Real-Time Object Segmentation Using a Bag of Features Approach

David ALDAVERT [a,1], Arnau RAMISA [c,b], Ramon LOPEZ DE MANTARAS [b] and Ricardo TOLEDO [a]

[a] *Computer Vision Center, Dept. Ciencies de la Computació, Universitat Autonòma de Barcelona, Catalunya, Spain*

[b] *Institut d'Investigació d'Inteligencia Artificial (IIIA-CSIC), Campus UAB, Catalunya, Spain*

[c] *INRIA-Grenoble, LEAR Team, France*

Abstract. In this paper, we propose an object segmentation framework, based on the popular bag of features (BoF), which can process several images per second while achieving a good segmentation accuracy assigning an object category to every pixel of the image. We propose an efficient color descriptor to complement the information obtained by a typical gradient-based local descriptor. Results show that color proves to be a useful cue to increase the segmentation accuracy, specially in large homogeneous regions. Then, we extend the Hierarchical K-Means codebook using the recently proposed Vector of Locally Aggregated Descriptors method. Finally, we show that the BoF method can be easily parallelized since it is applied locally, thus the time necessary to process an image is further reduced. The performance of the proposed method is evaluated in the standard PASCAL 2007 Segmentation Challenge object segmentation dataset.

Keywords. Object Segmentation, Bag Of Features, Feature Quantization, Densely sampled descriptors

Introduction

In this paper, we address the problem of segmenting objects in the image in real-time using a bag of features (BoF) approach. Currently, some of the most successful object segmentation method use BoF to model the probability that a pixel of an image belongs to a certain category. In spite of its simplicity, the BoF model has shown a performance comparable or better than more complex approaches [9,8,6] in many state of the art object segmentation datasets [4,13].

A straightforward way to obtain a *semantic* segmentation of an image using a BoF model consists in accumulating the visual words within a local region defined in the neighborhood of an image pixel and classifying the resulting histogram. Since this procedure has to be repeated for each pixel of the image, this method has a computational complexity not suitable for real-time applications. However, time constrains are crucial

[1]Corresponding Author: David Aldavert, Computer Vision Center, Dept. Cienciès de la Computació, Universitat Autonòma de Barcelona, Barcelona, Spain, E-mail: aldavert@cvc.uab.cat.

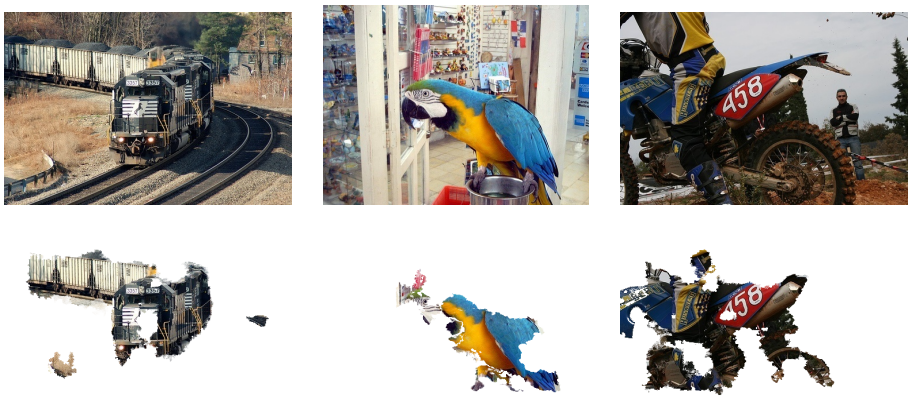


Figure 1. Examples of the segmentation results obtained using the proposed method for the object categories **TRAIN**, **BIRD** and **MOTORBIKE**.

in fields like advanced driver assistance systems or robotics, where the system has typically only few milliseconds to processes each image. Therefore, several authors have focused on reducing the computational cost of the BoF schema while maintaining a good segmentation accuracy. Shotton et al. [15] proposes the use of Semantic Texton Forest to efficiently segment objects in the image. A Semantic Texton Forest is an ensemble of decision trees, which obtain a texton category for a pixel directly from the pixels in its neighborhood. Therefore, the step of computing and accumulating image gradients, which is necessary by many state of the art local descriptors, can be avoided. The final pixel-level object classification is obtained applying locally the BoF method with a sliding windows approach, and using an integral image to accumulate the textons in the BoF histogram. This efficient method is able to process up to 8 images of 300×200 pixels per second. Fulkerson et al. propose a similar approach in [5], but using densely SIFT descriptors and Hierarchical K-Means to sample the visual words from image instead of texton forests. To make their schema as fast as possible, Fulkerson et al. propose the use of integral images to calculate the SIFT descriptor efficiently, and the Agglomerative Information Bottleneck to effectively reduce the dimensionality of the codebooks. However, the two previous methods require the use of an integral image for each visual word of the codebook, hence limiting its size in practice. Since the amount of visual words of the codebook is critical in order to obtain a good segmentation accuracy [11], Aldavert et al. propose the Integral Linear Classifier (ILC) to overcome this problem [1,2]. The ILC uses a single integral image to calculate the score of a linear classifier directly, avoiding the visual words histogram accumulation step.

In this paper, we evaluate the method of [1] with a new color descriptor to complement the gradient-based local descriptor used in the original work. This color descriptor is designed so that it can be efficiently calculated using integral images. Results show that color is a useful cue to increase the segmentation accuracy, specially in large homogeneous regions. Then, in order to improve the descriptive power of the image representation used, the well known Hierarchical K-Means (HKM) codebook is extended using the recently proposed Vector of Locally Aggregated Descriptors (VLAD) [7]. Finally, we show that since the BoF method is applied locally, it can be easily parallelized, greatly reducing the time necessary to process an image. The proposed method is evaluated in

the well known PASCAL 2007 Segmentation Challenge object segmentation dataset (in Figure 1 a segmentation example of three images of this data is shown).

The paper is organized as follows. In section 1 the efficient local descriptors used to obtain the image information are reviewed. Then, in section 2 the HKM codebook and the VLAD image descriptor are explained and in section 3 the efficient Integral Linear Classifier is presented. Finally, in section 4 the obtained experimental results are shown and in section 5 the final conclusions are presented.

1. Efficient Densely Sampled Features

Recently, some works have highlighted the advantages of densely sampled rotation variant descriptors with respect to keypoint-based approaches [12]. For example, more information can be extracted from image, or they allow to bypass the time-consuming keypoint detection step. Furthermore, since descriptors are rotation variant, they can be very efficiently calculated using integral images. Although rotation invariance is a good descriptor property for wide baseline feature matching, studies like [16] have shown its negative effect in BoF approaches. In this work, we have used two descriptors: the SURF descriptor [3], which is based on gradient information from the image, and the Integral Color Descriptor, our novel proposal to summarize the chromatic distribution information in a local region of the image efficiently.

1.1. Speeded Up Robust Features

As our aim is to obtain the image object segmentation in real-time, the SURF descriptor [3] becomes a natural choice, as its performance is similar to more complex descriptors like the IHOG [17], but with a lower computational cost. The SURF descriptor is a histogram of Haar wavelet responses accumulated at different spatial bins of the local region. Let d_x and d_y be the Haar wavelet response in the horizontal and vertical directions respectively. Then, a SURF descriptor with $P \times Q$ spatial bins is calculated by integrating the wavelets responses d_x and d_y over each bin sub-region. To take into account the polarity of the intensity images, the sum of absolute values of the responses, namely $\|d_x\|$ and $\|d_y\|$, is also extracted (see a SURF descriptor example in Figure 2). Hence, a

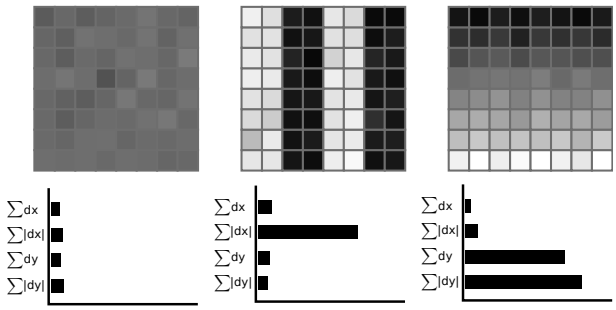


Figure 2. Example of SURF descriptors obtained from different image patches (from left to right): in a homogeneous region, in the presence of frequencies in the x direction, and when the intensity is gradually increasing along the y axis.

four-dimensional descriptor vector of the form $(\sum d_x, \sum \|d_x\|, \sum d_y, \sum \|d_y\|)$ is generated for each spatial bin of the descriptor. Finally, the resulting $P \times Q \times 4$ dimensional descriptor is normalized using L1-norm.

In order to make the SURF descriptor computable through integral images, we have dropped the Gaussian weighting mask with the origin at the feature center. Thus, in our implementation, only $(P+1) \times (Q+1) \times 4$ memory accesses and $P \times Q \times 4$ additions are necessary to compute a descriptor, regardless of the feature region size.

1.2. Integral Color descriptor

Although it is usually discarded, color is a source of information that can improve a bag of features object model by increasing the distinctiveness of certain object categories (e.g. grass is usually green, or fire extinguishers are red), or by incorporating information in homogeneous regions where gradient based descriptors are not reliable. To incorporate color information to our model, we propose the Integral Color (IC) descriptor, which can be calculated very efficiently in a way similar to that of the IHOG:

First, N *Hue* images, one for each *Hue* bin of the descriptor, are created interpolating the *Hue* value weighted by its *Saturation* for each pixel of the image. Then, creating an integral image for each *Hue* bin from these images, the value of a *Hue* bin can be calculated using only 4 additions and main memory accesses. Therefore, to calculate a IC descriptor with $P \times Q$ spatial bins, only $N \times (P+1) \times (Q+1)$ memory accesses and $N \times P \times Q$ additions are needed, regardless of the feature size. Finally, the obtained IC descriptor is normalized using a L1-norm. Figure 3 shows an example of the orientation and integral images used to calculate the IC descriptor.

2. Codebook

Once local descriptors are calculated from the image, they have to be quantized into visual words using a codebook. Since the computational cost of quantizing a D-dimensional descriptor with a linear codebook of V visual words is $O(DV)$, a sub-linear quantization complexity is desirable for large codebook. Therefore, a hierarchical code-

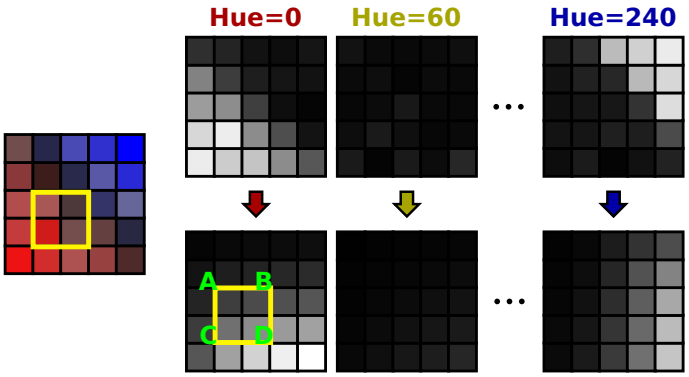


Figure 3. Example of IC descriptor computation: First a *Hue* image is generated for each *Hue* bin of the descriptor (top row). Then, an integral image corresponding to each color and orientation bin is created (bottom row), which allows to compute the bin value for any arbitrary region with only four additions.

book becomes a natural choice [11,10]. In this work, we use the Hierarchical K-Means (HKM) [11] codebook and extend it by using the VLAD method to reduce the effects of quantization errors.

2.1. Hierarchical K-Means

The HKM [11] defines a hierarchical quantization of the feature space by recursively applying the K-Means algorithm. Here, the k value determines the branch factor (i.e. the number of descendants of each node) of a tree. Then, an HKM is generated as follows: First, the k-means algorithm is used to split the training data into k clusters. Then, this clustering process is recursively applied to the cluster from the previous level until a certain amount of data samples or a maximum depth is reached. This recursive method creates a codebook with a reduced computational cost both in the training and descriptor quantization phases. The computational complexity of quantizing a D dimensional descriptor using a HKM with V visual words is $O(Dk \log_k V)$. In the original implementation of the HKM, all nodes of the tree are used as visual words to alleviate missclassification problems in the superior levels of the tree, and the contribution of each node of the histogram is weighted by $w_i = \ln(\frac{N}{n_i})$, where n_i is the number of BoF histograms that contains the visual word i, and N is the total number of BoF histograms. However, the use of these two refinements has a modest impact in the performance of the HKM. Consequently, they are removed from our implementation.

2.2. Vector of Locally Aggregated Descriptors

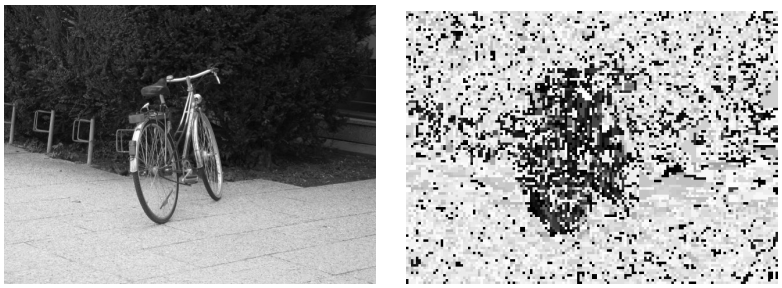
Instead of directly using the HKM codebook to quantize descriptors into visual words, we use the Vector of Locally Aggregated Descriptors recently proposed in [7]. The HKM codebook is used to search visual word corresponding to a given descriptor efficiently. Then, instead of directly accumulating the visual words given by HKM, the VLAD descriptor accumulates the differences between the given descriptor and the descriptor representing the center of the selected visual word.

The addition of the VLAD descriptor to the HKM codebook increases the information which contains the codebook, since the VLAD descriptor characterizes the distribution of the vectors with respect of the visual word center instead of just accumulating the different amount of visual words. Therefore, the size of the codebook is increased D times, where D is the dimensionality of the used local descriptor.

3. Pixel Categorization

A pixel is categorized by accumulating the visual words within a local rectangular region defined around the pixel into a histogram and using a linear classifier to obtain its category label.

Although the computational cost of accumulating the visual words for each pixel of the image would be extremely high, since a linear classifier is used, the classification score can be obtained by creating an image of the classifier weights associated to each visual word. Then, using integral images, it is possible to obtain the classification score for an arbitrary image sub-region in constant time [1,2]. In short, the score of a linear classifier can be obtained applying:



a) b)

Figure 4. Example of the image with the linear classifier weights for each visual words b) obtained from image a).

$$score = \frac{1}{\|N\|} \sum_{i=0}^{n} f_i w_i + b \tag{1}$$

where N is the L1 norm of the feature vector, f_i is the i-th component of the feature vector and w_i is the i-th component of the linear classifier weight vector quantized as an integer. Then, using integral images to accumulate the scores of the visual words, and subtracting W_{min} from the weight vector components, the sum of the previous equation for a rectangular image region R can be calculated as:

$$H_R = I_c(x_u, y_u) + I_c(x_b, y_b) - I_c(x_u, y_b) - I_c(x_b, y_u) \tag{2}$$

where (x_u, y_u) and (x_b, y_b) are respectively the upper left and bottom right corner coordinates of region R, and I_c is the integral image. Then, the output score of a linear classifier applied to any rectangular image region can be calculated as follows:

$$score = \frac{1}{N} H_R + W_{min} + b \tag{3}$$

which can be obtained in constant time independently of the region size. In Figure 4 an example of the I_c image generated from the **BIKE** classifier is shown.

4. Results

The performance of the proposed method is evaluated using the PASCAL 2007 Segmentation Challenge dataset (VOC2007) [4]. The VOC2007 dataset contains 21 categories with few training examples and extreme variation in deformation, scale, illumination, pose, and occlusion. To train our method, we use both training and validation images of the segmentation dataset. The performance measure for the dataset is the average pixel accuracy: For each category, the number of correctly classified pixels divided by the ground truth labeled pixels.

4.1. Experiment setup

Both types of local descriptors are calculated from a square region of 40 pixels and densely extracted every 4 pixels, with 2×2 spatial bins and the IC descriptors have 8 hue

Table 1. Accuracy results obtained in the VOC2007 dataset. The SURF label means that only the SURF descriptor is used while BOTH means that both descriptors are used together. The HKM stands for the HKM codebook used alone while VLAD means that HKM is used together with the VLAD descriptor.

Category	SURF+HKM	BOTH+HKM	SURF+VLAD	BOTH+VLAD
BACKGROUND	75.6%	76.0%	76.9%	78.0%
AEROPLANE	8,6%	9.8%	2.7%	3.9%
BICYCLE	0,4%	1.5%	1.5%	2.0%
BIRD	5.3%	5.7%	16.3%	16.5%
BOAT	1.8%	3.7%	4.3%	5.7%
BOTTLE	0.1%	1.1%	0.2%	1.1%
BUS	5.9%	7.8%	15.7%	16.1%
CAR	2.1%	2.0%	2.9%	4.0%
CAT	10.2%	12.1%	17.9%	18.3%
CHAIR	1.4%	1.7%	2.8%	3.1%
COW	12.7%	14.3%	48.6%	50.0%
DINNIGTABLE	20.7%	22.1%	14.3%	16.3%
DOG	30.9%	32.6%	37.6%	39.2%
HORSE	8.6%	10.3%	40.6%	41.7%
MOTORBIKE	21.4%	22.5%	60.3%	61.9%
PERSON	69.2%	70.1%	81.1%	81.7%
POTTEDPLANT	9.2%	8.0%	24.1%	22.3%
SHEEP	15.5%	16.4%	17.0%	17.9%
SOFA	1.2%	1.9%	2.3%	3.8%
TRAIN	8.2%	8.3%	11.4%	12.0%
TVMONITOR	0.4%	1.0%	2.5%	3.6%
AVERAGE	14.7%	15.7%	22.9%	23.8%

bins, resulting in a 16-dimensional SURF descriptor and a 32 dimensional IC descriptor. Two different codebooks are used to quantize the SURF and IC descriptors. The final histogram is obtained by concatenating both descriptors histograms. Finally, pixels are categorized by accumulating the visual words that are within a squared region of 40 pixels around the pixel. The computation times mentioned in the results have been obtained using a desktop computer with an Intel 2.6Ghz 960 i7 CPU with 12Gb of RAM.

4.2. Segmentation results

We have evaluated the segmentation performance of the proposed method both using the SURF descriptor alone and together with the IC descriptor, and with the HKM and the VLAD image descriptors. Table 1 displays the segmentation accuracy obtained for each category using the different configurations. The obtained results show that the use of a color descriptor slightly increases the segmentation accuracy: in average, it increases by 1 percent the performance of the proposed method. However, the use of the VLAD descriptor has a greater impact in the performance of the segmentation method, increasing in average the performance by a 51%. Finally, the obtained accuracy results using both descriptors and the VLAD method are similar to the results obtained by other methods of the state of the art [15,14,5].

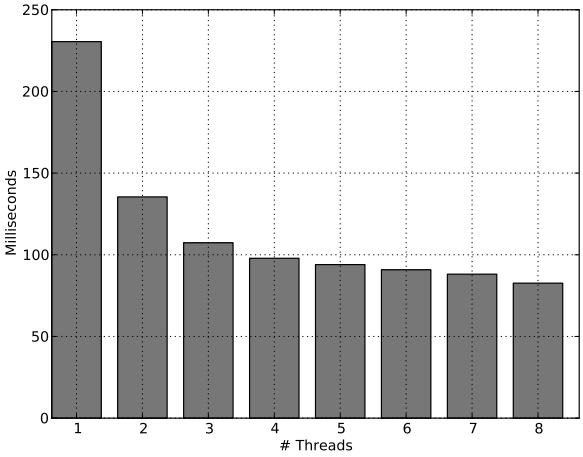


Figure 5. Average time required to process an image using different number of threads. Note that the processor actually only has 4 cores with Hyper-Threading.

4.3. Computational cost evaluation

Finally, we evaluate the computational cost of the proposed method. All the time results shown in this section correspond to the 21-class object segmentation obtained by the best classifier of the previous section: using both local descriptors with the HKM codebook together with the VLAD descriptor. Since both visual words and pixel categorization are locally calculated, the image can be divided in different regions and processed separately. Therefore, we have used a multi-threading approach to process different image parts in parallel. In Figure 5, the time needed to process an image using different number of threads is shown. The processor used in the experiment has 4 cores with Hyper-Threading, so that, the reduction in the computational cost is lower when more than 4 cores are used.

The obtained results show that the proposed method is suitable to be parallelized to significantly reduce its computational cost. Using a single thread, the average cost of processing an image is about 230.49 ms, while using 8 threads the computational cost reduces to 82.63 ms (the computational cost is reduced a 178%). Therefore, using the multi-threading approach, the proposed method can process up to 12 images per second.

5. Conclusions

In this paper, we present a real-time object segmentation method which is able to process up to 12 images per second. Our main contributions are the introduction of an efficient color descriptor, that can be calculated using integral images, and the incorporation of the recently proposed VLAD image descriptor to the HKM codebook. Experimental results show that the color descriptor has a modest impact in the performance of the segmentation method, but the VLAD descriptor notably increases the accuracy of the obtained segmentation. Finally, we show that using a multi-threading approach the computational

cost of the proposed method can be easily reduced, obtaining an average processing time of 82 milliseconds per frame in a consumer level four cores computer.

Acknowledgements

This work was supported by the Spanish Ministry of Education projects TIN 2006-15308-C02-02, MIPRCV Consolider Ingenio 2010 and the grant 2009-SGR-1434 of the Generalitat de Catalunya.

References

[1] D. Aldavert, A. Ramisa, R. Toledo and R. Lopez de Mantaras, *Efficient Object Pixel-Level Categorization Using Bag of Features*, Proceedings of the 5th International Symposium on Visual Computing, 44–54, 2009.

[2] D. Aldavert, A. Ramisa, R. Toledo and R. Lopez de Mantaras, *Fast and Robust Object Segmentation with the Integral Linear Classifier*, Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, 2010.

[3] H. Bay, A. Ess, T. Tuytelaars and L. van Gool, *SURF: Speeded Up Robust Features*, Journal of Computer Vision and Image Understanding (CVIU), 110, 3, 346–359, 2008.

[4] M. Everingham, L. van Gool, C.K.I. Williams, J. Winn and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[5] B. Fulkerson, A. Vedaldi and S. Soatto, *Localizing Objects with Smart Dictionaries*, Proceedings of European Conference on Computer Vision, 179–192, 2008.

[6] B. Fulkerson, A. Vedaldi and S. Soatto, *Class Segmentation and Object Localization with Superpixel Neighborhoods*, Proceedings of the International Conference on Computer Vision, 670–677, 2009.

[7] H. Jégou, M. Douze, C. Schmid and P. Pérez, *Aggregating local descriptors into a compact image representation*, Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, 2010.

[8] D. Larlus, J. Verbeek and F. Jurie, *Category level object segmentation by combining bag-of-words models with Dirichlet processes and random fields*, International Journal of Computer Vision, 2009.

[9] M. Marszalek and C. Schmid, *Accurate Object Localization with Shape Masks*, Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, 1–8, 2007.

[10] F. Moosmann, E. Nowak and F. Jurie, *Randomized clustering forests for image classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 30, 9, 1632–1646, 2008.

[11] D. Nister and H. Stewenius, *Scalable recognition with a vocabulary tree*, Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, 2161–2168, 2006.

[12] E. Nowak, F. Jurie and B. Triggs, *Sampling Strategies for Bag-of-Features Image Classification*, Proceedings of European Conference on Computer Vision, 490–503, 2006.

[13] A. Opelt, A. Pinz, M. Fussenegger and P. Auer, *Generic object recognition with boosting*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28, 3, 416–431, 2006.

[14] C. Pantofaru, C. Schmid and H. Martial, *Object Recognition by Integrating Multiple Image Segmentations*, Proceedings of European Conference on Computer Vision, 1, 2008.

[15] J. Shotton, M. Johnson and R. Cipolla, *Semantic texton forests for image categorization and segmentation*, Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, 1–8, 2008.

[16] J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid, *Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study*, International Journal of Computer Vision, 73, 2, 213–238, 2007.

[17] Q. Zhu, M.C. Yeh, K.T. Cheng and S. Avidan, *Fast Human Detection Using a Cascade of Histograms of Oriented Gradients*, Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, 1491–1498, 2006.