Contemporary Engineering Sciences, Vol. 9, 2016, no. 11, 539 - 545
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2016.6439

Adaptive Cell-Size HoG Based Object Tracking with Particle Filter

Whui Kim

Electronics and Telecommunication Research Institute
218, Gajeong-ro, Yuseong-gu, Daejeon, 34129, Korea

University of Science and Technology
217, Gajeong-ro, Yuseong-gu, Daejeon, 34113, Korea

Kyong-Ho Kim*

Electronics and Telecommunication Research Institute
218, Gajeong-ro, Yuseong-gu, Daejeon, 34129, Korea
&
University of Science and Technology
217, Gajeong-ro, Yuseong-gu, Daejeon, 34113, Korea
*Corresponding author

Abstract

Visual object tracking is one of the most vigorous research area in Computer Vision. A lot of algorithms already achieve high performance and accuracy. Whereas, most of visual object tracking algorithms proceed separately from detection algorithm because of difference between tracking and detection descriptor. Instead, we propose adaptive cell-size HoG (acHoG) based Particle Filter Tracking (PFT) algorithm. Using HoG enables to share information with detection algorithm based on HoG and complements the fast but inaccurate PFT. However, according to the characteristics of PFT, features are extracted more than two times on the most of the target region. To solve repeated feature extraction problem, we apply adaptive cell-size to HoG. Because acHoG shares intensity and angle of edge extracted from original target image, repeated computations can be reduced and same size features extracted from different size targets. Experimental results have proved that the acHoG is effective.

Keywords: Particle Filter, Adaptive Cell-Size, Histograms of Oriented Gradients

1 Introduction

Visual object tracking is one of the most vibrant research area along with object detection, recognition and pose estimation etc. in computer vision. Recently, Lots of tracking algorithm are submitted and evaluated on Visual Object Tracking Challenge (VOT) has been held annually since 2013. Several algorithm already achieve the high performance and accuracy [1]. However, most of submitted algorithms of VOT focus on object tracking and don't consider detection. Therefore, descriptors of these algorithm are irrelevant to descriptors of detection algorithm. If tracking and detection are required at the same time under the above conditions, they should be proceeded individually. When the number of object to tracking is large under the above conditions, computation cost will be more increased.

In contrast to algorithms of VOT, statistical methods such as Kalman and Particle Filter are efficient and reasonable for multi-objects tracking because of uncomplicated and rapid calculation. Even though those algorithms was proposed on several decades ago, these have been used in various fields until now [2], [3]. Performance of Kalman Filter is better than other in the highly restrictive environment such as linear system but it has problem on non-linear system. In order to solve this problem, extended Kalman filter, approximate grid-based methods, and particle filters were proposed and developed [4]. However, accuracy of traditional Particle Filter is worse than algorithm of VOT, because it generally uses edge or edge features.

In this paper, we improve Particle Filter by being combined with HoG to be robust to illumination and slight deformation. However, according to sampling characteristic of Particle Filter, HoG features are extracted more than twice on more than half of target region. We propose an adaptive cell-size HoG (acHoG) algorithm to solve this problem. Because the acHoG algorithm shares extracted intensity and angle of edge, repeated edge computation of original HoG can be reduced.

The paper consists of 4 section. Section 2 introduce Adaptive cell-Size HoG and our particle filter algorithm. The experiment results and conclusions are given in section 3 and 4.

2 Particle Filter based object tracking with acHoG

HoG is effective to detect not only pedestrian but also vehicles, etc. Therefore, we imply HoG to PFT. However, fundamentally, there are a lot of computations to extract HoG features. Therefore, we proposed algorithm sharing computed HoG features with fixed 8x8 cell-size suggested by Dalal et al. [5], [6]. Even though our previous method reduce the computation cost, there are preciseness and positioning problem.

In this paper, we propose adaptive cell-size HoG (acHoG) algorithm to solve this problems. According to an object size, we adjust a cell-size to the object size. Adjusting cell-size enables us to extract same-size HoG feature from different size object. To calculate likelihood of particle filter, extract HoG features of particles image by acHoG algorithm and calculate correlation between reference and particles HoG features. With calculated likelihood, update the weights, predict next state and make samples (particles).

2.1. Particle Filter

When the object to track determined, particle samples are randomly initialized with distribution of standard deviations determined in accordance with size of target object. Weights of particle samples are initialized with the uniform distribution and calculated with histogram correlation after initialization. To use the previous weights information, we update weights with weighted sum of previous and evaluated likelihood. With this method, particle samples could reflect not only the latest but also previous information. Although the number of particle samples which reflect resent weights decreases, particle samples that reflect previous weights enable to stably predict next state on abruptly moving case.

A process noise is added on transition process. As figure 1, process noise enables samples of same state to be transited different state. Thus, the process noise reduce repeated computation.

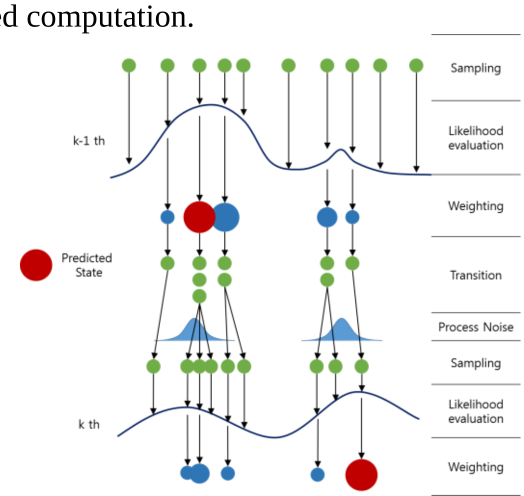

Fig. 1. Process of Particle Resample

2.2. The acHoG for Particle Filter

To calculate correlation between reference and particles HoG features by formula 1, each feature size should be same. When we fix cell-size, HoG features of different size particles are different without resizing particles. If we change cell-size per each particle, we can get same size HoG features from all particles and reuse computed gradients and angles extracted on the overall target region.

$$P(Z_{(k)}|X_{(k,i)} = \sum\{(H(X_{(k,i)},j)-_{MEAN}(H(X_{(k,i)})))_*(H_{ref}(j)-_{MEAN}(H_{ref}))\}/N_{term}$$
$$N_{term} = (\sum\{(H(X_{(k,i)},j)-_{MEAN}(H(X_{(k,i)})))\}^2{}_*\sum\{(H_{ref}(j)-_{MEAN}(H_{ref}))\}^2)^{1/2} \quad (1)$$

Reusing computed gradients and angles is considerably effective on PFT due to overlapped particles. Because all of particles are produced with Gaussian distribution, they are excessively close to each other. As the table 1, the bulk of particles are overlapped and acHoG is helpful to reduce repeated computations.

Sequence	n times HoG extraction region / overall target region (%)	
Name	One time	Two times ~
David	7.491	92.509
Car4	6.389	93.611
FaceOcc1	6.446	93.554
Dudek	15.180	84.820

Table 1. Ratio of regions on where HoG features extracted once and more than two times

Even if window sizes are different, feature-sizes are same by the acHoG. However, we need some restrictions such as fixed blocks per window and cells per block. When we suppose feature sizes are same, we can calculate cell-size by formula 3 derived from formula 2. For example on figure 2 (c), when we have two different size particles, we can get features of same size by changing cell size on restrictions of fixed blocks per window and fixed cells per block.

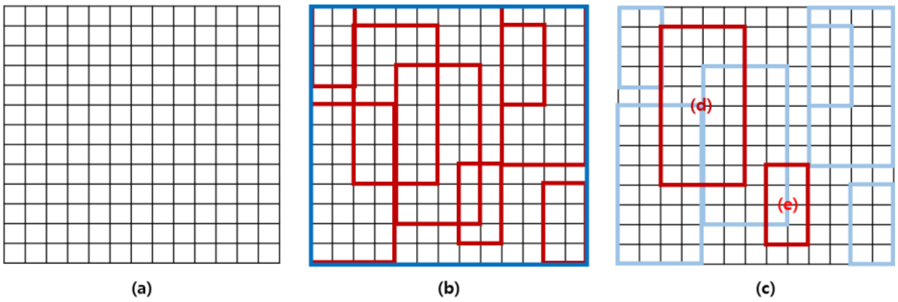

Fig. 2. (a) The size of grid is (8 x 8 pixel). There are computed gradients and angles of overall target region. (b) Red rectangles are Particle Filter Samples. Blue rectangle is overall target region. (c) If window, block, block stride, cell size of (d) are (32 x 64 pixel), (16 x 16 pixel), (8x8 pixel), (8x8 pixel) and size of (e) are (16x32 pixel), (8x8 pixel), (4x4 pixel), (4x4 pixel). Feature Sizes of (d) and (e) will be same 288 by Equation (2)

$$FestureSize = nbins_*((WinSize – BlockSize)/BlockStride + 1) \quad (2)$$

$$BlockStride = (WinSize – BlockSize)/(FestureSize/nbins - 1) \quad (3)$$

3 Experiment

To experiment, we used Visual c++ on an Intel i7-3517U dual core CPU with 4GB RAM laptop. The experiments were executed with visual tracker benchmark datasets [7]. Totally, 19244 Images were used to calculate computation time and accuracy. The accuracy was measured with formula 4.

$$\text{Accuracy} = \text{Area}(F_{ref} \cap F_{sample}) \,/\, \text{Area}(F_{ref} \cup F_{sample}) \qquad (4)$$

Sequence Name	Computation time (ms) HoG	acHoG
David	54.18634	18.72921
Car4	53.13914	18.36059
FaceOcc1	60.43637	17.82848
Dudek	56.49820	18.61466

Table 2. Computation time comparison between HoG and acHoG based Particle Filter. We use sequential Image sets provided from visual tracker benchmark. [7]

Table 2 compares the computation times of object tracking between HoG and acHoG based particle filter trackers. As Table 2, adopting adaptive cell-size HoG reduces computation time by 65.37%.

Sequence Name	# 10	# 50	# 100	# 150	# 200	Accuracy
David						0.759451
Car4						0.693183
FaceOcc1						0.66082
Dudek						0.780548

Table 3. Accuracy and sample images of results

Table 3 shows accuracy of proposed algorithm. At the Doll sequence, we used largest feature size. The rest of sequences except Doll sequence had feature of same size. As the table 3, the feature of larger size had higher accuracy.

4 Conclusions

According to the characteristics of Particle Filter, features are extracted more than two times on the most of target regions. To solve this problem, we propose adaptive cell-size HoG sharing extracted gradients and angles on the overall target region. By the acHoG, several feature extractions at about 90% of target regions are substituted with just one extraction. As the experiments, the acHoG is more efficient in aspect of computation times by 65% than traditional HoG. Additionally, In contrast with other visual object tracking algorithm, our method uses feature which could be used to detect object. If we apply acHoG to detection, we expect that computations of detection also decrease. Therefore, the acHoG based particle filter will be useful to detect and track objects at the same time.

In future work, we will apply acHoG to object detection and improve the accuracy of particle tracker using additional features such as color, LBP, and SIFT etc. Finally, we will combine detection and tracking algorithm with acHoG sharing extracted gradients and angles.

Acknowledgements. This research was supported by the Industrial Strategic Technology Development Program of MOTIE [10040927, Driver-oriented vehicle augmented reality system based on head up display for the driving safety and convenience], and also supported by “Development of Advanced Digital Cockpit System for Driving Safety and Convenience” Project.

References

[1] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay et al., The Visual Object Tracking VOT2014 Challenge Results, Chapter in *Computer Vision-ECCV 2014 Workshops*, (2015), 191-217. http://dx.doi.org/10.1007/978-3-319-16181-5_14

[2] R. Faragher, Understanding the basis of the Kalman filter via a simple and intuitive derivation, *IEEE Signal Processing Magazine*, **29** (2012), no. 5, 128-132. http://dx.doi.org/10.1109/msp.2012.2203621

[3] A. Doucet and A. M. Johansen, A tutorial on particle filtering and smoothing: Fifteen years later, *Handbook of Nonlinear Filtering*, **12** (2009), 656-704.

[4] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing*, **50** (2002), no. 2, 174-188. http://dx.doi.org/10.1109/78.978374

[5] K. Whui, Y. Changrak, K. Jungyu, K. Kyong-Ho, Mobile Processor based Pedestrian Detection Rate Improvement Method, In: *2015 IEIE Fall Conference*, (2015), 587--590.

[6] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, **1** (2005), 886-893. http://dx.doi.org/10.1109/cvpr.2005.177

[7] Visual Tracking Benchmark, https://sites.google.com/site/trackerbenchmark/home

Received: April 3, 2016; Published: June 2, 2016