior $p(\mathbf{X}_k|\mathbf{X}_{k-1})$ indicates that the evolution of the state is a Markov process, and $p(\mathbf{Y}_k|\mathbf{X}_k)$ denotes the observation density (likelihood function) in the dynamical system, in which the measurements are conditionally independent of each other given the states. The aim is to estimate recursively in time the filtering density $p(\mathbf{X}_k|\mathbf{Y}_{1:k})$, where $\mathbf{Y}_{1:k}$ denotes measurements from the beginning to the current time step k, which is described as follows:

$$p(\mathbf{X}_k|\mathbf{Y}_{1:k}) = \frac{p(\mathbf{Y}_k|\mathbf{X}_k)p(\mathbf{X}_k|\mathbf{Y}_{1:k-1})}{p(\mathbf{Y}_k|\mathbf{Y}_{1:k-1})} \tag{15}$$

where the prediction density $p(\mathbf{X}_k|\mathbf{Y}_{1:k-1})$ is

$$p(\mathbf{X}_k|\mathbf{Y}_{1:k-1}) = \int p(\mathbf{X}_k|\mathbf{X}_{k-1})p(\mathbf{X}_{k-1}|\mathbf{Y}_{1:k-1})d\mathbf{X}_{k-1} \tag{16}$$

Eq. (15) provides an optimal solution of the tracking problem, which, unfortunately, involves high-dimensional integration. In most cases involving non-Gaussianity and nonlinearity, analytical solutions do not exist, leading to the use of Monte Carlo methods.

3.1 Tracking Algorithm Based on Particle Filter

The basic principle of particle filtering (also known as the Sequential Monte Carlo algorithm) is that the posterior density is approximated by a set of discrete samples (called particles) with associated weights. For each discrete time step, particle filtering generally involves three steps for sampling and weighting the particles, plus one output step. In the sampling step, particles are drawn from the transition prior. In the weighting step, particle weights are set equal to the measurement likelihood. The outputs of the filter are the particle states and weights, used as an approximation to the probability density in state space. In the last step, particles are re-sampled, to obtain a uniform weight distribution. The detailed algorithm is presented in Fig. **??**.

4 Experiments

The program is implemented with C++ on a laptop with 2.0GHz mobile Pentium CPU and 256M Memory. Table **??** summarizes parameters used in the paper. The standard variances of color and structure likelihoods are set empirically. It doesn't need to tune them very carefully: once they are set, they are favorable to all of our experiments. Unless indicated explicitly, the image sequence for head tracking is of size 256×192, and for person tracking is of size 640×240. The particles whose probability are greater than 0.1 and the mean as the tracking result, are plotted, with dashed blue color and solid red color respectively.

Fig. **??** demonstrates head tracking exploiting only edge information in a typical office environment. The tracking results are satisfying when strong edge can be observed, but it fails when edge information becomes weak.

Particle filtering based tracking dependent merely on color likelihood works well, even when illumination changes, but not significantly, as show in Fig. **??**. But the algorithm collapses while encountered with significant lighting variations. Fig. **??** shows the relevant results.

Fig. **??** presents the result with algorithm which relies solely on structure information. The algorithm can successfully track face which undergoes pose variations and a small in-plane rotation, but will fail when the subject turns back.

We test our tracking algorithm fusing multiple image cues on two image sequences. The first image sequence concerns head tracking, which involves considerable lighting changes, distraction similar to skin color, agile motion of the object and complex background. Tracking depending only on one image cue or two image cues simultaneously will fail without exception. The tracking algorithm , running at about 10Hz, can well localize the target throughout the whole image sequence, making use of the three image cues simultaneously. Some tracking results are shown in Fig. **??**. Note that from frame 29 to 80 there occurs considerable lighting change, which affects heavily both the color and edge of the object, as such importance sampling depending either will fail in this case.

The second image sequence, recorded with a wide-len camera, is concerned with a person walking in front of the shop window [17]. The target appears at one

end and walk to the other end. The following factors existed that make tracking difficult in this case: the reflections from the ground and from the opposing window; occlusions from the text on the window; complex non-linear deformation due to the walking behavior and the wide-len; the similarity of the color between the subject's clothes and the background. Tracking would have been more simple by subtracting the background image from the image sequence since the camera is fixed. We do not make use of this advantage, in order to test our algorithm in this complex scenario. The tracking results as illustrated in Fig. **??** are satisfactory, thanks to the algorithm (running at about 9Hz) that fuses of edge and color information. Notice that in the vicinity of frame 95 the tracker are heavily disturbed by the clutter in the background, but it succeeds to overcome it several frames later.

5   Conclusions

In the paper, a tracking algorithm based on particle filter is presented which integrates multiple image cues in a probabilistic way. We first presents a color likelihood to capture color distribution of the object based on Bhattacharry coefficient, and a structure likelihood to represent high level knowledge regarding the object based on AdaBoost learning. We also consider a widely used edge likelihood. Then, under the assumption that the measurement processes related to the above three likelihoods are independent of one another, they are combined to contribute to an overall observation density. The experiments show that, in challenging environment, particle filter based tracking algorithms making use of only one image often fails. With fusion of multiple image cues, the tracker becomes much more robust to significant lighting changes, pose variations and complex background. The paper also demonstrates that the high level knowledge itself, through a likelihood built upon a boosted multi-view face detector, can be used for object tracking. It is straightforward to extend this to tracking of other class of objects, such as pedestrians and cars, which will be our future work.

References

[1] Isard, M., Blake, A.: Contour Tracking by Stochastic Propagation of Conditional Density. Proc. Eur. Conf. on Comp. Vis. Cambridge UK (1996) 343–356. 99

[2] Nummiaro, K., Koller-Meier, E.,Van Gool, L.: An Adaptive Color-based Particle Filter. Image and Vision Computing **21**(1)(2003) 99–110. 99

[3] Vermaak, J. , Pérez, P., Gangnet, M., Blake, A.: Towards Improved Observation Models for Visual Tracking: Selective Adaptation. Proc. Eur. Conf. on Comp. Vis. Copenhagen Denmark (2002). 99

[4] Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based Probabilistic Tracking. Proc. Eur. Conf. on Comp. Vis. Copenhagen Denmark (2002). 99

[5] Vermaak, J., Gangnet, M., Blake, A., Pérez, P.: Sequential Monte Carlo Fusion of Sound and Vision for Speaker Tracking. IEEE Int'l Conf. Comp. Vis. (2001) 741–746. 99

[6] Wu, Y., Huang, T. S.:A Co-inference Approach to Robust Visual Tracking. IEEE Int'l Conf. Comp. Vis. Vancouver Canada (2001) 26–33. 99

[7] Pérez, P., Vermaak, J., Blake, A.: Data Fusion for Visual Tracking with Particles. Proceedings of IEEE (issue on State Estimation) (2004). 99, 100

[8] Isard, M., Blake, A.: ICondensation: Unifying Low-level and High-level Tracking in a Stochastic Framework. Proc. Eur. Conf. on Comp. Vis. Freiburg Germany (1998) 893–908. 100

[9] Spengler, M., Schiele, B.: Towards Robust Multi-Cue Integration for Visual Tracking. Machine Vision and Applications. **14**(1) (2003) 50–58. 100

[10] Triesch, J., von der Malsburg, C.: Democratic Integration: Self-Organized Integration of Adaptive Cues. Neural Computation **13**(9)(2001) 2049–2074. 100

[11] Blake, A., Isard, M.: Active contours. Springe-Verlag, Berlin Heidelberg (1998). 100, 101

[12] MacCormick, M.: Probabilistic Models and Stochastic Algorithms of Visual Tracking. PhD thesis, University of Oxford (2000). 99, 103

[13] Comaniciu, D., Ramesh, V., Meer, P.: Real-time Tracking of Non-rigid Objects Using Mean Shift. IEEE Int. Conf. on Comp. Vis. and Pat. Rec. Hilton Head Island South Carolina (2000) 142–149. 102

[14] Viola, P., Jones, M. J.: Robust Real-time Object Detection. IEEE Workshop on Statistical and Computational Theories of Vision. Vancouver Canda (2001). 103

[15] Li, S. Z., Zhu, L., Zhang, Z.,Blake, A., Zhang, H., Shum, H.: Statistical Learning of Multi-view Face Detection. Proc. Eur. Conf. on Comp. Vis. Copenhagen Denmark (2002). 103

[16] Friedman, J., Hastie, T., Tibshirani, R.:Additive Logistic Regression: A Statistical View of Boosting. The Annals of Statistics, **38**(2)(2000) 337-374. 103

[17] http://pets2002.visualsurveillance.org 105