Muti-scale feature extraction for vehicle detection using PHis-LBP

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

METZLI RAMIREZ-MARTINEZ
Univ. Bourgogne Franche Comté
DRIVE EA1859 Laboratory
Nevers, FRANCE
Metzli_Ramirez-Martinez@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: Multi-resolution object detection faces several drawbacks including its high dimensionality produced by a richer image representation in different channels or scales. In this paper, we propose a robust and lightweight multi-resolution method for vehicle detection using local binary patterns (LBP) as channel feature. Algorithm acceleration is done using LBP histograms instead of multi-scale feature maps and by extrapolating nearby scales to avoid computing each scale. We produce a feature descriptor capable of reaching a similar precision to other computationally more complex algorithms but reducing its size from 10 to 800 times. Finally, experiments show that our method can obtain accurate and considerably faster performance than state-of-the-art methods on vehicles datasets.

Key–Words: Feature extraction, texture, vehicle detection, Local Binary Patterns, features pyramids

1 Introduction

Nowadays, it is becoming imperative to furnish a sensorial capacity onto intelligent systems like the ones human beings possess. Among various sensors, our most powerful tool to extract information from environment is sight. Computer vision is in charge of trying to replicate the different processes performed to extract and evaluate information. However, processing and classifying images is a really challenging task due to a wide range of changeability in real-life images. The most common causes of low homogeneity between images can range from defects capturing images to more complicated problems such as occlusions, non-rigid deformations, viewpoint and lighting changes[17].

One way to homogenize images is extracting features that are not visible to the human eye, these features are encoded in a manner that provide single or multi-scale information about a part of an image. Among the most used features for different purposes of detection, we find Scale Invariant Features Transform (SIFT)[11], Speeded-Up Robust Features (SURF)[12], Histograms of Oriented Gradients (HOG)[13], gradient magnitude and Local Binary Patterns(LBP)[20]. SIFT and SURF extract information from Laplacian of Gaussian of an image but in different ways, HOG uses gradient directions of smaller patches to describe an image. Finally, LBP employs fundamental properties of local image texture.

Texture classification has been a subject of a strong research in computer vision because it plays an important role in a wide variety of applications. The development of texture descriptors, such as LBP, have headed in an important direction by using alternative texture extraction methods with different problems including image segmentation, object recognition, image retrieval, aerial surveillance and pedestrian detection[8]. Texture is defined as a statistical measure of the intensity variation over an area, which quantifies properties such as smoothness and surface reflectance[9]. Compared to the color of images, texture requires an extra processing step to obtain a representation for each descriptor. Methods for texture recognition are basically divided in four categories: structural methods, statistical methods, mathematical methods and signal processing based methods[9].

An on-road vision-based system is defined by a three stage pipeline[1]: detection, tracking and behavior. Vehicle detection is the first approach for higher order tasks and involves different levels of complexity

due to the scene structure on mobile platforms. Essentially, vehicle detection needs to detect, recognize and localize a specific object and operate at frame rates that reaches real time performance in order to provide a margin time for making critical decisions in higher tasks. However, vehicles present a high variability in distinct aspects like: orientation, color, size and shape. For this reason, it is necessary to find a robust and uniform way to represent objects in the images. Typical appearance descriptors for vehicle detection include SURF[3], HOG[4][6][5] and HAAR-like features[7][2]. In this paper, we propose a multi-scale approach for vehicle detection using Local Binary Patterns. First, histograms of LBP are computed over a single scale per octave and a feature descriptor is formed by approximating remaining histograms of features through a power law. PHis-LBP is designed to be lightweight, robust and ideal for processing platforms with limited resources because only a certain number of channels have to be computed, improving processing time and memory space.
Experimental results show that PHis-LBP is smaller (memory space), faster and has a similar detection performance than its real counterpart (computing LBP at each scale) and other feature descriptors. The rest of this work is organized as follows: Section 2 presents a review of methods used for computing PHis-LBP. Formulation and description of the proposed approach is presented in Section 3. Section 4 describes dataset employed and evaluation method for PHis-LBP. The discussion of results and conclusions are made in Section 5.

2 Related work

2.1 Local Binary Patterns

This section give us a brief introduction to Local Binary Patterns (LBP), also we show its advantages and disadvantages. The general idea behind LBP consists in translating information in an image by mapping each pixel and its surroundings, creating micro patterns that contain information about edges and other local features[16].
The first version of LBP was introduced by[20] and is defined as follows:

$$LBP(P,R) = \sum_{n=0}^{P-1} s(d_c - d_n)2^n \qquad (1)$$

Where:

$$s(d) = \begin{cases} 0 & d < 0 \\ 1 & d \geq 0 \end{cases} \qquad (2)$$

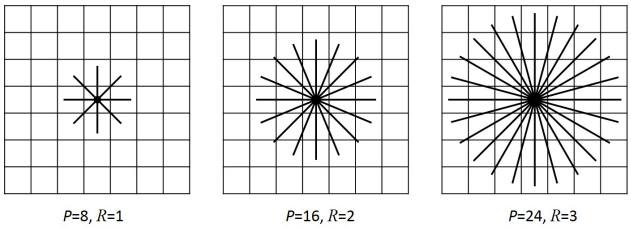

Figure 1: Local Neighborhoods with distinct parameters P and R.

Every local neighborhood is formed by a gray value of central pixel d_c and d_n corresponds to the gray values of P neighbors located on the perimeter of a circle with radius R $(R > 0)$ and spaced equidistant from each other. The position of every single one neighbor can be determined by equation (3). Neighbors that do not fit in the center of pixels are estimated using an interpolation method. Fig. 1 shows different local neighborhoods varying P and R.

$$(x_d, y_d) = (-R\sin\frac{2\pi n}{P}, R\cos\frac{2\pi n}{P}) \qquad (3)$$

LBP has two important features making it ideal for object detection, gray-scale (lighting changes) and rotation invariance. The first is accomplished by subtracting gray value of the neighboring pixels to central pixel and separating them in lighter or darker pixels. If a lighting change occurs, the difference between lighter and darker pixels will remain constant relative to the central pixel since all will suffer same gray level variation. Every member of neighbor set $(d_0$ to $d_n)$ is thresholded according to equation (1).

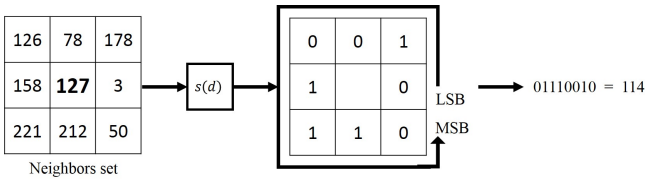

Figure 2: LBP encoding process.

LBP produces $2^P - 1$ different patterns, always first element of neighbors set (d_0) is assigned to the right of central pixel d_c and remaining elements are encoded counter-clockwise as in Fig. 2.
LBP achieves rotation invariance replacing $2^P - 1$ patterns by a reduced group called uniform patterns. Classification between uniform and non-uniform patterns is done by defining a uniformity measure U, indicating the number of spatial transitions, *i.e.* indicates how many bitwise changes *(0 to 1 or inversely)* occur in a pattern.
Only patterns with $U \leq 2$ are considered as uniform,

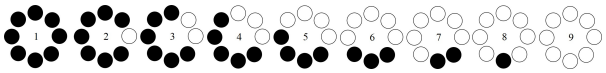

Figure 3: Uniform patterns of LBP (8, R).

typically in $LBP(P,R)$ exists $P+1$ uniform patterns and are labelled with the number of "1" in the pattern, non-uniform patterns are grouped under a same label: $P+2$. $LBP(8,R)$ uniform patterns are shown in Fig. 3, pattern 1 represents a flat area, pattern 9 a bright spot and other $P-1$ patterns are used to represent different properties of the image like edges and corners. Finally, a histogram of $P+2$ comprising uniform and non-uniform patterns, is employed to represent texture feature of an image.

2.2 Feature pyramids

A set of properties for multi-scale features are defined in[18], Let Φ be a function that takes an image I and applies any low-level shift invariant transformation. Φ outputs a new channel image $C = \Phi(I)$ containing a feature map, each pixel in C is estimated from a patch of I. Also, the global mean of features computed over I is represented by $f_\Phi(I)$. A traditional framework to obtain a feature map of a downsampled image is given by equation (4). It follows a straightforward path, downsample (D) an image by a factor s and right away computing its feature map C_s.

$$C_s = \Phi(D(I,s)) \tag{4}$$

Piotr et al.[19] propose a fast method for estimate new channels C_s from image I of size m x n downsampled by a factor s using the following approach:

$$\frac{f_\Phi(I_{s1})}{f_\Phi(I_{s2})} = (\frac{s_1}{s_2})^{(-\lambda_\Phi)} + \epsilon \tag{5}$$

If we set $s_1 = s$ as our scale of channel to be computed and $s_2 = 1$ as scale of image I, we can infer an approximation for C_s as follows:

$$f_\Phi(I_s) \approx f_\Phi(I) s^{(-\lambda_\Phi)} \tag{6}$$

But, f_Φ represents a global mean of features in channel C:

$$\frac{1}{m_s n_s}\sum_{i,j} C_s(i,j) \approx \frac{1}{mn}\sum_{i,j} C(i,j) s^{(-\lambda_\Phi)} \tag{7}$$

$$C_s \approx D(C,s) s^{(-\lambda_\Phi)} \tag{8}$$

Finally, we can obtain an approximation of a downsampled channel only by downsampling a higher scale channel and multiplying it by a scale factor as we can observe in Eq.(8). Proposed method by Piotr et al. and typical framework for channel scaling are shown in Fig. 4.

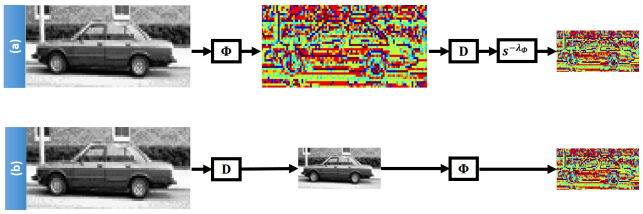

Figure 4: (a) Channel scaling using channel approximation, (b) Channel scaling by traditional framework.

Also, Piotr et al. propose a method to obtain a feature pyramid through channel scaling by modifying Equation (8), this method produces an approximation using a higher or lower channel scale for reference as follows:

$$C_s \approx D(C_R, \frac{s}{s^R})(\frac{s}{s^R})^{(-\lambda_\Phi)} \tag{9}$$

Where $s^R \in \{1, 1/2, 1/4, \ldots\}$, s^R represents a computed real channel used to approximate a number of downscaled channels per octave. C_s chooses the nearest scale s^R to provide an efficient trade-off between computation speed and accuracy. Feature pyramids computed by channel scaling are shown in Fig. 5, this method shows increased speed but lesser accuracy than traditional framework. However, achieved accuracy is enough for object detection.

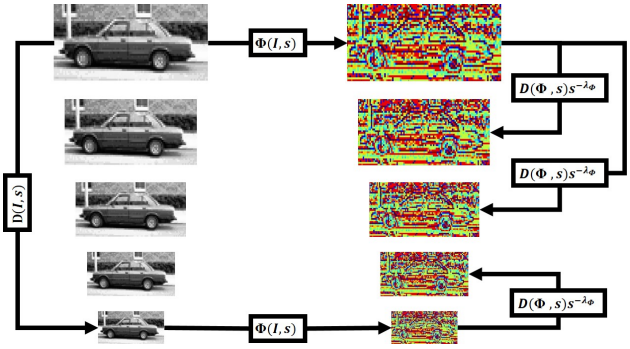

Figure 5: Feature pyramids approximation.

3 PHis-LBP: Pyramidal Histograms of Local Binary Patterns

Multi-resolution approaches for object detection are based on a basic principle, make richer representations of objects in different channel types and sizes. However, it implies that computation time and storage will increase according to size of scale represen-

tation. Piotr et al.[19] found that certain kinds of channels, especially those related to gradient computation, preserve its structure through resampled images. We show below the related concepts behind pyramidal representation for histograms of LBP.

We start defining LBP as a first-order circular derivative of an *m x n* discrete signal, where each pattern is formed by concatenating the binary gradient direction of a specific neighboring set. In other words, LBP delivers a histogram of patterns that measures the distribution of gradient directions. Gradient magnitude and direction is defined as follows:

$$M(i,j)_G = \frac{\partial I}{\partial y}(i,j)^2 + \frac{\partial I}{\partial x}(i,j)^2 \tag{10}$$

$$D(i,j)_G = \arctan(\frac{\partial I}{\partial y}(i,j), \frac{\partial I}{\partial x}(i,j)) \tag{11}$$

Normally, a gradient histogram of an image is estimated by a pixel-wise voting, where gradient direction is quantized into *q* bins and weighted according to an indicator function F_v and gradient direction *D*. Each voting for *qth* bin of gradient histogram is shown in equation(12).

$$h_q = \sum_{i,j} qD(i,j)_G F_v \tag{12}$$

We observe from equation (1) and (12) that a LBP histogram can be interpreted as a gradient histogram, every pixel contributes with a pattern that is weighted according to gradient directions of a neighboring set. Gradient direction D_G around a neighborhood is quantized into *P* bins and each bin is weighted as in equation (1). Then, each LBP pattern provides an unitary vote to the LBP histogram *H*.

We could infer automatically that the structure of upsampled and downsampled images I' are related to its original scale image I by a constant factor *k*. Otherwise, our interest is to determine how features generated from images are propagated to upsampled or downsampled channels. Consequently and using the definition of derivative, it is demonstrated that $\frac{\partial I'}{\partial x} = \frac{1}{k}\frac{\partial I}{\partial x}$ and $\frac{\partial I'}{\partial y} = \frac{1}{k}\frac{\partial I}{\partial y}$ are also related by a sampling factor. We can extend this approach to 2D discrete signals and find the same relation for gradient magnitude and direction. Therefore, gradient histograms hold the same relation between histograms computed from I' and I as is shown in equation (13).

$$h'_q = kh_q \tag{13}$$

Re-ordering for LBP histogram, we obtain:

$$H' = \sum_{q=1}^{P+2} h'_q = \sum_{q=1}^{P+2} kh_q = kH \tag{14}$$

Previous definition is required to prove that a LBP histogram is an approximation of the same histogram but computed in a higher or lower scale. An approximation for any feature channel is done according to equation (6), where $f_\Phi(I)$ represents a global mean of features in an image, and rearranging it in terms of LBP histogram is shown in equation (15). Image size is represented by *m* and *n*, *q* refers to each value assigned to uniform patterns in LBP histogram

$$f_\Phi(I) \approx \frac{1}{mn}\sum_{q=1}^{P+2} qh_q \tag{15}$$

Substituting equation (15) into equation (6), we obtain:

$$\frac{1}{m'n'}\sum_{q=1}^{P+2} qh'_q \approx \frac{1}{mn}\sum_{q=1}^{P+2} qh_q s^{-\lambda_\Phi} \tag{16}$$

From equation (16), we deduce a similar relationship as in equation (8):

$$h'_q \approx D(h_q, s)s^{-\lambda_\Phi} \tag{17}$$

$$H' \approx D(H, s)s^{-\lambda_\Phi} \tag{18}$$

According to equation (9), we construct a pyramidal representation by downsampling histograms of LBP. Workflow of PHis-LBP is shown in Fig. 6, where LBP features are computed only in highest and lowest scale in an octave, then these are used to model other histograms. A downsampled histogram is equivalent to multiply original histogram by its desired scale s, thus we avoid downsampling images or channels in order to save computation time.

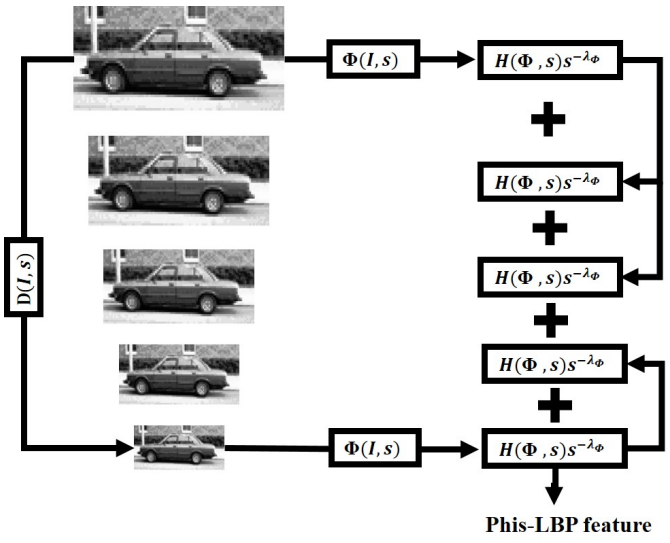

Figure 6: Feature creation process of PHis-LBP.

4 Experimental results

We tested our proposed method using UIUC Image Database for Car Detection[15] which contains side view images of cars. Proposed experiments are implemented using MATLAB software on an Intel Core i5 PC with 8 GB RAM. The evaluation is done using dataset provided by [15], this dataset provides 700 images of side views of cars and 500 non-car images. We established our training set as 400 positive images and 300 negative images. Testing set is defined by 200 images of multi-scale cars and 200 images of non-car images.
We used an adaptive boosting [14] to train and classify our vectors generated by different algorithms including PHis-LBP and its non-approximated counterpart (P-LBP). Evaluation was done using Percentage of Wrong Classifications (PWC) as metric. PWC is defined as follows:

$$PWC = 100 * \frac{FN + FP}{TP + FN + FP + TN} \quad (19)$$

Where, wrong classifications (False Negatives and False Positives) are divided by the total population of classifications (True positives, False Negatives, False Positives, and True Negatives).

4.1 Neighbors and Radius

In this section, we evaluate different parameters used in the LBP algorithm and their relation to vehicle detection performance. The first parameter analyzed is the number of neighbors P used to describe an LBP pattern. In traditional LBP algorithm, P is proportionally related with performance because we can represent better an image section with a bigger number of points in the neighborhood. This behavior is shown in Fig. 7.
Conversely, as we increase size of neighborhood P, performance in P-LBP and PHis-LBP is decreased. This gradual reduction is due to interpolation of uniform patterns, an interpolated uniform pattern value has more options to choose an approximate value in a bigger neighborhood set and consequently is placed erroneously as another pattern. Interpolation effect in both multi-scale approaches is shown in Fig. 8.
The next parameter involved in LBP is the distance between central pixel and its neighbors. Liao et al.[10] discovered that LBP used in face recognition decreases its performance by increasing radius R, this relation promotes the creation of non-uniform patterns. In vehicle detection, we observed the same response as in face recognition. On the other hand, multi-scale approaches are not affected by changing the length of radius. Fig 9. shows the performance of traditional LBP and multi-scale approaches using different radiuses.

4.2 Accuracy, speed and size

PHis-LBP was compared against two single-scale methods (LBP and HOG) and multi-scale methods (pyramids of LBP, gradient magnitude and HOG). In single-scale methods, an unique descriptor is generated from applying LBP or HOG algorithms in each dataset image. Also, we evaluated PHis-LBP against its real version, i.e., P-LBP computes each histogram directly from downsampled images. Finally, the proposed method in [19] was used to generate other feature pyramids using gradient magnitude and HOG as extracted channels. The first comparison, regarding to measure performance in object detection, where we present an average of PWC from all tested algorithms varying different parameters as neighbor number and radius length. Object detection performance is shown in Fig. 10.
Traditional algorithms as LBP and HOG present a lower performance than PHis-LBP, caused by its non multi-resolution representation. As we expected, PHis-LBP performance is not degraded with respect to fully computed version (P-LBP) because every single histogram is estimated accordingly to its nearest scale. Feature pyramids of gradient magnitude and HOG were also tested, presenting slightly better performance but increasing the number and complexity of operations required to compute features.
Concerning processing speed, results are shown in Fig. 11, all tested algorithms were applied to images of size 40 x 100. LBP has the highest frame rate because it only performs simple operations like thresholding and bitwise rotations. HOG is the most complex algorithm, since in each window belonging to the image, histogram voting takes place. Single-scale HOG reaches to process around 60 frames per second. Speed performance of PHis-LBP outperforms about 20% relative to P-LBP, we save operations in PHis-LBP but it is still necessary to compute intermediate scales as reference. Other feature pyramids like HOG and gradient magnitude are 50% slower than our proposed method.
The next metric is about the size occupied for storing computed features, normally an LBP feature has a size of $P + 2$ elements, where the most common values for P are 8, 16 and 24. Consequently, PHis-LBP is sized according to the number of scales used to form a pyramid. Feature sizes of other algorithms are shown in table 1. It should be noted that PHis-LBP is at least 10 times smaller than other single and multi-scale algorithms.

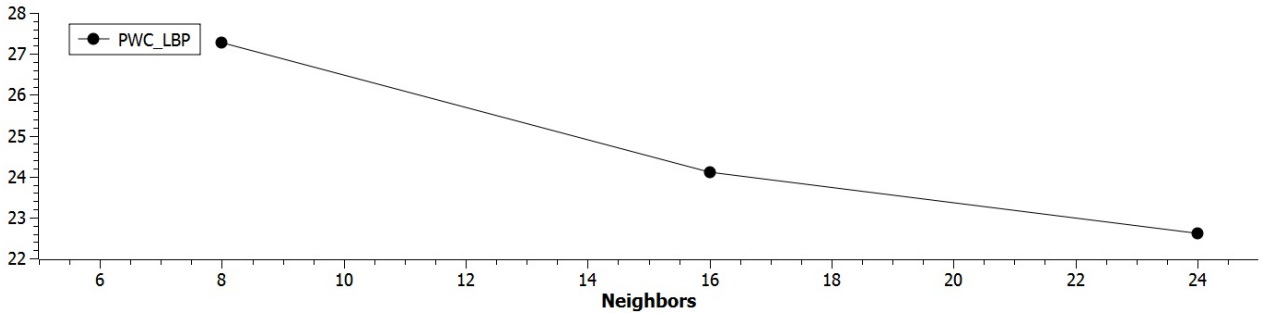

Figure 7: PWC of traditional LBP varying size of neighborhood set.

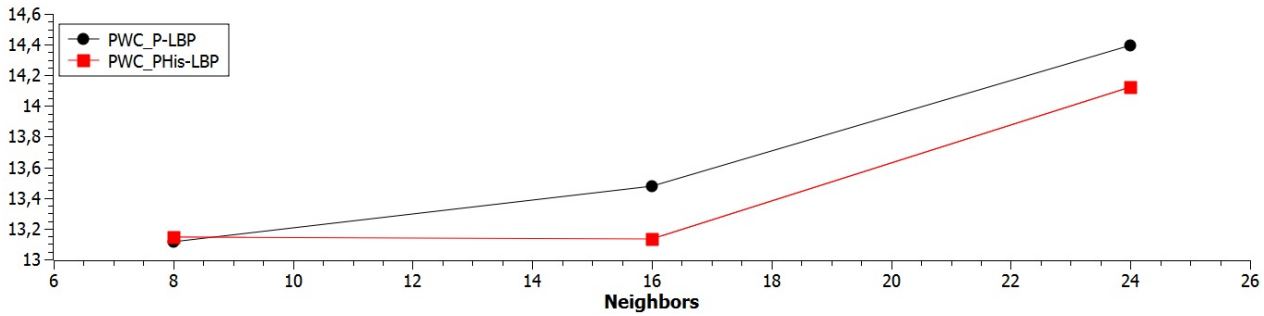

Figure 8: Performance of multi-scale approaches varying size of neighborhood set.

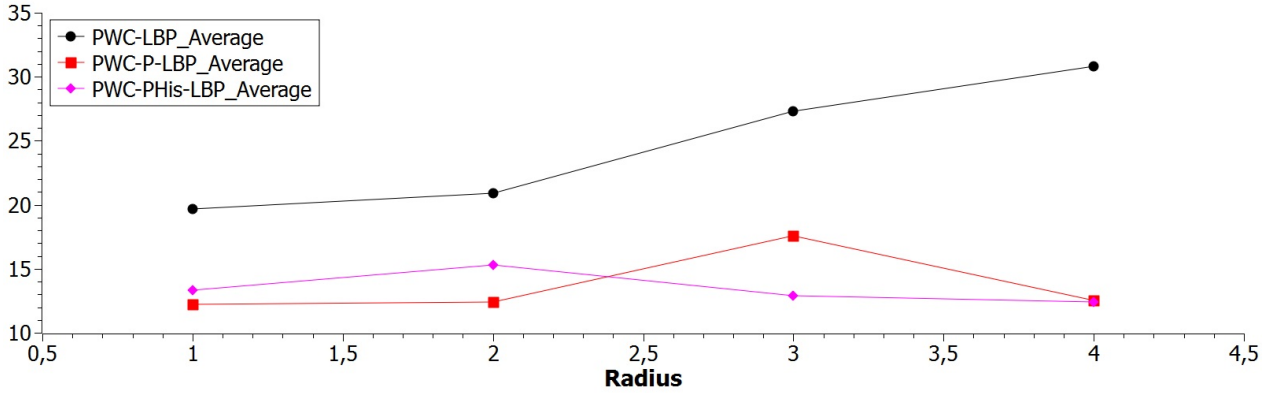

Figure 9: Performance of traditional and multi-scale LBP in vehicle detection with different radiuses.

5 Conclusions

In this paper a multi-scale method for vehicle detection that avoids high computational cost using approximations in a certain number of scales has been proposed. The method is based in two main parts, object detection is performed using local binary patterns which deals perfectly with two major drawbacks in object detection, illumination and rotational changes. On the other hand, we constructed a histogram pyramid of LBP but a given number of histograms are computed directly and others approximated in order to reduce processing time.

We compared several methods against our proposed method and it performed a similar detection rate than other robust detection algorithms. PHis-LBP decreases percentage of wrong classifications by half regarding to single-scale version of LBP. Proposed method is 20% and two times faster than its fully computed version and against other feature pyramids, respectively. Another important aspect of our method is that its feature pyramids have a good trade-off between accuracy and memory storage. In some cases, a PHis-LBP feature is 10 to 800 times smaller than other object detection features, this aspect becomes more important when we have limited resources.

We observe that PHis-LBP is a robust and lightweight

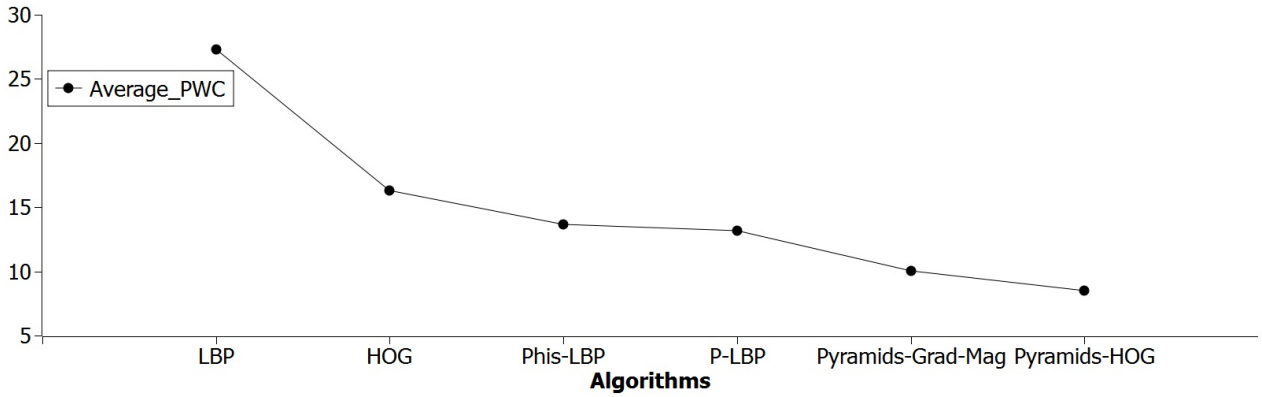

Figure 10: PWC comparison between PHis-LBP and other object detection algorithms

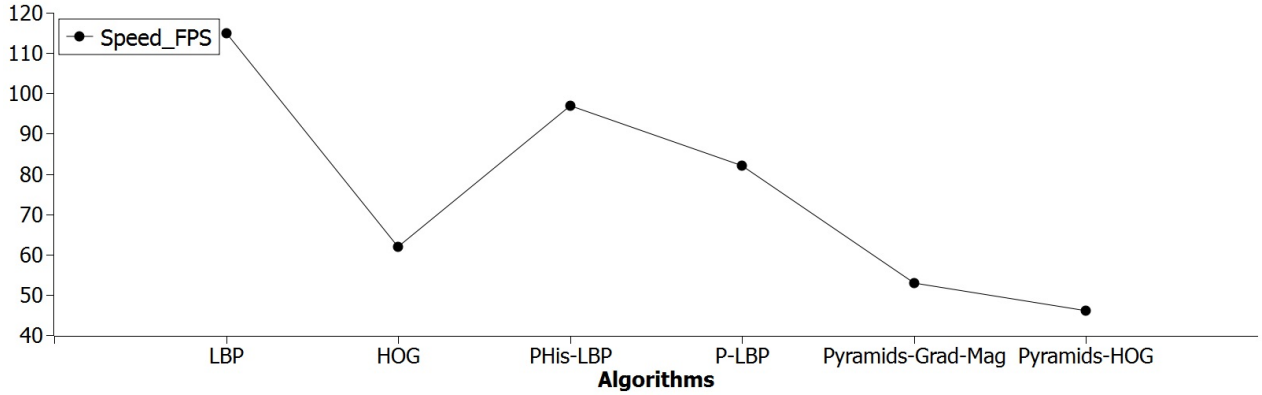

Figure 11: Speed performance of proposed method and other algorithms in vehicle detection.

Table 1: Number of elements per feature in different vehicle detection algorithms.

Algorithm	size
Pyramids-HOG	143382
Pyramids-Grad-Mag	23897
HOG	1584
PHis-LBP-17	170
PHis-LBP-13	130
PHis-LBP-9	90
PHis-LBP-5	50
PHis-LBP-3	30
LBP	10

method, ideally suitable for circumstances where a real-time processing solution that is constrained by hardware resources like processor speed, memory or energy is needed. An example of this are unmanned aerial vehicles, surveillance tasks require real-time processing to achieve its goals. However, these kinds of vehicles have limited resources including energy and hardware, it is there that our method fits well.
In our ongoing work we are particularly interested to employing parallelism and FPGA implementation to speed up computation 10-15 times faster than its current speed, which will allow us to process bigger images in a real-time environment.

References:

[1] S. Sivaraman and M. M. Trivedi, “Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis,” IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1773–1795, 2013.

[2] W. C. Chang and C. W. Cho, “Online boosting for vehicle detection,” IEEE Trans. Syst. Man, Cybern. Part B Cybern., vol. 40, no. 3, pp. 892–902, 2010.

[3] B. F. Lin et al., “Integrating appearance and edge features for sedan vehicle detection in the blind-spot area,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 2, pp. 737–747, 2012.

[4] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: Features and classifiers," IEEE Trans. Image Process., vol. 15, no. 7, pp. 2019–2034, 2006.

[5] H. Tehrani Niknejad, A. Takeuchi, S. Mita, and D. McAllester, "On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation," IEEE Trans. Intell. Transp. Syst., vol. 13, no. 2, pp. 748–758, 2012.

[6] Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Learning a family of detectors via multiplicative kernels," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 3, pp. 514–530, 2011.

[7] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," IEEE Trans. Intell. Transp. Syst., vol. 11, no. 2, pp. 267–276, 2010.

[8] Y. Dong and J. Ma, "Wavelet-based image texture classification using local energy histograms," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 247–250, 2011.

[9] A. Jain and M. Tuceryan, "HandBook of Pattern Recognition and Computer Vision," World Sci. Publising Co., pp. 207–248, 1998.

[10] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Li, "Learning Multi-scale Block Local Binary Patterns for Face Recognition," in Advances in Biometrics, 2007, pp. 828–837.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, Nov. 2004.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," Comput. Vis. Image Underst., vol. 110, no. 3, pp. 346–359, Jun. 2008.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 2005, vol. I, pp. 886–893.

[14] Y. Freund and R. E. Schapire, "A Decision-theoretic Generalization of On-line Learning and an Application to Boosting," J. Comput. Syst. Sci., vol. 55, pp. 119–139, 1997.

[15] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. May 2002, pp. 1475–1490, 2004.

[16] L. Nanni, A. Lumini, and S. Brahnam, "Survey on LBP based texture descriptors for image classification," Expert Syst. Appl., vol. 39, no. 3, pp. 3634–3641, 2012.

[17] S. A. Orjuela Vargas, J. P. Yañez Puentes, and W. Philips, "Local Binary Patterns: New Variants and New Applications," vol. 506, S. Brahnam, L. C. Jain, L. Nanni, and A. Lumini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 207–248.

[18] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral Channel Features," BMVC 2009 London Engl., pp. 1–11, 2009.

[19] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[20] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987, Jul. 2002.