7

Combination of Sparse Scan and Dense Scan for Fast Vision-based Object Recognition

Tam Phuong Cao
Department of Electronic Engineering, La Trobe University, Bundoora, Vic 3086 Australia

1. Introduction

Real-time processing speed is desirable for most vision-based object recognition systems. It is critical in some applications such as driver assistant systems (DAS). In DAS, the vision-based system is expected to operate on a moving platform and the system is required to inform the driver in a timely manner. Therefore, real-time frame rate is very important.

Much improvement has been made over the years to improve the speed of computer-vision systems. The cascaded classifier architecture and the integral image Viola & Jones (2001) are two of those major improvements in recent years. Following this approach many vision based systems, including face detection Viola & Jones (2001), can process images at high frame rate. However, with increasing tasks' complexity and image size, real-time processing speed for object detection systems remains a challenge.

Hardware platforms such as field programmable gate array (FPGA) have been employed to speed up the system performance. Cao Cao & Deng (2008) has shown that vision system can achieve real-time frame rate (60fps) with FPGA. However, the development time for an FPGA system is significantly longer than a system on computers. Hence, it is helpful to have real-time or close to real-time systems on computers as a proof of concept before porting them to hardware platform(s).

Frintrop Frintrop et al. (2007) proposed a visual attention system which combines bottom up and top down approaches to guide the search to interested regions. The bottom-up approach computes contains saliency maps based on image features such as intensity, colour and orientation. The top-down approach computes features specific to the target objects. Test results showed that using integral images Viola & Jones (2001) significantly improve system speed compared to the original method by Itti Itti et al. (1998). However, due to the complexity of the system, the processing frame rate reported (with optimized implementation) was only about 5.3 fps with 800×600 images or equivalent of about 7 fps with 752×480 images.

Zhang Zhang et al. (2007) proposed a method for detecting objects using multi-resolutions. In this method, multiple down sampled copies of the original image are used for detection purpose. The lowest scale is first computed, and then the detection process progresses to higher (resolution) scale. This method improves the speed of the processing system while maintaining or even marginally improve the system's accuracy compared to using only one (original) scale approach. By process a pixel in every 64 (8×8) pixels, the system reported

a frame rate of 25fps with image size of 320×240, or equivalent of about 6fps at 752×480. The proposed method, however, may not work well on small target objects where excessive down sampling may deteriorate the clarity of the target object, thus making it harder to detect. Another potential issue with this method is the large amount of memory required to store multiple copies of the original image, this memory usage could create a bottle neck when this method is used for large images.

Forssen Forssen et al. (2008) also used multi-resolution to detect and track objects. Due to the complexity of the search algorithm, this system could not achieve fast detection speed, only about 5 frames per minute or 0.08 fps. This approach also requires complex and expensive camera system to operate. Other multi-resolution approaches including Ma's Ma & Staunton (2005), Walther's Walther et al. (2005), Meger's Meger et al. (2008) and Cho's Cho & Kim (2005) are either too computational costly or not built to reduce system's computational cost.

From a different point of view, part-based object detection has been investigated for applications such as human detection Mohan et al. (2001); Wu & Nevatia (2007) or car detection Agarwal et al. (2004). The main purpose of these methods are searching for different parts of the target object and their relative relationship to detect the target object. The object's parts are either hand picked (by a trained person) Mohan et al. (2001); Wu & Nevatia (2007) or automatically selected Agarwal et al. (2004). In these part-based algorithms, searching for parts of target object is not to improve detection speed but to detect the target object itself.

In this paper, we propose a vision-based object detection method that combines part-based and multi-resolution approaches. This method detects target object based on the appearance of some of its parts as a result of sparsely scan through the original image. A finer scan in the image only happens at promising locations obtained from the previous sparse scan stage. This method does not require any down sampling of the original image. This method is similar to the cascaded classified in that it quickly processes easy features in simple stage (spare scan) then difficult features are processed by more powerful (but high computational cost) stage.

The remainder of this paper is organised as follows. Section II introduces the general dense scan and describe the proposed method. Section III describes the example systems employing the proposed method. Experiment results and discussions are also included in Section III. Finally, some conclusions are made in Section IV.

2. Proposed method

2.1 Conventional dense scan

As a common approach, to detect the target object in an image, a detection window at a specified size is scanned across the image. After pixels within that window are processed, the window is moved to the next location and the process repeats. The space between two windows' locations determine how many windows are to be processed in an image. Given the same algorithm and computing platform, an increased number of detection window to be processed will increase the computational cost, therefore increase the processing time required for an image. The most popularly used scanning methods is the dense scan (DS) method where the detection window is scanned pixel by pixel within the image. To process a large image, the DS method needs to process a very large number of detection windows, resulting in long processing time.

A simple idea that has been widely used to reduce computational cost in many practical

applications is skipping pixels, i.e. a certain number of rows and/or columns of pixels or certain areas in the image are ignored, leaving a smaller number of pixels being processed normally. As a result the amount of data to be processed per image is reduced. However the detection performance of the algorithm may be degraded. The amount of accuracy reduction depends on the robustness of the algorithm and the size of the target object. For example, in a pedestrian detection system proposed by Dalal Dalal & Triggs (2005), the 64×128 pixel detection window is sparsely scanned (every 8^{th} rows and columns) to achieve faster processing. It was reported that the detection performance of the system increased by 5% when the detection window scanned the image more densely (every 4^{th} rows and columns) while computational time increased significantly. This indicates that some accuracy has been sacrificed for speed by skipping pixels. In the example system Dalal & Triggs (2005), the target object is large, hence the small increase in the scan step did not significantly affect the detection performance. The effect of skipping pixels to the system's performance is expected to be larger when a small object is to be detected within a high resolution image.

2.2 Sparse scan

In this paper a combined scanning method is proposed to improve speed while maintaining accuracy of the detection algorithm. In the SS part of the combined scanning method, detection window is moved multiple rows and columns every time the detection window finishes processing at a location. This is similar to the case of skipping pixels. The major difference between the SS, DS scanning (and pixel skipping) methods is in how positive and negative samples are defined during the training process. During the training process, positive samples used for DS and pixel skipping method assume full appearance of the target object while the SS method assumes only some parts of target object are presented. This is similar to detecting parts of the object in the image.

In the case of the DS method, a positive detection window contains an image of the target object together with some noise. In the some transform feature space, such as HoG Dalal & Triggs (2005), the positive detection window is an input vector $\mathbf{x_d}$ which is made up of:

$$\mathbf{x} = \mathbf{x_d} + \mathbf{n} \tag{1}$$

where $\mathbf{x} = \{x_1, x_2, ..., x_p\}$ is a p-dimensional vector generated from a detection window, $\mathbf{x_d} = \{x_{d1}, x_{d2}, ..., x_{ds}\}$ is p-dimensional input vector from the target object, $\mathbf{n}$ is noise in the image such as lighting variations, rotation, skew or white noise.

In the DS method, the detection window passes through every location in the image. Therefore, if there is a target object of the right size in the image, it will fully appear in the detection window and be detected at some point. An example of a possible positive example for the DS classifier is shown in Fig.1.(a) and Fig.1.(c) where a full stop sign or a person appears in the detection window.

For the SS classifier, a positive detection window contains an image of major parts of the target object, image of random background and noise. In some transformed feature space, such as HoG Dalal & Triggs (2005), a positive input vector $\mathbf{x_s}$ is made up of:

$$\mathbf{x_s} = \mathbf{u} + \mathbf{t} + \mathbf{n} \tag{2}$$

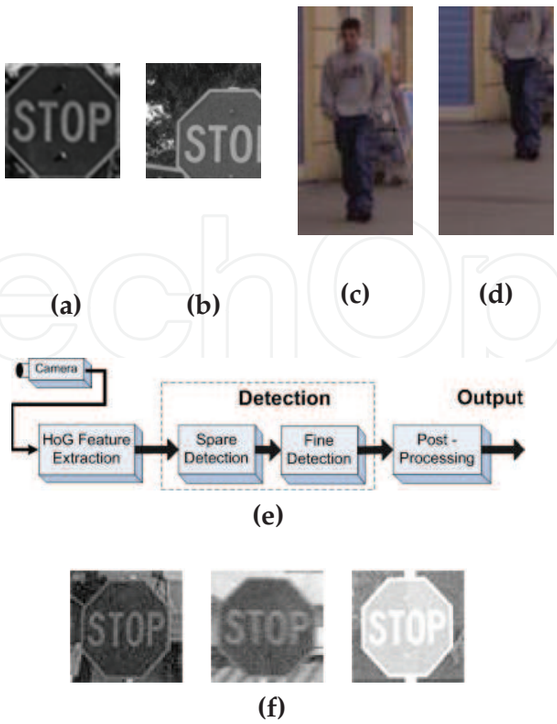


Fig. 1. Positive samples. (a) A positive sample for DS method. (b),(c)Positive sample for SS method which contains majority of stop sign and some random background.

where $\mathbf{u} = \{u_1, u_2, ..., u_p\}$ is a p-dimensional input vector generated from the visible parts of target object within the detection window; $\mathbf{t}$ is another p-dimensional vector that is generated from the random background in the detection window; $\mathbf{n}$ is noise as in (1), which caused by lighting variations, rotation and/or white noise. Equation (1) is a special case of (2) when $\mathbf{t} = \mathbf{0}$, which means the entire stop sign appear inside the detection window. Examples of positive detection windows for the SS classifier are shown in Fig.1(b) and Fig.1.(d) where only some parts of the target object (stop sign or pedestrian) appear in the detection windows.
With the SS technique, a detection window is scanned block by block across the image instead of pixel by pixel. Each block is selected to have certain size such as 3×3 or 6×6 pixels. There is no prior information regarding which parts of the target object are visible or missing, and what type and size of background $\mathbf{t}$ and noise $\mathbf{n}$ in the image. Compared to (1) which has only one random variable $\mathbf{n}$ to be estimated in the training. Therefore (2) is harder to estimate during training process and different training data is required for SS classifiers.

We propose a combined method containing different SS classifiers and DS classifier. With this method, the SS classifiers quickly and roughly process the image then output a map of interested regions that may contain the target object. This map is then used by the DS method to thoroughly process the interested regions to detect whether the target object is actually presented. This combination is similar to multi-resolution approach where the sparse scan (larger block size) acts as processing low resolution image. The finer (smaller) SS and the final DS classifiers act as subsequent higher resolution images. The algorithm of this combined method is shown in Fig.2. This method can speed up the detection process as most of the

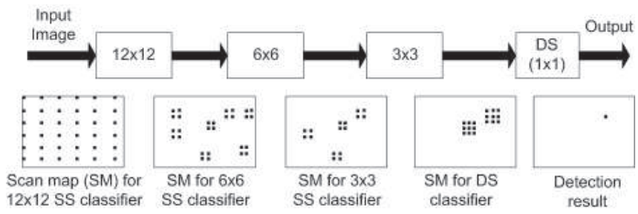


Fig. 2. Detection algorithm using combined SS and DS classifiers. Input image is first processed by the 12×12 classifier using initial scan map (SM). Then new SMs are generated, in which positive entries are at interested regions and their neighbours indentified by previous SS classifier. Final DS produces the detection result.

required processing is handled by the fast SS classifiers which sparsely scan through the image. Only small amount of processing is handled by the DS classifier. This method has another advantage of not requiring to perform resampling and store them in memory. In addition, because only the full resolution image is used, the target object is large hence easier to detect or lower FPPW rate.

3. Example system

3.1 Training data

To compare the performance of different approaches, SS and DS classifiers together with multi-resolution classifiers Zhang et al. (2007) are trained and analysed. Stop signs are chosen to be the target objects. For other target objects, the implementation process is similar but the performance is likely to differ.

Data for training as well as for testing were collected with an automotive grade camera, the Micron MT9V022, mounted near the rear view mirror of a car, as shown in Fig.3. This camera has native resolution of 480×752 pixels. A database of positive and negative sample images is built to train different classifiers. The positive sample set contains 225 extracted images at size of 36×36 pixels containing stop signs. These signs may be affected by size variations, rotation, skew, motion blur or added white noise. The stop signs in the positive sample set have the size of 36 ± 2 pixels across. Three of the positive examples are shown in Fig.1(f). It should be noted that the noise appear in those positive examples are actual noise recorded in the image. Added random noise (with maximum amplitude of 20 for 8-bit grayscale pixels) further degrades the quality of the image. This is done to improve the systems' robustness against noise in different lighting conditions and camera settings.

Different classifiers are trained with different positive and negative examples extracted from the training database. The positive examples of SS classifiers (3×3, 6×6, 9×9, 12×12 and the combination of 12×12, 6×6 and 3×3) were extracted from the positive sample set by using appropriate portion of stop sign (the u portion of e.q.(2)) and randon values (the t portion of e.q.(2)). For example, one member of the positive sample set can generate 9 positive examples when training the 3×3 SS classifier because the t portion in the positive examples of 3×3 SS classifier can range from 0 to 2 pixels in vertical and horizontal directions.

The negative sample set contains 195 negative images mostly at the resolution of 480×752 pixels. These negative images capture scenes of roads, buildings people, and road signs (other than stop sign). Negative examples are collected by moving a detection window within

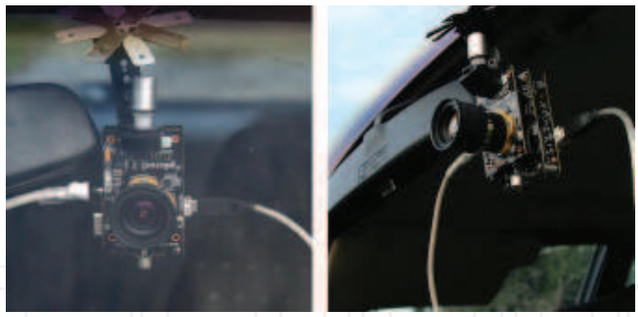


Fig. 3. Camera used to collect data for training and testing. The camera is powered by USB cable connected to a laptop PC.

the negative images. Due to the resolution difference between positive and negative sample images, the total number of negative examples is much higher than those of positive examples. A test set consists of about 9700 images extracted from the 29 video sequences are used to test the performance of classifiers.

Different stop sign detectors were trained on a PC using Matlab following the DS only, combined SS and DS, and multiresolution approaches. These detectors employ AdaBoost and cascaded of classifiers techniques by Viola Viola & Jones (2001). The detection algorithms in these system were based on a variant of the HoG Dalal & Triggs (2005) feature set where only gradient angle is used (pixel's gradient magnitude was disregarded for faster computation of HoG features) similar to Cao & Deng (2008). The overview of detection system using the proposed SS and DS methods is shown in Fig.1.(e). Detection system that employs only the conventional DS technique is similar to the system shown except that the sparse scan step is by passed. The sparse scan block is equivalent to the low resolution processing in the detection system using the multi-resolution approach following Zhang's Zhang et al. (2007) approach.

3.2 Performance of classifiers

To compare computational cost of the DS, SS and simple multiresolution methods, we quantify computational cost of an algorithm as the total number of detection windows processed. According to the training results of different classifiers, shown in Fig.4.(a), it can be said that the smaller the block size, the lower FPPW the classifier can achieve. This is expected because the random portion t of (2) decreases with the reducing block size. Fig.4.(b) compares detection results based on the low resolution (resampled) images with those of the proposed SS methods. The Half-DS and Quarter-DS classifiers represent classifiers that process resampled images at half or quarter respectively of the number of rows and columns compared to the original image. This was to implement the multi-resolution detection where low resolution image is processed first. Interestingly, the Half-DS classifier performed worse (higher FPPW) than the 3×3 SS classifier and the Quarter-DS was much worse. This may be attributed to the difficulty of detecting the target object in low resolution images. Our proposed combined SS, whose structure is shown in Fig.2, has the FPPW approaching that of Half-DS (at lower computational cost). The 3×3 SS classifier aas the best performance with FPPW of 3.37×10^{-3}. Some example outputs of different sparse detection classifiers are shown in Fig.5. This figure clearly shows that full scan in low resolution produce a lot of false positive.

To compare the cost between different classifiers, let's assume that the total number of

Factors	SS Cost	DS Cost	Total Cost
3×3	34808	1058	35866
6×6	8702	15287	23989
9×9	3867	37587	41454
12×12	**2175**	74043	76218
Combined SS	6833	**192**	**7025**
Half-DS	78320	7894	86214
Quarter-DS	19580	17896	37476
DS only	0	313280	313280

Table 1. Cost of Different Block Size

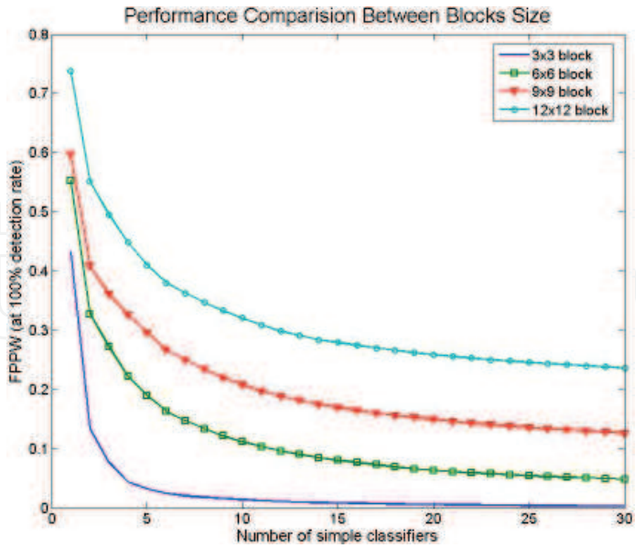
Factors	**Detection**	**FPPW**	**Processing Time**
3×3	97.74%	8.82×10^{-7}	6.24 sec
6×6	96.22%	9.93×10^{-7}	4.10 sec
Half-DS	97.59%	9.16×10^{-7}	19.66 sec
Quarter-DS	97.60%	1.69×10^{-6}	8.57 sec
Proposed	97.44%	7.99×10^{-7}	1.98 sec

Table 2. Overall Performance of Different Classifiers

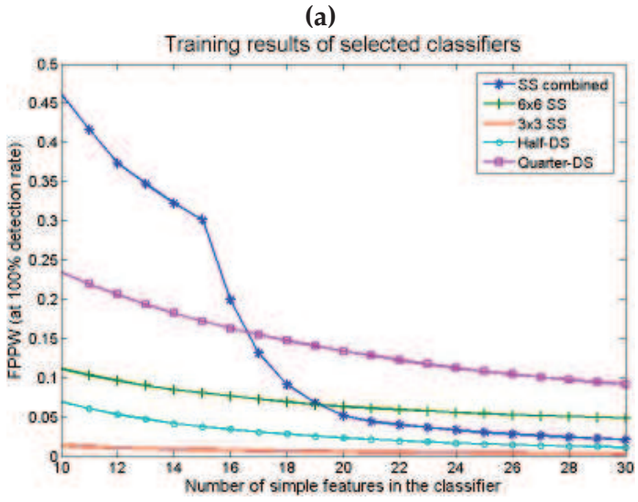
detection windows in a 480×752 image is 313,280 windows (excluding border pixels). When the DS only approach is taken, the total cost is simply 313,280 windows. When the system uses both SS and DS classifiers, with the structure shown in Fig.2, the number of windows processed is made up of the initial number of window processed by the SS classifier(s) (SS cost) and the number of windows need to be processed by the DS classifier (DS cost) as resulting from a positive output of the SS module. The SS cost is fixed and depends only on the block size of the SS classifier. The SS cost of different block sizes for the SS classifier is shown on Table.1. The DS cost depends on the performance of the SS classifiers. For example, if an SS classifier with block size of 3×3 accepts 100 windows as positive windows, then the DS cost is $100 \times 3 \times 3 = 900$ windows. As one would expect, the larger the block size, the less powerful the classifier will be, resulting in a higher DS cost. Based on training result, it is shown on Table.1 that the proposed combined SS and DS method has the least total computational cost. The Half-DS classifier has the second highest computational cost, behind the full DS.

3.3 Experimental results

After training, different classifiers were used to implement completed detection systems including the verification module following structure shown in Fig.1.(e). The same DS classifier is shared mong those systems. These system were tested against the test data set and the test results on a 2.6 GHz PC are summarised in Table.2. It is shown in Table.2 that the combined SS and DS systems have the lowest processing time which is about 20 times faster than the conventional method using only DS scanning technique. This speed improvement is due to the low computational cost of the combined SS classifiers, as shown on Table.1. It should be noted that the systems implemented on Matlab did not implement the integral images for each angle bin of the HoG features as used in Zhu et al. (2006). Using integral images is expected to further improve systems' speed without affecting the accuracy.



(a)



(b)

Fig. 4. Training results of Different Block Sizes for different classifiers. (a). Different SS classifiers. (b). Compare some SS classifers with other multi-resolution classifiers.

In terms of detection rate, all systems have similar detection performance because all of them share the last DS classifier. This similarity may change if different DS classifiers were trained and used for each system. The overall FPPW rate of each systems depends on the number of windows that passed the SS classifier(s) which is represented by the DS cost on Table.1. The combined different SS method has the least DS cost, hence it has the lowest FPPW as shown on Table.2. The false positive rate is greatly affected by the variations of the stop sign's size when the video sequences are manually annotated.

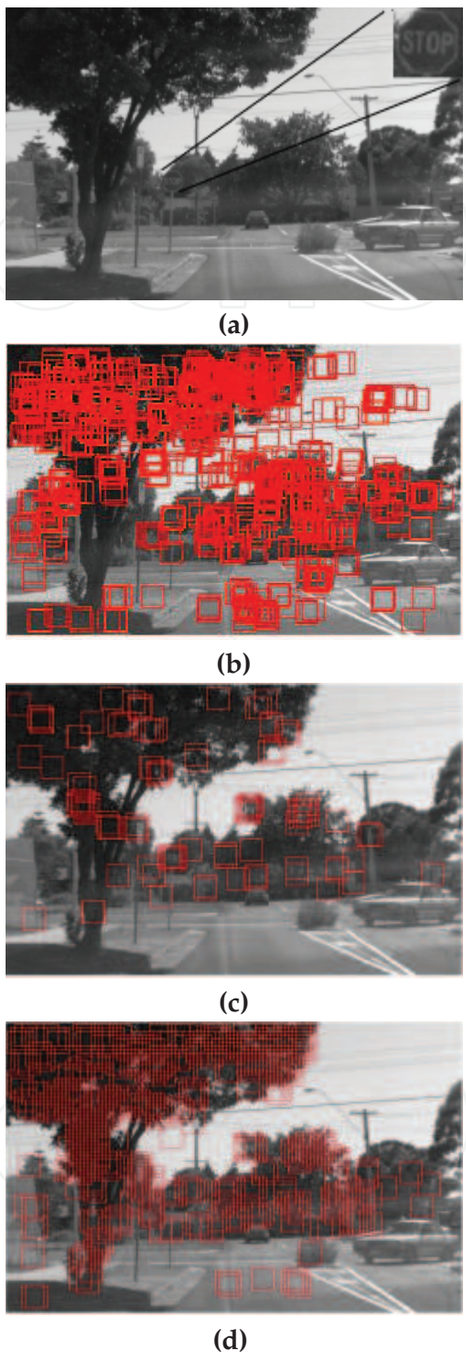


Fig. 5. Example output of SS and Half-DS classifier (a). Original Image. (b). Output of Half-DS classifier. There are a lot of false positives because there are 78320 windows to be processed per 376×240 image. (c). Output of 3×3 SS classifier. (d). Output of 6×6 Classifiers

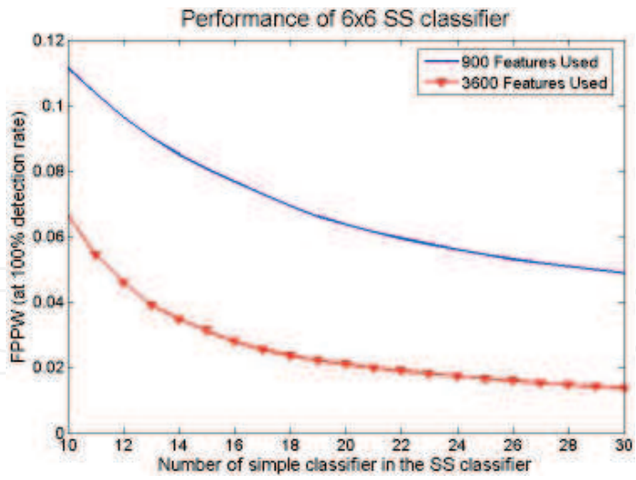


Fig. 6. Performance comparision of 6×6 classifier when using different feature sets

Classifier	DS	6×6 SS-DS	3×3SS-DS
Processing Time	216 ms	77ms	108ms

Table 3. Processing time of classifiers implemented using C : initial result

3.4 Discussions and future work

The performance of the systems studied in this paper can be further improved. The first technique could be used is to use a large more comprehensive feature set in training the classifiers. Generally, the SS and DS classifiers have a better performance ,i.e. lower FPPW rate, when more features are used in the training process. For example the 6×6 SS classifier can improve the performance to 0.014 FPPW when training the classifier with a feature set that contains 3600 HoG features, as shown in Fig.6. It is expected that increasing number of simple features in a classifier also decreases the FPPW rate. With the larger target object, larger sizes of the SS classifiers could be used. In our example, the target object is rather small 36×36, therefore the maximum scan step size considered was 12×12. With larger target object, it is possible to use larger block size to further reduce the computational cost.

The detection systems described in this paper are in the process of being ported to a C implementation. As shown on Table.3, with the initial C implementation using integral images Viola & Jones (2001), the 6×6 SS-DS classifier takes about 77ms to process a 752×480 image on a 2.6GHz PC (not including time for reading and writing input and output files). With those classifiers implemented in C, the time taken to construct four IMaps (one for each angular bin) is about 55ms. The amount of time for constructing IMaps is the major computing time required in the 6×6 and 3×3 SS-DS classifiers. If not taking to account the time taken for the IMaps construction, the 6×6 SS-DS classifier runs about 7 times faster than the original DS only method.

4. Conlcusion

In this paper, a combined SS and DS method for fast vision-based object recognition is proposed. This method is based on the combination of part-based and multi-resolution object

detection. The processing of the image starts by sparsely scan the image to find parts of the target object. Finer scan is performed at those locations where positive output was detected at sparse scale. This method shows significant improvement in terms of speed while system's comparable accuracy compared to previously proposed multi-resolution methods. With the use of integral map and optimized software implementation, this method expected to be suitable for real-time applications.

5. References

Agarwal, S., Awan, A. & Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation, *IEEE Journal of Pattern Analysis and Machine Intelligent* **26**: 1475–1490.

Cao, T. P. & Deng, G. (2008). Real-time vision-based stop sign detection system on fpga, *Proc. 2008 Digital Image Computing: Techniques and Applications*, Canberra, Australia, pp. 465–471.

Cho, J.-H. & Kim, S.-D. (2005). Object detection using multi-resolution mosaic in image sequences, *Signal Processing: Image Communication* **20**(3): 233–253.

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection, *2005 IEEE Computer Scociety conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 886–893.

Forssen, P.-E., Meger, D., Lai, K., Helmer, S., Little, J. J. & Lowe, D. G. (2008). Informed visual search: Combining attention and object recognition., *Procs. IEEE International Conference on Robotics and Automation*, pp. 935–942.

Frintrop, S., Klodt, M. & Rome, E. (2007). A real-time visual attention system using integral images, *Proc. 2007 International Conference on Computer Vision Systems*, Germany, pp. 3385–3390.

Itti, L., Koch, C. & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis, *IEEE transactions on Pattern Analysis and Machine Intelligence* **20**: 1254–1259.

Ma, L. & Staunton, R. (2005). Integration of multiresolution image segmentation and neural networks for object depth recovery, *Pattern Recognition* **38**(7): 985–996.

Meger, D., Forssén, P.-E., Lai, K., Helmer, S., McCann, S., Southey, T., Baumann, M., Little, J. J. & Lowe, D. G. (2008). Curious george: An attentive semantic robot, *Robot and Autmation System* **56**(6): 503–511.

Mohan, A., Papageorgio, C. & Poggio, T. (2001). Example-based object detection in images by components, *IEEE transactions on Pattern Analysis and Machine Intelligence* **23**: 349–361.

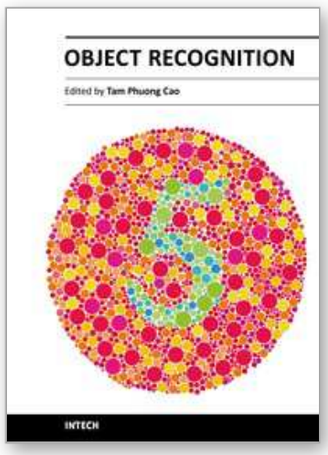
Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascaded of simple features, *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Vol. 1, pp. 511–518.

Walther, D., Rutishauser, U., Koch, C. & Perona, P. (2005). Selective visual attention enables learning and recognition of multiple objects in cluttered scenes, *Computer Vision and Image Understanding* **100**(1-2): 41–63.

Wu, B. & Nevatia, R. (2007). Improving part based object detection using unsupervised, online boosting, *Proc. IEEE Computer Society conference on Computer Vision and Pattern Recognition*, Minnesota, USA, pp. 1–8.

Zhang, W., Zelinsky, G. & Samaras, D. (2007). Real-time accurate object detection using multiple resolutions, *Proc. 2007 IEEE International Conference on Computer Vision*, Vol. 1, Rio De Janeiro, Brazil, pp. 1–8.

Zhu, Q., Avidan, S., Yeh, M.-C. & Cheng, K.-T. (2006). Fast human detection using a cascade of histogram of oriented gradients, *Proc. IEEE computer society conference on computer vision and pattern recognition*, New York, USA, pp. 683–688.



Object Recognition

Edited by Dr. Tam Phuong Cao

ISBN 978-953-307-222-7
Hard cover, 350 pages
Publisher InTech
Published online 01, April, 2011
Published in print edition April, 2011

Vision-based object recognition tasks are very familiar in our everyday activities, such as driving our car in the correct lane. We do these tasks effortlessly in real-time. In the last decades, with the advancement of computer technology, researchers and application developers are trying to mimic the humanâ€™s capability of visually recognising. Such capability will allow machine to free human from boring or dangerous jobs.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tam Phuong Cao (2011). Combination of Sparse Scan and Dense Scan for Fast Vision-based Object Recognition, Object Recognition, Dr. Tam Phuong Cao (Ed.), ISBN: 978-953-307-222-7, InTech, Available from: http://www.intechopen.com/books/object-recognition/combination-of-sparse-scan-and-dense-scan-for-fast-vision-based-object-recognition

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen