Tracking Object Using Depth and Color Features

Cong Lin and Chi-Man Pun

University of Macau, Macau S.A.R, China
{yb17403,cmpun}@umac.mo

Abstract: In this paper, we proposed a simple tracking method for color and depth video stream based on traditional particle filter. Both the depth information and color information are used and integrated into a feature vector. Moreover, scale of change of the target object is estimated by change of depth. The proposed method is efficient to be a real-time tracker since it avoids expensive computation for scale as most in literature. The experiments were carried out on a video dataset captured by Kinect. The results have been satisfied and showed the estimated trace follows the target object very closely.

Keywords: Object tracking, particle filter, depth, scale of change, real-time tracker, Kinect.

1 Introduction

Since digital camera developed quickly in recent decades, visual object tracking has drawn lots of attention from researchers. The visual object tracking techniques could be integrated into many applications like video surveillance, gesture recognition and even augment reality. The aim of object tracking is to estimate the object location from consecutive frames and output the trajectory for further analysis.

The framework of tracking system generally includes three parts: 1) Tracking model; 2) Object features to track; 3) Ancillary machine learning approach for handling object variants. Some systems might be able to track multiple objects by using object detection methods as well. The simplest tracking models is KL tracker proposed by Lucas and Kanade (K-L tracker) [1]. The K-L tracker makes used of both spatial intensity gradient of the image and Newton-Raphson iteration technique to find the best match. The more frequently used are Kalman filter and particle filters which based on sequential Monte Carlo methods and Bayesian inference[2]. The filters consist of two stages: prediction and update. In the prediction stage, the state posterior probability density function (pdf) is predicted from a measurement from one frame to the next. After that, prediction pdf is updated with the latest measurement. The later proposed particle filters generally outperforms Kalman filters and many variants of particle filters emerged in recent years[3]. However, even though the particle filter is good tracking model, an obvious drawback of it is the computational

cost. The tracking accuracy improves as number of particles increases, making it costly in computation and difficult to be a real-time application. This is one of the problems that we attacked in this paper. Some Tracking models are independent to object features. The object features are vectors extracted from the target objects/candidate regions which could represents the characteristics of them. There are many object features in literature, e.g., Gabor [4], local binary pattern (LBP)[5], Haar-like feature [6, 7], color. The underlying principle of Gabor, Binary Pattern and haar-like features is to project object target or candidate patches on to a set of basis. The coefficient vectors are used in matching stage as feature vectors. In most cases, the color information is considered as statistical information and rendered as histogram. And distances for these statistical information are generally computed by the Bhattacharyya Coefficient [8]. During long term tracking, target object may change its appearance gradually, making it different from the original object. The occlusions, change of scale or rotation may lead to a mismatch. To enhance tracking performance and prevent target drift, some methods also incorporate training approaches, like On-line Random forest [9] and On-line boosting[10]. The training method help the tracking system 'remember' the course that how the object changes. Another advantage of using these training approaches is the detection could be easily implemented. With detection capability, if the object moved out of the scene, it could still be detected and tracked once it moves in again. Experiment proved training approaches improve the tracking result significantly. Moreover, if the process speed is fast enough, the 'tracking by detection' scheme is feasible.

In this paper, we proposed a particle based tracking method which used features from both depth and color information. Although many methods used color or texture as features and some of these in literature are successfully implemented, methods using depth information are relatively new. Thanks to latest development of image/video capturing technology, the depth map along with color map could be easily acquired with Kinect or PrimeSensor. The advantage is very obvious as the target object usually may have different depth from its background. The main contribution of our idea is to tried used depth information in two ways: 1) Auxiliary tracking feature since depth is kept stable in a short period generally; 2) As known by all that the object scale mapped to acquired images is linear to the distance from the camera, we could calculated change of scale of the target object by the variation of its depth. We implement it in our experiment and proved it works very robustly. The outline of this paper is as follows. In the first section, an introduction is given and literature is reviewed. Basic framework and tracking model is explained in section 2. In coming section 3, the proposed feature is introduced. Details of experiments and its results are presented in section 4. Finally, we made conclusion in last section.

2 Particle Filter

Particle filter is a Monte Carlo method that approximates the target density by a number of particles. Each particle is represented $^{by\{x'w'\}N}$ Particles are distributed around a possible area that the target possibly exists. x' is the state (in our

case is the location) of the particle i at time t while W is the weight for it, which represent how important the corresponding particle is and how similar this particle is to the target object. The particle filter model is given by:

$$p(x_r \quad z_{1''}) \; \mathrm{GC} \; p(z \; x,) E_{j=1}^{N} \; w \; . \; P(xt \qquad (1)$$

-,,..., N i

The w, Iwt 1 2 I is normalized vector and has a unity sum. The is updated each time by:

The q(x: z') q is called proposal distribution which draws N particles {x,}

Proposal distribution draws the particles from the state space. $P^{\prime z}t$) is the similarity of the i hypothesis and the target object ^{z}t. In our case, in order to simplify the tracking system, we set P

(x' x)= q(x Ix' z) . The formula (2) becomes:

$$w_r \; \mathrm{oc} \; w_t \; P\;(z, \qquad (3)$$

3 Feature Generation

Features both generated from color and depth map are used in matching the original object patch (target) and the candidates patches which is the bounding boxes that centering particles. The distances between target and particles are converted into weights for each particle. The location of object at time t is estimated by formula (3). On the other hand, a scale changing linear equation is computed on the specific physical property of the lens of camera. In our experiment which the Kinect is used, we have the following depth-to-scale relation:

$$Sc(t + 1) = Sc(t) + K * Adepth \qquad (4)$$

where Sc(t+1) and Sc(t) are the scales for the object at time t+1 and t respectively. K is the variation parameter which is set to -0.24 in this experiment for Kinect. A depth is depth value changed from the previous frame. Therefore, by measuring how the depth changes in a time, the tracker is able to adjust the length and width of the bounding box according. One of the advantages of this scheme is that it helps reducing computational cost owing that the tracker is no longer necessarily to

compute particles in different scales at a same time frame, which would increases multiple times of the cost.

Each frame will be considered an individual image. As the image is original captured in RGB color space, color distribution in RGB space is easily affected by illumination conditions which may change the feature vector dramatically and lead to tracking drift. Thus, we convert the image into HSV color space. Further, in order to combine 3 channels into one and simplify computation, we have following process for each pixel and generate a new map for a single frame. [11].

$$M = Ph \cdot Ps + Pv \qquad (5)$$

where P " ' P_s ' P v are the values of each channel in HSV space of a pixel respectively. For computing color distribution of each patch in bounding boxes (particles), feature vectors with 42 dimensions are generated by accumulating numbers of intensity value distributed on 42 equidistance levels. Finally, the feature vectors are normalized to unity sum.

The depth map (a frame of depth stream) also captured from camera may contain various noises. Many of them presented some part of the patches or image in zero values. Using mean value of the patch could cause inaccurate estimate of the depth of the object. We consider the median depth value as appropriate representation for a patch and the satisfied experimental result proved our choice.

The similarities are computed by Bhattacharyya Coefficient which is usually used in measuring distance/similarity of two probability distributions:

$$Sim(Tg,H:) = -\ln\left(1 - \sum_{J=1}^{M} \sqrt{VTg(j)H:(j)) * b(D: -} \right), \quad M = length(HI) \qquad (6)$$

where T_g is the color feature from target template while ^{rr} from i hypothesis at time t. ^{D} and $^{D}I^{-1}$ are depth of i hypothesis at time t and estimated depth of object at last frame. The length of $^{H}1$ is fixed and equal to 42. The weight vector ^{W,} is the normalized $\{Sim(Tg,H;)\}$

4 Experimental Results

The experiments were carried out on Matlab with a video clip with depth stream. Although the experiment is off-line, our method could be implemented in real-time application thanks to the efficiency of our algorithm. The process for each frame is completed within 0.03 second, making it capable to handle video stream with 30 frames per second. Owning to very few video dataset existed in literature, we created a simple dataset by ourselves. The dataset is captured by Kinect via OpenNI

toolbox. Fig. 1 shows our tracking result. The object template was manually picked up by setting a bounding box at the first frame. The tracker successful tracked down the face of a toy lion and the continuous frames show the scale of the bounding box smoothly changes accordingly as the object moves forward and backward. Even though there are parts of noise in depth stream (parts in dark blues are noise in 0 values), the trace of the object remained very stable and robust.

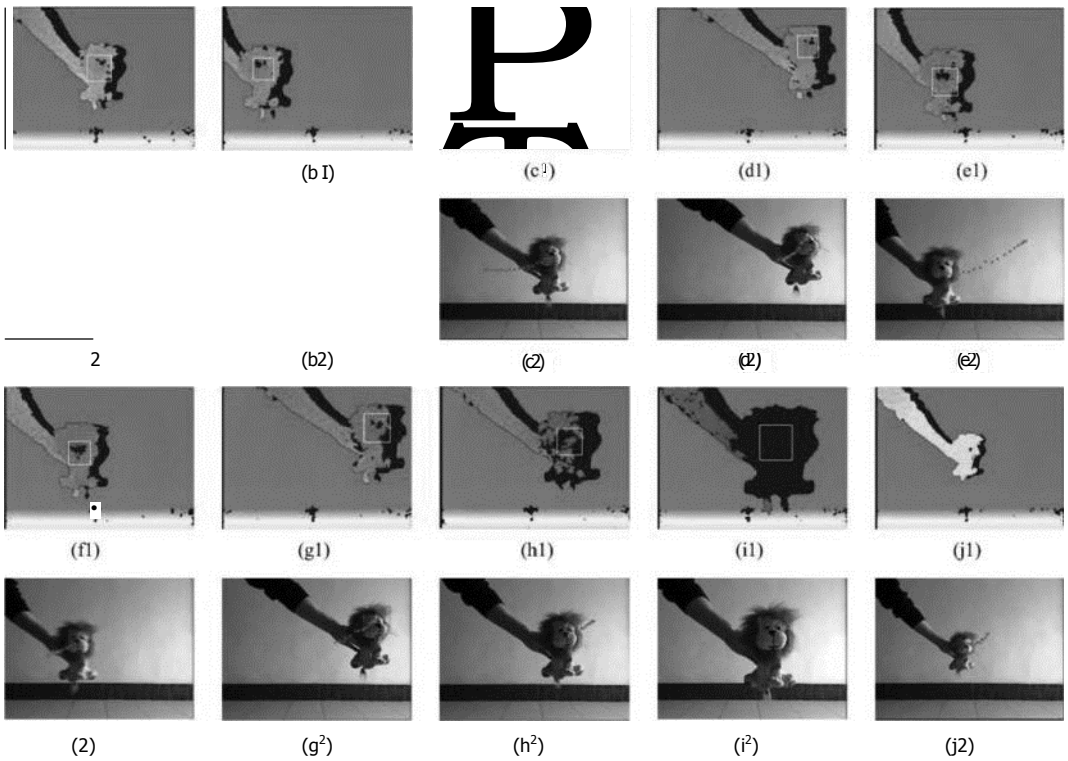

Fig. 1. (a)-(j) shows the snapshot of tracking result of a video clip with a interval of 25 frames. (al)-(fl) are depth maps while (a2)-(f2) are of RGB color space. Bounding boxes are in different color from the background. Pink dots are the traces of objects, i.e. , the preview estimated locations of the target.

5 Conclusion

By incorporating depth information into features and computing change of scale, we proposed a simple tracking method for color and depth video stream based on traditional particle filter. The proposed tracking system is very efficient and avoids expensive computation for scale as most in literature. Moreover, it is fast enough to be implemented as a real-time tracker. The experimental results have been satisfied and showed the estimated trace follows the target object very closely.

Reference

[1] B. D. Lucas, and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proceedings of the 7th international joint conference on Artificial intelligence Volume 2, Vancouver, BC, Canada, 1981.

[2] M. S. Arulampalam, S. Maskell, N. Gordon *et al.*, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174-188, 2002.

[3] R. Hwang Ryol, and M. Huber, "A particle filter approach for multi-target tracking."Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on. pp. 2753-2760, 2007.

[4] T. Feng, and T. Hai, "Non-orthogonal Binary Expansion of Gabor Filters with Applications in Object Tracking."Motion and Video Computing, 2007. WMVC '07. IEEE Workshop on. pp. 2424, 2007.

[5] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971-987, 2002.

[6] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features."Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. pp. 1-511-1-518 vol.1, 2001.

[7] R. Lienhart, and J. Maydt, "An extended set of Haar-like features for rapid object detection."Image Processing. 2002. Proceedings. 2002 International Conference on. pp. I-900-I903 vol.1, 2002.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift."Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. pp. 142-149 vol.2, 2000.

[9] A. Saffari, C. Leistner, J. Santner *et al.*, "On-line Random Forests."Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. pp. 1393-1400, 2009.

[10] C. Leistner, A. Saffari, P. M. Roth *et al.*, "On robustness of on-line boosting - a competitive study."Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. pp. 1362-1369, 2009.

[11] K. Okuma, A. Taleghani, N. de Freitas *et al.*, "A Boosted Particle Filter: Multitarget Detection and Tracking," *Computer Vision - ECCV 2004*, pp. 28-39, 2004.