Fast Object Detection Using Multistage Particle Window Deformable Part Model

Wei-Ta Chu
National Chung Cheng University
Chiayi, Taiwan
wtchu@ccu.edu.tw

Ming-Hung Hsu
National Chung Cheng University
Chiayi, Taiwan
h12258@gmail.com

Abstract—**For object detection, evaluating all sliding windows at various scales draws a computational efficiency issue. In this paper, we propose a fast object detection framework using the multistage particle window strategy to accelerate the cascade deformable part model (DPM). Coupling this strategy with the proposed early jump scheme, adaptive particle window generation, and efficient preprocessing, we demonstrate that the proposed method runs 34.5 times faster than the conventional DPM to detect objects in images, and is able to efficiently detect vehicles and pedestrians in on-road videos.**

Keywords- deformable part model; multistage particle window; early jump; adaptive particle window generation

I. INTRODUCTION

Object detection is one of the fundamental challenges in computer vision. It can be seen as a classification problem (object vs. non-object), where sliding windows at all possible positions and scales are examined to determine whether the targeted object is included. This task is both time consuming and challenging due to different object appearances, sizes, illumination/viewpoint changes, and non-rigid deformation.

The deformable part model (DPM) [1] is one of the state-of-art works achieving promising object detection. Feature pyramids of a test image are extracted, and a sliding window approach is adopted to calculate the response of a specific window to the object model. The window causing the highest response is selected as the detection result. Although this approach achieves promising performance, construction of feature pyramids and response evaluation over all sliding windows cause computational problems:

- Feature pyramid construction: To detect objects of unknown sizes, extracting features at different scales is necessary and time consuming. To solve this problem, Dollar et al. computed features at a single scale and approximated features at nearby scales [7].
- Response evaluation: A few studies adopted the cascade strategy to speed up the detection process. This strategy quickly rejected most true negatives in early stages using simpler classifiers, and left complex evaluation in later stages only if necessary [4].

Although the cascade strategy makes detection faster, the computational cost is still high because all sliding windows were exhaustively examined. In this work, focusing on reducing the cost of response evaluation, we propose a new paradigm to accelerate the DPM: the multistage particle window DPM (mspw-DPM). Motivated by [3] that adopted the Monte Carlo sampling scheme to efficiently reduce the search space, we embed the idea of multistage particle window into the framework of the cascade DPM [4]. Based on the mspw-DPM, more speed-ups are further made from three perspectives: early jump in the detection procedure, adaptive particle window generation, and effective preprocessing based on the BRIEF descriptor [8].

The rest of this paper is organized as follows. Section II describes the mspw-DPM and the three strategies to further accelerate object detection. Section III employs the proposed model to detect objects in images, as well as pedestrians and vehicles in on-road videos. Section IV demonstrates detection performance, followed by conclusion in Section V.

II. MULTISTAGE PARTICLE WINDOW DPM

A. MSPW-DPM

We separate the detection process into stages. At the first stage, appropriate numbers of sliding windows (corresponding to particle windows in the proposed scheme) are sampled according to a density function that is initialized as a uniform distribution. Each particle window is then tested by the cascade DPM [4]. The likelihood of a particle window containing a targeted object is estimated by the ratio of layers passed by this particle window to the total number of layers in the cascade DPM. Scores of all particle windows are then collected to update the density function, by which new particle windows are sampled at the next stage. After several stages, the sampled particle windows are expected to be close to the true object locations. Figure 1 illustrates the process when we employ the *bicycle* cascade DPM to detect bicycles in an image. As can be seen, the sampled particle windows gradually converge stage by stage.

According to the density function $q_i(I)$, at the ith stage totally N_i particle windows are sampled for the image I. Note that the number N_i is much smaller than the total number of all possible sliding windows, and thus at each stage there are only moderate numbers of response evaluations. The width and height of each particle window are randomly given. In response evaluation, the score of the jth particle window $w_i^{(j)}$ at the ith stage is $r_i^{(j)} = \frac{\ell_i^{(j)}}{L}$, where L is the total number of layers in the cascade DPM, and $\ell_i^{(j)}$ is the number of layers passed by $w_i^{(j)}$. A particle window with low score means that its nearby region unlikely contains the targeted object.

The main idea of the mspw-DPM is skipping those regions with low scores, and sampling more particle windows closer to the regions with high scores at new stages. To update the density function at each stage, after response evaluation, each particle windows is assigned a Gaussian kernel $G(\mu_i^{(j)}, \Sigma_i^{(j)})$, where its mean $\mu_i^{(j)}$ is set according to the center of the j th particle window $w_i^{(j)}$, and the covariance matrix $\Sigma_i^{(j)}$ is consistent for all particle windows. The density function for sampling particle windows at the $(i+1)$th stage is then updated by

$$q_{i+1}(I) = \sum_{j=1}^{N_i} \pi_i^{(j)} \cdot G(\mu_i^{(j)}, \Sigma_i^{(j)}), \quad (1)$$

$$\pi_i^{(j)} = \frac{r_i^{(j)}}{\sum_{k=1}^{N_i} \pi_i^{(k)}}, \quad (2)$$

where $\pi_i^{(j)}$ denotes the weight of the j th Gaussian component, $r_i^{(j)}$ is the score the particle window $w_i^{(j)}$.

With the updated density function, more particle windows will be sampled at the positions closer to where the targeted object locates. Figure 1 shows a test image and its corresponding density function after one update. We can clearly see that, after only one update, the updated density has efficiently evolved to indicate possible locations of the bicycle object. Keeping the same process for a few stages, accurate object locations can be efficiently identified.

Figure 2 shows the distributions of particle windows at different stages, where we can see that the density function quickly converges and accordingly many samples would be drawn in a small locality. When the sampled particle windows are close enough to the targeted object, responses of them are similar. Therefore, there is no need to draw as many samples as we initially do at the first stage. In this work, we set $N_i = N_1 \cdot e^{\gamma(1-i)}$, where γ is empirically set as 0.44. As can be seen in Figure 2, the number of particle windows reduces stage by stage. If the parameter γ is smaller, fewer samples will be drawn, making detection fast but less accurate. On the other hand, if the parameter γ is larger, generally more accurate detection result can be obtained but the detection process is slower.

At each stage, if response of a particle window is larger than a threshold, this particle window is viewed to have the targeted object inside. In our work, totally five stages of detection and update are conducted in the test procedure.

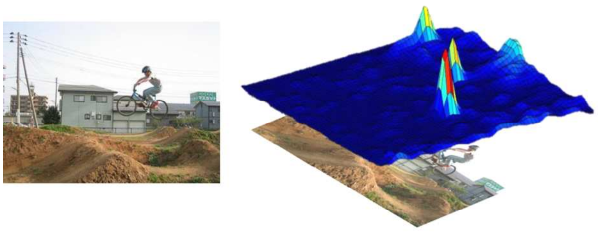

Figure 1. Left: The test image where we try to detect the *bicycle* object. Right: The updated density function after one update (for sampling particle windows at the second stage).

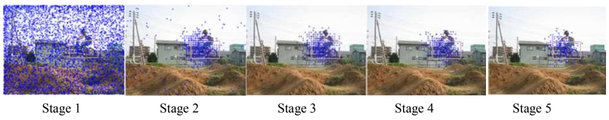

Figure 2. Distributions of particle windows at different stages. Each blue dot denotes the position of the left-top corner of a particle window.

The proposed method speeds up the cascade DPM [4] by 1) sampling just parts of sliding windows, and 2) adaptively reducing the number of samples. To further accelerate detection, we further propose processes from three perspectives, as described in the following

B. Early Jump Scheme

For a test image, we examine all object models to detect which objects are in the image. If we can know there is no targeted object in the test image earlier, the corresponding object model can be skipped at early stages to speed up the whole process. To achieve this, at each stage we evaluate scores of all sampled particle windows and stop the detection and density update processes if all scores are lower than a threshold. If we skip an object model at the ith stage, the number of response evaluation we save is $\sum_{j=i+1}^{m} N_j$, where m is the total number of layers of a cascade DPM, and N_j is the number of particle windows at the jth stage. This improvement speeds up object detection by disregarding inappropriate object models as early as possible.

C. Adaptive Particle Window Generation

Because there may be multiple targeted objects in an image, if we have likely detected one of them at a stage, we would like to put more efforts on the regions other than the detected region at the following stages. Assume that, at the ith stage, one targeted object has been detected in a particle window centered at (x, y). When updating the density function to be used at the $(i+1)$th stage, we set the scores of particle windows located within the particle window centered at (x, y) as zero. With this design, the peak of density function will move somewhere else, and more particle windows will be sampled at the positions far from (x, y). Namely, we will put more efforts on unexplored regions. Overall, the design of this updating strategy more effectively allocates computation resources.

D. Effective Preprocessing based on BRIEF

The BRIEF descriptor [8] is a binary vector computed from an image patch, and is designed to facilitate efficient image patch matching. If two image patches are similar, the Hamming distance between their BRIEF descriptors should be small. Because BRIEF descriptors and Hamming distances can be computed extremely fast, we exploit this idea to speed up the detection process.

Figure 3 describes how we utilize the idea of BRIEF in preprocessing. The root filter of the car model and a particle window can be separately viewed as two different patches. The root filter is divided into 8×8 blocks, each of which is represented by a histogram of oriented gradient (HOG). To construct the HOG-based BRIEF descriptor for the root filter, we randomly select pairs of blocks and randomly select a histogram bin to compare. By random selecting and comparing for 256 times, we obtain a 256-dimensional binary vector to represent a patch. To construct the counterpart for a particle window, HOGs are extracted from the region covered by the particle window, and then the HOGs and bins same as for the root filter are selected and compared to form a 256-dimensional binary vector. By calculating the Hamming distances between two HOG-based BRIEF descriptors, we can efficiently remove the particle windows that unlikely contain the targeted object.

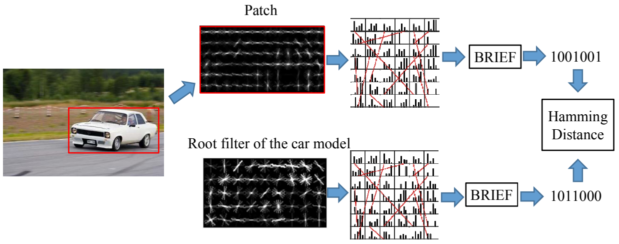

Figure 3. Preprocessing based on the BRIEF descriptor.

III. FAST OBJECT DETECTION AND TRACKING

Object Detection in Images. The goal of this task is finding locations of targeted objects in test images. We employ the proposed mspw-DPM to detect objects in images of the PASCAL VOC challenge 2007 dataset [2]. Because there is no clue to knowing what objects would appear in a test image in advance, for each test image we detect objects by going through all object models.

On-Road Pedestrian Detection. The constructed person mspw-DPM is employed to detect pedestrians in the image sequences proposed in [9], which were recorded at a resolution of 640 × 480 pixels and 15 fps using a camera mounted on a children's stroller. The first sequence (containing 450 frames, with 2,388 pedestrians) was taken on a sidewalk in a cloudy day, and the second sequence (containing 354 frames, with 1,900 pedestrians) was taken on a sidewalk in a sunny day. Different weather conditions, frequent partial occlusion, and complex background make the task extremely difficult.

On-Road Vehicle Detection. As the evaluation dataset, two videos were recorded at a resolution of 640 × 480 pixels and 30 fps using a camera mounted in a car. For efficiency, we adopt the color-based probabilistic tracking technique [10] based on a particle filter to locate object location rapidly, because appearance of objects in nearby frames does not change too much. Generally, we can detect the object in the t th frame and then track the object in $(t+i)$ th frame, $i = 1, 2, ..., n$. After tracking the object over an interval of n frames, we can do detection again to avoid error propagation and to handle new object coming.

IV. EXPERIMENTS

We evaluate performance of the proposed mspw-DPM, which was implemented in Matlab/C and executed in a PC with Intel Core 2 Duo 1.86GHz, 4GB RAM.

A. *Datasets and Setup*

Object Detection in Images. The PASCAL VOC challenge 2007 dataset [2] contains 9,963 images with 24,400 annotated objects. There are 20 object categories (aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, and tv/monitor) and each category contains hundreds of images.

A detected bounding box is considered correct if it overlaps the ground-truth bounding box by more than 50%. If several detected bounding boxes meet this criterion, only one detected bounding box is viewed correct, and the others are false positives. The average precision of the precision-recall curve is used to show overall performance [6].

On-Road Pedestrian Detection and Vehicle Detection Datasets. Table I and Table II respectively show detailed information of the datasets used for on-road pedestrian detection and vehicle detection. We compute precision and recall rates to demonstrate detection performance. Similarly, a detected bounding box is considered correct if it overlaps the ground-truth bounding box by more than 50%. We used the person model and the car model provided by [1][4], associated with the proposed multistage particle window strategy and acceleration approaches. In these two experiments, we did not utilize the BRIEF preprocessing.

TABLE I. DETAILED INFORMATION OF THE ON-ROAD PEDESTRIAN DATASET.

Video ID	resolution	duration(s)	fps	#Labeled pedestrian
1	640 × 480	30	15	2,388
2	640 × 480	23.6	15	1900

TABLE II. DETAILED INFORMATION OF THE ON-ROAD VEHICLE DATASET.

Video ID	resolution	duration(s)	fps	#Labeled car
3	640 × 480	113	30	2,308
4	352 × 240	168	15	2,253

Experimental Settings. The parameters of mspw-DPM are described below. The total number of particle windows at the first stage N_1 is set as $N_b/30$, where N_b is the total blocks of the HOGs in the feature pyramid. Each cascade DPM has 18 layers. The dimension of a binary BRIEF descriptor is 256, and the threshold ϵ_1 for preprocessing is 120. The threshold ϵ_2 for early jump is set as 8. About the parameters of object tracking, the total number of particles is 100, and the tracking interval is 10 frames, i.e., the whole detection process is performed for every 10 frames.

B. *Performance of Object Detection in Images*

Table III shows the average precision of detection performances obtained by various methods. DPM denotes the conventional DPM [1], csc-DPM denotes the cascade DPM [4], ESS (efficient subwindow search) denotes the branch-and-bound DPM [5], mspw-DPM denotes the proposed multistage particle window DPM, and the extra improvements 'early jump scheme', 'adaptive particle window generation', and 'BRIEF preprocessing' are extra denoted by e, a, and b respectively. Overall, average precision of the mspw-DPM decreases a little because sometimes the targeted object cannot be found in five stages. Increasing the numbers of stages or particle windows may improve average precision, but take more time to detection. Table IV shows the detection time and the speedup factor with respect to the cascade DPM. The proposed mspw-DPM is significantly faster than the ESS and the cascade DPM.

TABLE III. PASCAL VOC CHALLENGE 2007 RESULTS.

Methods	DPM	csc-DPM	ESS	mspw-DPM	mspw-DPM(e)	mspw-DPM (e+a)	mspw-DPM (e+a+b)
Average precision	0.307	0.280	0.276	0.241	0.242	0.243	0.202

TABLE IV. PASCAL VOC CHALLENGE 2007 DETECTION TIME AND SPEEDUP (IN SECONDS).

Categories	DPM	csc-DPM	ESS	mspw-DPM	mspw-DPM (e)	mspw-DPM (e+a)	mspw-DPM (e+a+b)
Mean exe time	19.267	1.364	1.049	0.754	0.697	0.716	0.542
Speedup factor	-	-	1.301	1.809	1.957	1.905	2.517

TABLE V. SPEEDUP FACTORS OF DEFERENT METHODS IN THE PASCAL VOC CHALLENGE 2007.

Method	Fourier-DPM	csc-DPM	ESS	mspw-DPM	mspw-DPM (e)	mspw-DPM (e+a)	mspw-DPM (e+a+b)
Speedup	7.4	14.4	18.3	24.5	25.4	26.9	34.5

TABLE VI. PERFORMANCE OF ON-ROAD PEDESTRIAN DETECTION.

Method	Video ID	precision	recall	mean detection time(s)
csc-DPM	1	0.527	0.351	2.314
mspw-DPM(e+a)	1	0.596	0.207	1.139
csc-DPM	2	0.833	0.730	2.145
mspw-DPM(e+a)	2	0.895	0.384	1.819

TABLE VII. PRECISION, RECALL, AND MEAN DETECTION TIME OF ON-ROAD VEHICLE DETECTION

Method	Video ID	precision	recall	mean detection time(s)
mspw-DPM(e+a)	3	0.799	0.645	0.454
mspw-DPM(e+a)	4	0.867	0.651	0.344

Comparing Table III with Table IV, the adaptive particle window generation policy makes detection faster and does not degrade performance. The BRIEF preprocessing can fast prune useless particle windows and speed up detection, but it decreases average precision a little because sometimes this descriptor is not so robust.

We also compare the speedup factors of the cascade DPM [4], the Fourier transform approach [11], ESS [5], and the mspw-DPM, with respect to the conventional DPM [1]. Table V shows the speed-up factors of deferent methods. Generally the mspw-DPM achieves the speed-up significantly higher than others, with slight degradation on average precision.

C. Performance of On-Road Pedestrian Detection and Vehicle Detection

Table VI and Table VII respectively show precision and recall rates, and average detection time of on-road pedestrian detection and vehicle detection. In Table VI we compare our detection results with the cascade DPM. Although the proposed detection process is averagely 1.5 times faster than the cascade DPM, we find that recall of the proposed method is much lower. The reason is that some pedestrians are mis-detected when too many pedestrians appear in a frame.

V. CONCLUSION

We have shown a fast object detection framework employing the multistage particle window strategy associated with the cascade DPM. The detection process is iterated in several stages, and dynamically sampled particle windows are evaluated. The density function used to sample particle windows keeps updating based on responses of particle windows to object models. Instead of exhaustively examining all possible sliding windows, the proposed method efficiently samples parts of them and largely reduces detection time. Experimental results demonstrate the proposed method at most runs 34.5 times faster than the conventional DPM in detecting objects in images. Coupling with a tracking technique, the average time to detect vehicles in on-road videos is 0.344 seconds.

Acknowledgement

The work was partially supported by the Ministry of Science and Technology in Taiwan under the grants NSC101-2221-E-194-055-MY2 and MOST103-2221-E-194-027-MY3.

REFERENCES

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. “Object detection with discriminatively trained part based models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1627 - 1645, 2010.

[2] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The PASCAL visual object classes challenge 2007 (VOC2007) results,” [Online] Available: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

[3] G. Gualdi, A. Prati, and R. Cucchiara. “Multistage particle windows for fast and accurate object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1589 - 1604, 2012.

[4] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. “Cascade object detection with deformable part models,” Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 41-50, 2010.

[5] I. Kokkinos. “Bounding part scores for rapid detection with deformable part models,” Proceedings of European Conference on Computer Vision, 2012.

[6] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” International Journal of Computer Vision, pp. 303-338, 2010.

[7] P. Dollar, S. Belongie, and P. Perona. “The fastest pedestrian detector in the west,” Proceedings of British Machine Vision Conference, 2010.

[8] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, “BRIEF: Computing a local binary descriptor very fast,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1281-1298, 2012.

[9] A. Ess, B. Leibe, and L. Van Gool, “Depth and appearance for mobile scene analysis,” Proceedings of IEEE International Conference on Computer Vision, pp. 1-8, 2007.

[10] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking,” Proceedings of European Conference on Computer Vision, pp. 661-675, 2002.

[11] C. Dubout and F. Fleuret, “Exact acceleration of linear object detectors,” Proceedings of European Conference on Computer Vision, pp. 301-311, 2012.