Context-Aware Model Applied to HOG Descriptor for People Detection

METZLI RAMIREZ-MARTINEZ
Univ. Bourgogne Franche Comté
DRIVE EA1859 Laboratory
Nevers, FRANCE
Metzli_Ramirez-Martinez@u-bourgogne.fr

FRANCISO SANCHEZ-FERNANDEZ
Univ. Bourgogne Franche Comté
DRIVE EA1859 Laboratory
Nevers, FRANCE
Francisco.Sanchez-Fernandez@u-bourgogne.fr

Philippe Brunet
Univ. Bourgogne Franche Comté
DRIVE EA1859 Laboratory
Nevers, FRANCE
Philippe.Brunet@u-bourgogne.fr

Sidi-Mohammed Senouci
Univ. Bourgogne Franche Comté
DRIVE EA1859 Laboratory
Nevers, FRANCE
Sidi-Mohammed.Senouci@u-bourgogne.fr

El-Bay Bourennane
Univ. Bourgogne Franche Comté
LE2I Laboratory
Dijon, FRANCE
ebourenn@u-bourgogne.fr

Abstract: This work proposes and implements a method based on Context-Aware Visual Attention Model (CAVAM), but modifying the method in such way that the detection algorithm is replaced by Histograms of Oriented Gradients (HOG). After reviewing different algorithms for people detection, we select HOG method because it is a very well known algorithm, which is used as a reference in virtually all current research studies about automatic detection. In addition, it produces accurate results in significantly less time than many algorithms. In this way, we show that CAVAM model can be adapted to other methods for object detection besides Scale-Invariant Feature Transform (SIFT), as it was originally proposed. Additionally, we use TUD dataset image sequences to evaluate and compare our approach with the original HOG algorithm. These experiments show that our method achieves around 2x speed-up at just 2% decreased accuracy. Moreover, the proposed approach can improve precision and specificity by more than 2%.

Key–Words: Object detection, pedestrian detection, tile-based method, saliency, regions of interest

1 Introduction

People detection is a very common problem in many automatic vision systems. It can be used in plenty of applications like person identification, congestion analysis, and automotive pedestrian detection. These applications require both, robust and fast detection. In consequence, many accurate methods have been developed in the last years [1, 2, 3, 4, 5]. Unfortunately, it appears that in many cases, improving the accuracy detection implies increasing computational cost. For that reason, one of the biggest challenges in this kind of algorithms is to reduce the execution time.

In this article, we show an approach based on CAVAM model [6], which allows reducing time and data processing. The original CAVAM model [6] uses SIFT method for object detection. We propose to integrate HOG method with CAVAM model, by adapting the concept of familiarity from SIFT to HOG. Thereby, we test the flexibility of the model to be applied to different algorithms, and simultaneously, our method speeds up HOG algorithm about 2 times with virtually no accuracy reduction. Furthermore, in Sec. 4 we show that our approach outperforms other methods for accelerating people detection.

This paper is organized as follows. We consider the related work in Sec. 2. In Sec. 3 the framework of the method is explained, where different algorithms for people detection are compared before giving details of our approach. In Sec. 4, we explain our experiments and compare the results of our approach and other people detection algorithms. Finally, the achievements, advantages, and weaknesses of our method are reviewed in Sec. 5.

2 Related work

There are plenty of works and studies for human detection but there are still many challenges to solve. Therefore, several methods have explored different ways to improve performance and reliability. For example, Zang et al. [7] present an approach based on motion analysis for pedestrian detection. This helps significantly to speed up the detection, but it is con-

ditioned by the people motion. In other words, if we have a static person in the scene, this kind of algorithm is not suitable. Other examples are methods specialized on specific problems, like the approach proposed by Tang et al. [8], which is focused on occlusion. They developed a two parts joint method, the first part detects single people and the second part detects pairs of people under occlusion. The model improves the detection results and it provides an important contribution in the area. Nevertheless, the principal idea of the method is based on a well-known algorithm, Deformable Parts Model (DPM) [9], which is a variant of HOG algorithm. In this way, we can see why algorithms like HOG, SIFT, Speeded Up Robust Features (SURF), and Hierarchical Model and X (HMAX), among others, persist through the time, because they are capable of working under basic conditions without any other requirement, therefore, they are still the basis of many current works.

Due to the slowness of many object detection algorithms, several acceleration methods have been developed in the last few years. For example, Li et al. [10] present a method for pedestrian detection, which reduces the redundant information of the original HOG descriptor. This method analyzes the effectiveness of each channel used to compute the gradient histograms, and it reduces the features dimension with an error rate increment not bigger than 5%.

Alternatively, methods like Binary Robust Invariant Scalable Keypoints (BRISK) [11] and Fast Retina Key- point (FREAK) [12] also seek to accelerate multi-scale algorithms. Both methods belong to SIFT and SURF family and both claim to improve the performance and execution time than their predecessor algorithms. They reduce the descriptor dimensionality, and they also limit the processing to regions of interest (ROI). Similar to our work, these approaches use a saliency criterion to determining the ROI, resulting in a reduction of the amount of data to process. Particularly, each method reduces the descriptor dimensionality by a different technique of sampling. While BRISK uses an equally spaced circular pattern, FREAK uses a circular pattern with the higher point density near the center. According to Ref. Schaeffer2012, BRISK and FREAK algorithms indeed outperform SURF method in people detection context. They achieve 2% and 6% more accuracy, and they are also almost 2 and 3 times faster than SURF, respectively. In the following sections, we can find a detailed comparison of our method with all these approaches.

Additionally, in the last few years, machine learning methods have been taking relevance in the field, among witch methods based on Convolutional Neural Networks (CNN) [5] have been positioned as one of the most accurate algorithms in object detection, and the majority of the software implementation exceeds the speed of HOG. In contrast to traditional algorithms, where the feature computation and the classification are different stages, CNN treats the feature extractor and the classification in exactly the same way.

However, even if methods based on CNN show considerable advantages, they also have important limitations. These methods require a large number of samples for training and it takes a lot of time. Additionally, multiple traditional methods continue to be used to complement CNN for reducing the error rate or improving the speed. Some examples of these works are [14], [15] and [16]. This last work, proposed by Tang et al., uses HOG algorithm in a first stage to adjust the CNN classifier and reduce noise. Therefore, it shows the usefulness of the HOG method at the present.

Although CNN algorithm is at least one order lower than HOG, CNN by itself is far to achieve a real-time speed, while hardware implementations consume much more energy than HOG hardware architectures, as we can verify in [17]. Despite we are not describing a hardware architecture, we want to highlight the fact that the utilization of this framework in HOG, could be used to implement in a dynamically reconfigurable hardware architecture for the efficient management of resources, because the computation of ROI allows us to reconfigure according to the amount of data to process. Thus, the high energy consumption of CNN architectures is an important disadvantage.

Moreover, HOG made a breakthrough in people detection and, as we already mentioned, it continues to be used in multiple methods of detection. But mainly, HOG is a reference point in virtually all the current works about object detection. For that reason and the performance showed in the experiments, HOG was selected for this work as we can see in the next sections.

3 Framework overview

CAVAM model [6] proposes a tile-based method to select ROI. This uses the information of saliency, object familiarity, and temporal familiarity to select the significant tiles belonging to the ROI. Once these tiles have been detected, a detailed object detection is performed only in these regions, reducing the amount of data to process, and in consequence, also reducing the execution time. In Fig. 1, the general diagram of CAVAM model is shown. The original work utilizes SIFT descriptor as detector. In our case, we compare different algorithms in order to find a better option for people detection.

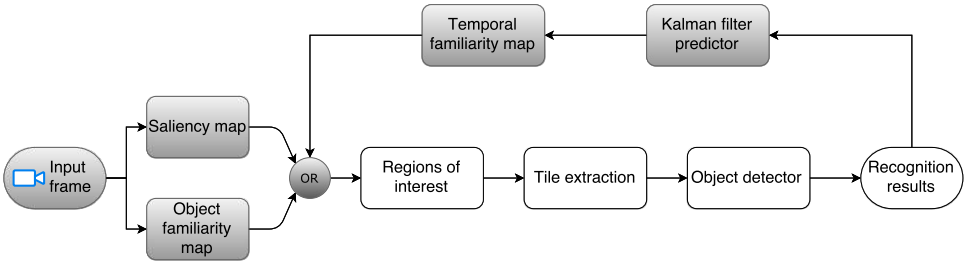


Figure 1: General architecture of CAVAM model. It is composed of two principal sections: the grayed out blocks correspond to the maps calculation to determine the ROI, and the white blocks correspond to the detailed processing of the ROI. In our approach, we change the detection algorithm from SIFT to HOG. In addition, the classification in our model is performed by support vector machines (SVM).

3.1 People detector

In order to propose a specialized method in pedestrian detection, we compare some of the most popular methods in this area: HOG, SIFT, SURF, and HMAX. In this first experiment, we use two classifiers: the nearest neighbor for SIFT and SURF, and SVM for HOG and HMAX. Table 1 shows that HOG and HMAX outperform SIFT and SURF methods in recall and precision. Where recall and precision are defined as follows:

$$Recall = \frac{TP}{TP + FN}, \tag{1}$$

$$Precision = \frac{TP}{TP + FP}. \tag{2}$$

TP, TN, FP, and FN mean true positives, true negatives, false positives and false negatives respectively.

Table 1: Comparison of algorithms for people detection

Method	Recall	Precision
SIFT	83.39%	78.26%
SURF	68%	70.83%
HOG	96%	85.7%
HMAX	94%	97.9%

In addition, we also consider some other studies like the Ref. [18] and Ref.[13] which evaluate a SIFT-SVM approach and a SURF-SVM approach, respectively. The results state an accuracy no bigger than 91%, which do not exceed the accuracy of HOG or HMAX. In general, the best recall-precision performance was obtained by HMAX, but it is by far the slowest algorithm in the test. The Fig. 2 presents the results of the speed comparison among all the methods without considering the classification time, in this way, the principal factor in the results is the complexity of each compared descriptor. Particularly, HOG gets the second best results in precision, the best result in recall and also it is the fastest compared algorithm. For these reasons, we selected HOG method to be implemented in our approach.

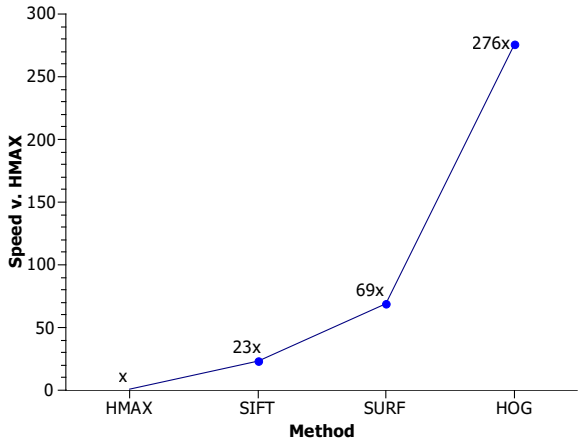


Figure 2: Speed comparison of different algorithms for object detection, in the context of human detection. We take as reference the slowest algorithm (HMAX). The fastest algorithm is HOG, which reaches a speed of 66 tiles per second.

3.2 CAVAM-HOG approach

In Fig. 1, we see the general diagram of the model. The first block is the input frame, which is used to compute the saliency map and the object familiarity map, explained in Sec. 3.2.1 and 3.2.2, respectively. Additionally, the temporal familiarity map gives information about the temporal relation of objects detected in consecutive frames [6]. This is calculated using

the linear Kalman filter method [19] to track and predict the future positions of the detected objects. The Kalman filter is an estimation method which sets a linear model of the system. This consists of two principal stages: observation and prediction. In the observation stage, the system takes location, velocity and acceleration data of the object detected to correct the system state variables. Subsequently, in the prediction stage, an estimation of the next position of the detected object is performed. This prediction is captured on the temporal familiarity map, and finally, once we have computed the three maps, they are joined by a logical OR operation.

3.2.1 Saliency map

The Saliency map was inspired from the capacity of primates to select relevant information of a scene and to spend only a few resources in the remaining information. Currently, there are many methods to calculate the visual salience of an image, in Ref. [20] and Ref. [21] two interesting comparisons of different methods are shown. Some of the most accurate methods take into consideration the global image structure, like those presented in Refs. [22, 23]. However, this kind of methods suffers from combinatorial complexity, hence they are applicable only to images with specific features.

In our work, a fundamental issue is the algorithm speed, therefore, we implemented the Itti method [24] as the original CAVAM model suggests. This method is based on three features: color, intensity, and orientations (see Fig. 3). It is composed of 5 mainly stages: The first stage is a linear operation to decompose the color information of the input image in four broadly-tuned color channels: red, green, blue and yellow. These colors channels were selected because the primates retina is especially sensitive to the red, green and blue lightwaves. Additionally, yellow also has an important impact in attracting human attention as it is shown in the experiments presented in Ref. [25]. Furthermore, the intensity is obtained averaging the three color components of the RGB image, and orientations are calculated using oriented Gabor pyramids of 9 scales and 4 angles (0°, 45°, 90° and 135°).

In the second stage, Gaussian pyramids are computed for each color channel and the intensity channel, these pyramids are calculated using 9 different scales. In this way, we obtain a total of 72 maps: 8 for each color channel, 8 for intensity channel and 32 for orientation pyramids.

Typically, visual neurons are most sensitive in the center of the visual space and the neuronal response decreases in the surround. This kind of center-surround architecture is suitable to detect objects which stand out from their environment. The center-surround design is implemented in the model as the difference between fine and coarse scales producing the across scale difference operation. This operation interpolates the finer scale, where the central pixel is located, and subtracts pixel by pixel of the surround located on the coarse scales. Thus, we reduce the number of maps to 12 for color, 6 for intensity and 24 for orientations. Subsequently, in the fourth stage, all the maps are reduced to only three, by computing across scale addition, which scales down each map to scale four and adds each element point by point. Finally, in the last stage, we normalize and average the remaining three maps. Itti algorithm is performed in a time $O(kN)$, where k is the size of the Gabor filter and N is the number of pixels of the image. This order is determined by the calculation of the Gabor pyramids because this is the most computationally expensive part of the algorithm. A detailed description of the calculation process of the saliency map is explained in Ref. [24], we only give an overview in order to introduce our approach.

Moreover, saliency map is one of the principal parameters that determine the ROI in CAVAM model, unfortunately, this calculation increases the computational cost and execution time of the algorithm. To reduce these increments, we suggest decreasing the image scale before computing the saliency map. This means less amount of data to process and consequently a lower execution time. To measure the size reduction effect, we tested 404 images in three different scales: 640x480 pixels (full image), 320x240 pixels (75% reduced) and 171x128 pixels (93% reduced). The Table 2 shows the results in time and miss rate.

Table 2: Comparison of saliency map with different image sizes

Image size	% Reduction	Average time	Miss rate
640x480	full size	0.8606s	15%
320x240	75%	0.7631s	10.78%
171x128	93%	0.7364s	11.2%

In this case, the miss rate or false negative rate is a fundamental metric, it indicates the number of people leaving outside the ROI, which directly affects the method accuracy. By the contrary, the false positives or false detections could be corrected in the detailed processing of the ROI. Thus, in this section, the miss rate is considered the primary measure to evaluate this stage.

As expected, the saliency map calculation is faster

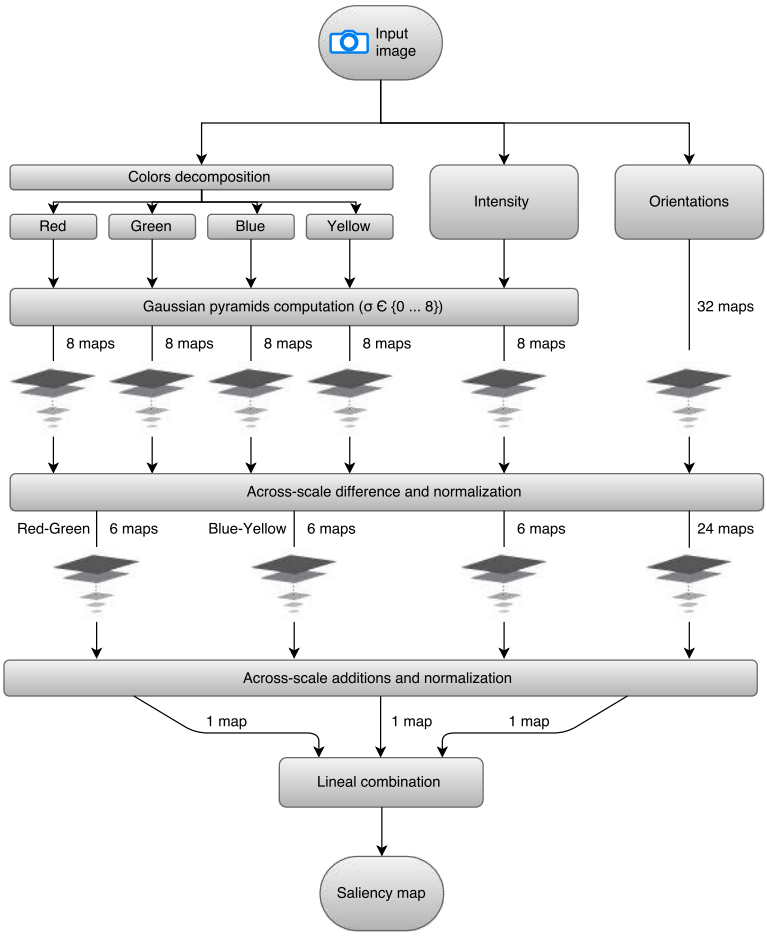


Figure 3: General architecture to compute the saliency map (detailed description in Ref. [24]).

in smaller images. However, while the difference in time between the third scale and the second is only 0.027 seconds, the difference between the second and the first scale is more than 3 times bigger. All this time will be accumulated by each frame of the image sequence, resulting in a significant factor in the algorithm speed. Additionally, the miss rate of the second scale is the lowest. Taking all this into consideration, the second scale (240x320 pixels) was selected, which reduces the original image 75%.

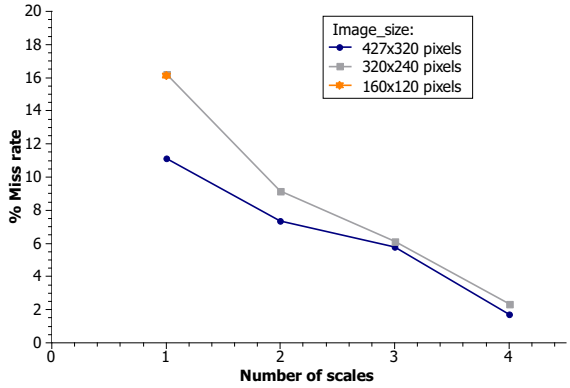
Even if the saliency map does not achieve a very low miss rate by itself, we must consider that the final ROI are obtained by using two more maps (familiarity map and temporal familiarity map), which help to improve the final result.

3.2.2 Object familiarity map

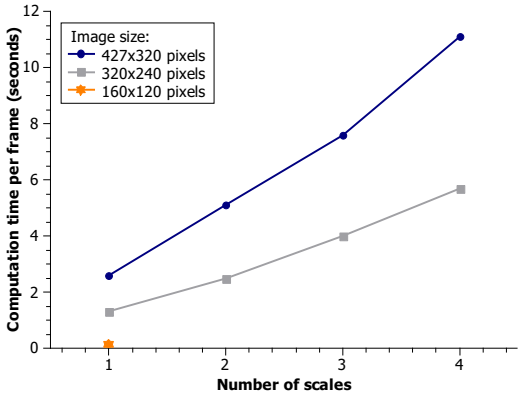
Familiarity is defined as a similarity measure of the input image features with the object that we want to detect. It is used to guide the attention to the region that probably contains the object of interest. In previous works, like Ref. [26], familiarity is applied to SIFT method by using intermediate results of the detection process. In our work, the algorithm proposed in Ref. [26] is modified in order to adapt to HOG descriptor. Specifically, we need to obtain a rough and fast result of HOG algorithm. Thus, different reductions of the image size were tested. This operation has a direct effect in the computation time of HOG, because the frame size affects the tile size, and therefore this decreases the number of blocks in which the tile is divided. Despite it is possible changing the blocks and cells sizes used to compute HOG descriptor, we consider that keeping the same sizes used in the detailed detection stage is enough for the purpose of this map.

In addition, we propose decreasing the number of detection scales. Since HOG is a tile-based detector, each frame processed is scanned by tiles at all positions and scales possible, then a descriptor to every tile has to be computed. This makes HOG computationally expensive. For this reason and in order to simplify this stage, the image reduction is combined with a decrease in the number of scales. Three different reductions in the image size and 4 different and equally

spaced scales were tested. The experiments were performed with 300 images of TUD-datasets [27] and the results are shown in Fig. 4.



(a) Miss rate



(b) Time comparison

Figure 4: Results of the different variations of the object familiarity map. Figure 4a and 4b show the miss rate results and speed of each different configuration, respectively.

Our experiments show that a bigger image resolution and a greater number of scales decreases the miss rate, but the execution time is increased. Therefore, taking into account the degradation of the results, a tradeoff between time and miss rate was chosen. Specifically, we selected reducing the images size to 320x240 pixels and computing HOG algorithm using 3 detection scales. As a result, the execution time decrease 93 times from the original algorithm with less than 5% of miss rate increment, which could be improved by the other maps to compute ROI. The diagram of familiarity map is shown in Fig. 5.

3.2.3 Tiles thresholding

CAVAM model is a tile-based method, this means that the image is divided into tiles. As expected, not all the tiles are completely inside of the ROI. While the tiles completely outside of the ROI are discarded and the tiles completely inside are kept, the tiles half inside and half outside of the ROI have to be classified. We count the number of pixels belonging at the ROI and we establish a threshold, if this sum of pixels is greater than the threshold, the tile is kept, otherwise, it is discarded. This threshold is a fundamental parameter to regulate the performance of our approach. It is explored in Sec. 4.1.

After the tile selection, HOG descriptor is computed from each tile retained in the ROI. We select a cell size of 8x8 pixels and a block size of 2x2 cells, according to the results of Ref. [1]. Finally, the descriptors are classified by SVM method.

4 Experiments and analysis results

All our experiments were performed in Matlab R2015 in a computer with a CPU Intel i7-3610QM @ 2.3 GHz with 8GB RAM. The previous experiments to determine the algorithm for people detection and the parameters of saliency map were performed using INRIA and $GRAZ\_02$ dataset of the Graz University of Technology. In the next experiments, we tested the complete approach and we used six different image sequences of TUD dataset of the Marx Plank Institute computing department [27, 28, 29]. We evaluated and compared the different variations of our approach with the traditional HOG algorithm. Additionally, in Sec. 4.2, a global comparison of CAVAM-HOG with other people detection approaches is shown.

4.1 HOG vs CAVAM-HOG comparison

In the next experiments, we vary the threshold for classifying tiles belonging to the ROI, with the goal to test all the possible configuration of our method. This threshold helps to decide if a tile, divided by the limit of the ROI, is part of this region or not (see Sec. 3.2.3). If the sum of the tile pixels that are inside of the ROI is bigger than the threshold, this tile is considered part of the ROI, and otherwise, it is discarded. Five different threshold values are tested, from 50% to 90% of the total size of the tile.

In previous sections, we already defined recall and precision metrics, but we also evaluate our model using the accuracy and specificity, which are calculated according to Eqs. (3) and (4), respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Specificity = \frac{FN}{FN + TP} \tag{4}$$

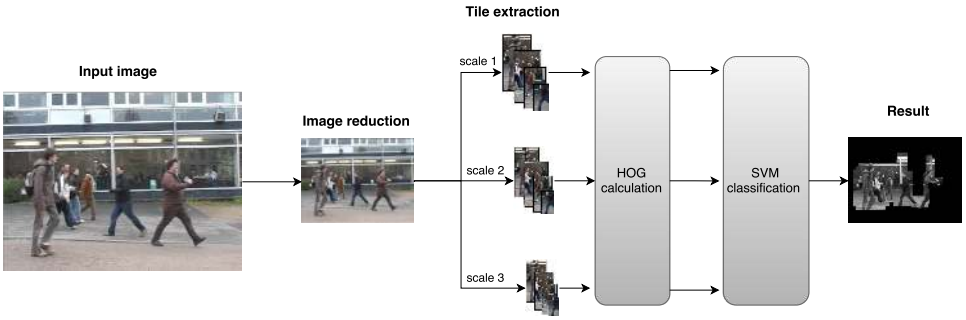


Figure 5: General diagram of the object familiarity map architecture.

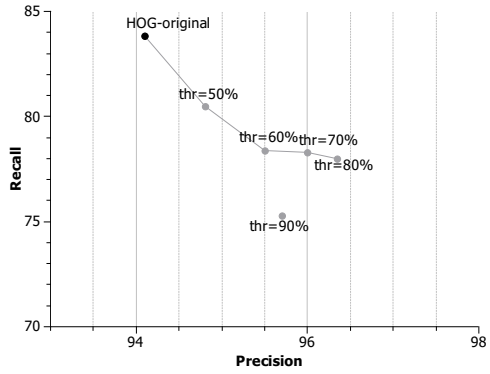
The comparison between the original method and different variations of our approach shows similar performances in specificity, accuracy, and precision, but we can notice some difference in the recall. In Fig. 6a we observe the recall-precision curve. On one hand, the recall decreases when the threshold of the ROI increases, but on the other hand, the precision improves. The only exception is the approach corresponding to the 90% threshold, when the method performance starts to decline. Nevertheless, in the majority of the metrics and sequences used to compare our approach with the original method, the results are close and even in some cases CAVAM-HOG outperforms HOG results.

Figure 6b shows the average results of all the sequences evaluated, in which we see a difference in recall of up to 5% between the original HOG method and our approach. This difference can be reduced according to the selected threshold and the characteristics of the image sequences. In general, the image sequences that have smaller objects of interest present the worst recall performances. However, we successfully detect people at a size up to 50x100 pixels. In contrast, precision and specificity outperform the original HOG method in all the CAVAM-HOG approaches. In some of the best cases, CAVAM-HOG has achieved up to 3.9% and 3.8% better precision and specificity, respectively. In addition, the accuracy shows similar results than the original method, except for CAVAM-HOG at 50% where the results start to deteriorate.

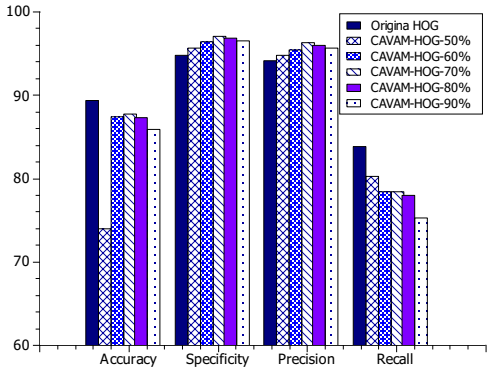
Moreover, computing HOG algorithm for a tile composed of N pixels and with a block size b takes $O(Nb^2)$ time. If we extrapolate this time to a complete image, we must multiply this result by a factor F. This factor represents the total number of tiles produced by all the different scales of detection, resulting in an order of $O(FNb^2)$.

In contrast, our approach is composed of four principal stages: saliency map, temporal familiarity map, object familiarity map and detailed HOG computation. The first stage (saliency map), has an order of $O(kN)$, as mentioned in Sec. 3.2.1, k is the size of the Gabor filter and N is the number of pixels of the image. Nevertheless, in our approximation the size of the images is reduced 75%, resulting in $O(kN/4)$. The second stage is the temporal familiarity map, this is calculated using the linear Kalman filter method, thus, its run time complexity is also linear. The next stage is the object familiarity map, which is computed using HOG algorithm, but reducing the size of the image and the number of detection scales to 3. Therefore, we can set the time complexity in $O(fNb^2/4)$, because the image size is reduced 75% and f is the number of tiles belonging only to the 3 detection scales processed. Finally, HOG algorithm is computed in the ROI, this process has a time complexity of $O(f'Nb^2)$, where f' is the number of tiles belonging to the ROI.

Once we have contemplated all the complexities of each part of our approach, we can see that they are dominated by the complexities of the object familiarity map and the HOG computation in the ROI. Hence, CAVAM-HOG requires $O((f/4 + f')Nb^2)$ time. If we compare this time with the complexity of the original HOG algorithm $O(FNb^2)$, it is clear that our approach is only useful if $F > (f/4 + f')$, where f' is directly linked to the tile classification threshold. According to our experiments, we observe a general acceleration of the method, this acceleration varies in accordance with the different characteristics of each image set that produce different results in the ROI calculation. While the worst case (CAVAM-HOG 50%) obtains an acceleration only 1.2 times faster than the original method, the best case (CAVAM-HOG 90%) is up to 3.3 times faster. Figure 7a, shows the number of times that each approach with different threshold contributes to speed up HOG algorithm, and Fig. 7b shows the average speed of each approach in pixel per second (pps). As we see, CAVAM-HOG outperforms the original method in all our experiments. This indicates that in all these cases $(f/4 + f\prime)$ is smaller than F and the possibility to find an opposite case in images for pedestrian detection is negligible, especially

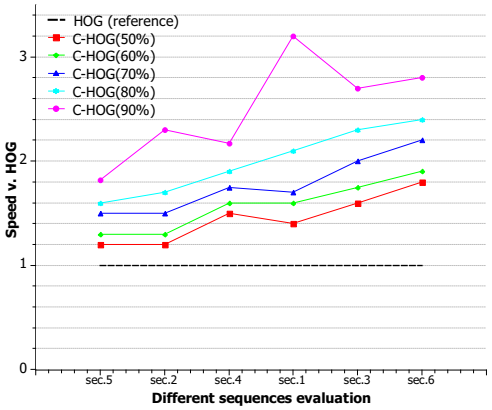


(a) recall-precision

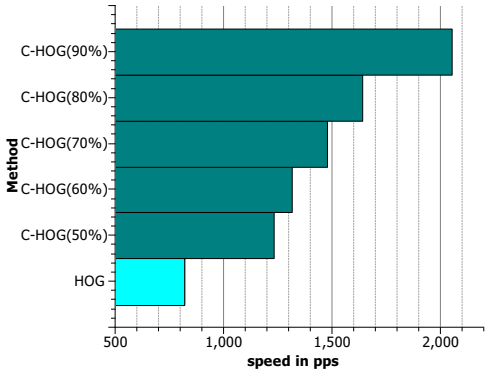


(b) Average

Figure 6: Experiments results of the different configurations of CAVAM-HOG method compared to the original HOG algorithm. Each configuration is tested with a different threshold value (from 50% to 90%). Figure 6a shows the recall-precision curve and the Fig. 6b presents the average performance of each approach.



(a) Speed comparison of all sequences



(b) Average speed

Figure 7: Figure 7a shows the speed variation of the different CAVAM-HOG configurations (denoted as C-HOG) in the diverse evaluated sequences. The dotted line represents the speed of the original HOG algorithm, which is used as reference. In Fig. 7b we see the average speed of each approach in pixels per second, including original HOG method.

if we carefully select the tile classification threshold.

In this section, we have evaluated and compared our approach CAVAM-HOG with the original HOG method. Furthermore, the response produced by the variation of the threshold for the tiles classification was examined. We can notice that generally, the best performance is carried out by CAVAM-HOG with a threshold value of 80%. Because although on average, CAVAM-HOG (70%) has slightly better results, in the individual evaluations this is outperformed, in most of the cases, by CAVAM-HOG (80%). In addition, utilizing a higher threshold increments the algorithm speed.

4.2 Other methods comparison

In previous sections, we have reviewed some method for accelerating people detection, they include methods like SURF [4], BRISK [11], FREAK [12] and Modified HOG [10]. In Ref. [13], a comparison of SURF, BRISK and FREAK in the context of pedestrian detection is explored. Thus, in order to compare the results with our approach, we performed an evaluation of the HOG method with the same dataset (NICTA) [30]. The accuracy comparison is shown in Table 3. As we can see, HOG method gets the best accuracy, but while the methods belonging to SIFT and SURF family show a decreased accuracy above 5%, our approach differs only 2% from the original HOG algorithm.

Furthermore, Modified-HOG [10] is an acceleration method which reduces the dimension of the features in the descriptor computation. This reports 5% of decreased accuracy from the original HOG algorithm, which is higher than our 2% of accuracy reduction. Additionally, the execution time compari-

Table 3: Accuracy comparison in methods for accelerating people detection.

Method	Accuracy
SURF	85%
BRISK	87%
FREAK	91%
HOG	96.85%
CAVAM-HOG(80%)	95%

son is shown in Fig. 8. The results show that we achieved the fastest execution time with a speed 2x faster than the original HOG method, followed by Modified-HOG. In this way, we achieve interesting results in performance and execution time.

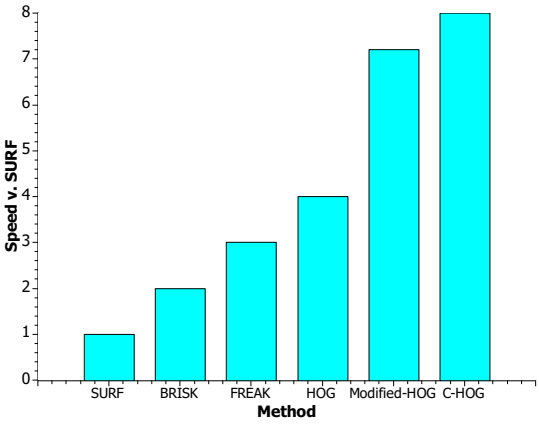


Figure 8: Speed comparison among four different acceleration methods for people detection and our approach CAVAM-HOG (80%). The y-axis represents the number of times that each algorithm is faster than SURF, which is the slowest algorithm compared.

5 Conclusions

Our approach was tested on images sequences for pedestrian detection, in which we achieved similar results as the traditional HOG algorithm, and even we improve specificity and precision. These improvements are the consequence of two principal features of CAVAM: the first is the introduction of Kalman filter in the model, and the second is to focus the processing in the ROI. On one hand, the Kalman filter helps to identify easily the detected people in previous frames, even if HOG detector has missed these persons in the current frame. On the other hand, when the processing is limited to the ROI, the amount of useless information is reduced. This helps to decrease the false positive rate and consequently improves the accuracy, precision, and specificity.

In conclusion, choosing an appropriate threshold, our approach achieves almost the same or better results than the original HOG method, but we decrease execution time. The ROIs help to reduce the amount of data to process, and in consequence, the algorithm speed increments. As mentioned above, we suggest using a threshold value of 80%, which reduces by half the execution time. On average, CAVAM-HOG (80%) decreases the accuracy no more than 2%, but the specificity and precision increase in equal measure.

Additionally, we show that CAVAM model is not restricted to be used only with SIFT algorithm, as it was originally designed. We show that it can be adapted and implemented to other kinds of algorithms like HOG method. Therefore, our approach opens up the possibility to expand the applications of the CAVAM model.

In addition, our future work includes the implementation of this approach in hardware, with the goal to obtain a reconfigurable people detector in real time.

Acknowledgements: The authors would like to express their gratitude to the Mexican National Council for Science and Technology (CONACYT) and University Bourgogne Franche Comte for financing this work. .

References:

[1] N. Dalal and B. Triggs, *Histograms of Oriented Gradients for Human Detection,* IEEE Trans. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 1, 886–893, 2005.

[2] M. Riesenhuber and T. Poggio, *Hierarchical models of object recognition in cortex,* Nature neuroscience 2(11), 1019–25, 1999.

[3] D. G. Lowe, *Sift,* in Computer vision. The proceedings of the seventh IEEE international conference on, 2, 1150–1157, 1999.

[4] H. Bay, T. Tuytelaars, and L. Van Gool, *SURF: Speeded up robust features,* Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3951 LNCS(Chapter 32), 404–417, 2006.

[5] M. Szarvas, a. Yoshizawa, M. Yamamoto, et al., *Pedestrian detection with convolutional neural*

networks, IEEE Proceedings. Intelligent Vehicles Symposium, 224–229, 2005.

[6] J. Oh, G. Kim, J. Park, et al., *A 320 mW 342 GOPS real-time dynamic object recognition processor for HD 720p video streams,* IEEE Journal of Solid-State Circuits 48(1), 33–45, 2013.

[7] S. Zhang, D. A. Klein, C. Bauckhage, et al., *Fast moving pedestrian detection based on motion segmentation and new motion features,* Multimedia Tools and Applications 75, 6263–6282, 2016.

[8] S. Tang, M. Andriluka, and B. Schiele, *Detection and tracking of occluded people,* International Journal of Computer Vision 110(1), 58–69, 2014.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan, *A discriminatively trained, multiscale, deformable part model,* 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR , 1–8, 2008.

[10] W. Li, H. Su, F. Pan, et al., *A fast pedestrian detection via modified HOG feature,* Chinese Control Conference, CCC 2015-Septe, 3870–3873, 2015.

[11] S. Leutenegger, M. Chli, and R. Siegwart, *fBRISKg: Binary Robust Invariance Scalable Keypoints,* Proceedings of the International Conference on Computer Vision (fICCVg) , 2548–2555, 2011.

[12] A. Alahi, R. Ortiz, and P. Vandergheynst, *fFREAKg: Fast Retina Keypoint,* fIEEEg Int. Conf. Computer Vision and Pattern Recognition (CVPR) , 510–517, 2012.

[13] C. Schaeffer, *A Comparison of Keypoint Descriptors in the Context of Pedestrian Detection: FREAK vs. SURF vs. BRISK,* tech. rep., Stanford University CS Department, California, 2012.

[14] L. Wang and B. Zhang, *Boosting-Like Deep Learning For Pedestrian Detection,* arXiv, 2015.

[15] X. Chen, P. Wei, W. Ke, et al., *Pedestrian Detection with Deep Convolutional Neural Network,* 354–365, 2015.

[16] S. Tang, M. Ye, C. Zhu, et al., *Adaptive pedestrian detection using convolutional neural network with dynamically adjusted classifier,* Journal of Electronic Imaging 26(1), 013012, 2017.

[17] A. Suleiman, Y. H. Chen, J. Emer, et al., *Towards closing the energy gap between HOG and CNN features for embedded vision,* Proceedings - IEEE International Symposium on Circuits and Systems, 2017.

[18] T. Nguyen, E.-a. Park, J. Han, et al., *Object Detection Using Scale Invariant Feature Transform,* Genetic and Evolutionary Computing , 65–72, 2014. [

[19] R. E. Kalman, *A New Approach to Linear Filtering and Prediction Problems 1,* Journal of Fluids Engineering 82, 35–45, 1960.

[20] F. Perazzi, P. Krahenbuhl, Y. Pritch, et al., *Saliency Filters : Contrast Based Filtering for Salient Region Detection,* IEEE Trans. Computer Vision and Pattern Recognition (CVPR), 733–740, 2012.

[21] M.-m. Cheng, N. J. Mitra, X. Huang, et al., *Global Contrast based Salient Region Detection,* IEEE Transactions on Pattern Analysis and Machine Intelligence 37(3), 569–582, 2015.

[22] S. Goferman, L. Zelnik-Manor, and A. Tal, *Context-Aware Saliency Detection,* IEEE Transactions on Pattern Analysis and Machine Intelligence 34(10), 1915 – 1926, 2012.

[23] T. Liu, Z. Yuan, J. Sun, et al., *Learning to Detect a Salient Object,* IEEE Transactions on Pattern Analysis and Machine Intelligence 33(2), 353–367, 2011.

[24] L. Itti, C. Koch, and E. Niebur, *A Model of Saliency-Based Visual Attention for Rapid Scene Analysis,* IEEE Transactions on Automatic Control 20(11), 1254–1259, 1998.

[25] E. D. Gelasca, D. Tomasic, and T. Ebrahimi, *Which colors best catch your eyes: a subjective study of color saliency,* First International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-05 , 16, 2005.

[26] S. Lee, K. Kim, J. Y. Kim, et al., *Familiarity based unified visual attention model for fast and robust object recognition,* Pattern Recognition 43(3), 1116–1128, 2010.

[27] M. Andriluka, S. Roth, and B. Schiele, *People-tracking-by-detection and people-detectionby-tracking,* IEEE Conference on Computer Vision and Pattern Recognition, 1–8, 2008.

[28] C. Wojek, S. Walk, and B. Schiele, *Multi-cue onboard pedestrian detection,* in Computer Vision and Pattern Recognition, CVPR, 794–801, 2009.

[29] M. Andriluka, S. Roth, and B. Schiele, *Pictorial structures revisited: People detection and articulated pose estimation,* IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 1014–1021, 2009.

[30] G. Overett, L. Petersson, N. Brewer, et al., *A new pedestrian dataset for supervised learning,* IEEE Trans. Intelligent Vehicles Symposium, 373–378, 2008.