Scan Window Based Pedestrian Recognition Methods Improvement by Search Space and Scale Reduction

Raluca Brehar and Sergiu Nedevschi

Abstract—Most of computation time when dealing with a pedestrian detector is spent in the feature computation and then in the multi-scale classification. This second step consists of applying scanning windows at multiple scales. Depending on the number of scales and on the image dimension, this step is slow because a large number of windows is generated. An efficient pruning algorithm able to remove most of the scan windows brings a significant contribution on the overall execution time. We propose a scan window pruning algorithm based on a combination of several filters: (1) remove windows based on the relation between the dimension and the position in the scene; (2) remove uniform regions from the image such as the sky or the road; (3) remove regions with high density of horizontal edges such as vegetation parts; (4) keep local maxima windows having a high density of connected vertical edges. The combination of these four filters eliminates more than 90% of the scanning windows in a given image and maintains the windows that are overlapped on regions with a high probability of representing a pedestrian.

We have integrated our method in a generic framework for pedestrian detection [1] and we have studied two aspects: (1) how the performance of the algorithms varies with respect to the filters proposed by our pruning strategy and (2) what is the speed gain when the quality loss is negligible. The proposed filters have a negligible loss in performance and even improve it in some cases while the execution time is improved.

I. INTRODUCTION

Pedestrian detection for automotive applications is an open field for scientific and industrial exploration. Most approaches combine information cues from intensity images, infrared data, motion and depth information of the traffic scenes striving to obtain high detection accuracy at low execution costs.

One of the most explored information cue is the intensity image. The main issue when dealing with intensity images is the slow speed and sustained effort has been made by numerous research teams to improve this aspect by:

- Efficient feature computation like: integral images, integral channel features.
- Feature scaling or approximation of features at multiple scales instead of image scaling.
- Appropriate classifiers trained for different sizes of the detection window.
- Prior knowledge about the scene: provide a region of interest where pedestrians might appear; a pedestrian is not likely to appear in the upper part of the traffic scene where we usually find the sky.
- Speed-up improvements by parallel implementations (GPU, MPI).

Most of the above solutions use in a way or another a scanning window when performing the detection. There are several popular approaches:

1) The classical approach that uses multiple image scales and a window of constant dimension.
3) Multiple-scale pedestrian detection models and one image scale [4].
4) Crosstalk cascades that use a correlation between detector responses at nearby locations and scales [5].

Still with all these approaches the overall execution time can be high and a GPU implementation brings the necessary speed-up improvement.

Our work brings a contribution in the scan window removal procedure. The algorithm we propose identifies the windows of interest with a high accuracy and with a low execution time on a CPU implementation. We have performed extensive experiments on four popular pedestrian datasets in order to extract laws that model the profile for the windows in which the pedestrians might be. The contributions reside in the proposal of a set of decision rules (that we call filters) able to reduce the number of scanning windows with 60% up to 90% and keep a high score in the accuracy of the correct pedestrian localization.

Our setup is particularly suitable for images of traffic scenes because we exploit the fact that closer pedestrians have a large height opposed to far pedestrians that are smaller. Using this automotive setup our purpose is to determine in each region of the image the dimension of the scanning windows that must be retained. Hence for each image region we deduce the suitable scales for the scan windows. The reduction factor of the algorithm is further refined and decreased by applying a series of filters that successively eliminate part of the scan windows. The filters use information about vertical and horizontal edges of the image. Based on this information we remove uniform areas such as the sky or parts of the road or buildings, some of the vegetation, and we enhance the windows having a high density of connected vertical edges. The algorithm was evaluated on four popular datasets: Daimler [6], ETH [7], TUD-Brusseless[8], Caltech [9].
II. RELATED WORK

Significant studies have been made for the general task of pedestrian detection in the context of advanced driver assistance systems [10], [11], [12] and in particular for the context of monocular pedestrian detection [13]. All those study identify the importance of region of interest generation and provide a summary of some existing methods.

In what follows we will refer only to similar approaches that deal with the reduction of the scanning window in monocular images.

A. Scanning window reduction methods

In the case of scanning window reduction methods we start with the pioneers Viola & Jones where an image is scaled several times and then it is traversed by a scanning window of constant dimension. But re-scaling the image and computing features for each image scale is computationally expensive.

Another approach is based on the idea of scaling the features instead of scaling the image. Even if popular features such as HOG are not scale invariant the feature responses can be computed at a single scaled and then approximated by interpolation across multiple scales. Dollar [2] uses this method and scales the image only a few times (sparsely sampled image pyramid). Dollar [3] also proposes finely sampled feature pyramids that are computed very fast while keeping the detection performance high. The study a broad family of features computed at octave-space scaled intervals and prove that these are sufficient to approximate features on a finely-sampled pyramid.

Multi-scale pedestrian detection models are learned in the approach of Benenson [4]. They use one image scale and instead of approximating the features, the thresholds of the detection models are approximated across the different scales.

An evaluation of adjacent windows and a correlation between detector responses at nearby locations and window scales is proposed by Dollar [5]. They introduce the cross-talk cascades in which scanning windows are evaluated in two steps: excitation for which a sparse set of detector responses is computed and then sampled more densely around promising locations and inhibition step during which window candidates can be pruned if their scores are sufficiently smaller than the neighbors.

The idea of inferring ground truth constraints to the scanning window position and heights was presented by [14]. They investigate how geometric constraints can be used for efficient sliding-window object detection. Starting with a general characterization of the space of sliding-window locations that correspond to geometrically valid object detections, they derive a general algorithm for incorporating ground plane constraints directly into the detector computation. They demonstrate the potential of the method in a fast CUDA implementation of the HOG detector.

Exploiting the constraints of a typical automotive setup where far pedestrians are smaller and close pedestrians are higher we perform an analysis of the pedestrian bounding boxes profile. This analysis has as main objective the determination of regions of interest for pedestrian localization.

The regions of interest (ROI) are modeled by scanning windows having a high probability of containing a pedestrian. We propose a positioning filter that defines, for each region of interest, the applicability scales and sizes of the scanning windows inside that area. The results of this filter are refined by a couple of edge based filters that combine horizontal edge density, energy of vertical edges, density of vertical edge contours for reducing the number of scanning windows inside each region.

A. Spatial localization filter

In a classical approach a scanning window traverses the image from the top-left corner with a certain stride on both axes (4 pixels for example) and then the window is scaled several times and the scanning is done in the same fashion. This approach generates a huge set of windows.

Using the configuration of a typical automotive application and the physical knowledge about possible pedestrian heights we can deduce a relation between the position of a pedestrian and its dimension in the image, hence we can infer information about the spatial localization of the pedestrian. Figure 1 depicts a pinhole camera setup. In the scene depicted by figure 1 a pedestrian is standing very close to the camera. As one can notice its height in the image plane is large. We suppose that the possible pedestrian heights are in the range $[H_{\text{min}} \ldots H_{\text{max}}]$ meters. (An example could be the range $[1\text{–}2.2]$ meters that covers children and adults). Given this range for each distance from the camera we can compute the range of projected heights in the image plane as follows.

Consider the world coordinate system and a pedestrian touching the ground at point $P_1(X_w, Y_1, Z_w)$. This point is projected on the image plane and its correspondent is $p_1(u, v)$ in the image coordinate system. If we know that the range of heights is the world coordinate system is $[H_{\text{min}} \ldots H_{\text{max}}]$ meters we can find out the range of window heights having as top left corner the point $p_1$ in the image plane. Consider $P_2(X_w, Y_2, Z_w)$ the point at distance $H$ above $P_1$, where $H = |Y_2 - Y_1|$ is the current height.
of the pedestrian and it is in the range $[H_{\text{min}} \ldots H_{\text{max}}]$. Let $p_2(u, v)$ be the projection of $P_2$ on the image plane. We want to find the possible ranges of $|v2 - v1|$.

Using the mathematical model of the pinhole camera, we know that the projections of points $P_1$ and $P_2$ are given by:

$$s \begin{bmatrix} u \\ v1 \\ 1 \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}; \quad s \begin{bmatrix} u \\ v2 \\ 1 \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$  \quad (1)

where the matrix $P$ is the product between the internal camera matrix, $A$ and the concatenation of the world to camera rotation matrix, $R_{WC}$ and the translation from world to camera matrix $T_{WC}$. The factor $s$ is a scaling factor of the camera system.

$$P = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$ \quad (2)

where $f_x$ and $f_y$ are the focal length on horizontal and vertical axes and $r_{ij}$ represent the rotation coefficients.

Generally we can express $P$ as:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}$$ \quad (3)

Using equation 1 we can express $v1$ and $v2$ as:

$$s \cdot v1 = p_{21}X_w + p_{22}Y_w + p_{23}Z_w + p_{24}$$ \quad (4)

$$s \cdot v2 = p_{21}X_w + p_{22}Y_w + p_{23}Z_w + p_{24}$$ \quad (5)

Hence the height of a pedestrian at coordinate $p_1$ in the image plane is given by:

$$|v2 - v1| = \frac{P_{22}}{s} |Y2_w - Y1_w| \leq \frac{P_{22}}{s} [H_{\text{min}} \ldots H_{\text{max}}]$$ \quad (6)

Consider the bottom right coordinate $v_b$ of a scanning window in the image plane. Using the mathematical relation in equation (6) we draw the conclusion that if $v_b$ is large than we consider scanning windows of higher dimension and if $v_b$ is medium to small we consider scanning windows of smaller sizes. This assumption was verified also by the ground truth annotations in the datasets we have analyzed.

Figure 2 shows the variation of bounding box heights with respect to the bottom $v$ coordinate in the image plane, for three of the datasets we have used. We have made this analysis for all the datasets used in the experiments and we have noticed the same variation trend for all.

Hence we have defined specific ranges for each pedestrian height. Small variations of these ranges are allowed. Figure 4 shows how the number of scanning windows was reduced after the spatial localization filter has been applied. Figure 4(a) depicts all the scanning windows that can be generated for 18 scales. A point in the figure corresponds to a scanning window. Figure 4(b) shows the distribution of scanning windows after the spatial localization filter was applied.

From figure 4 one can deduce how we have chosen the height with respect to the $v$-bottom coordinate. For example for a $v$-bottom coordinate in the range 175–375 pixels we have considered a height in the range 100–140 pixels, for a given $v$-bottom coordinate in the range 225–400 pixels we have considered scanning window heights in the range 140–160 pixels and so on.

The spatial localization filter reduces the number of scanning windows with about 40%.

B. Edge information filters

A main characteristic of pedestrians in traffic scenes is that they are usually up-standing. This premise allows us to analyze the vertical and horizontal edges and the information they bring in the pedestrian localization model. We extract vertical and horizontal edges using Sobel operators. We model the local information that is provided especially by the vertical edges by the definition of an energy function. We divide the vertical edge image into overlapping blocks of equal size. For each block we compute the edge intensity distribution function $p$ as follows:

$$p(b) = h(b)/\sum_{b=1}^{\text{bins}} h(b)$$ \quad (7)

where $h$ is the histogram of intensity levels in the block and bins represents the total number of gray levels (256). High intensity levels in the edge image mark strong edge points, while low intensity values mark weak edge points.

Next we define an energy function in each block as follows:

$$E = \sum_{b=0}^{\text{bins}} (p(b))^2$$ \quad (8)

The energy is low when the number of strong edges is high. The energy is maximum when we have a large number of weak edge points in the block. The proposed energy function measures the homogeneity or the uniformity of edge
information of a block. A block is homogenous if there are no edges or if there is a uniform variation of the edge intensity levels. By definition the energy is high when we have no edges hence those areas will be removed.

We propose another filter that removes blocks having low energy and small standard deviation. That is we have a large number of edge points but there is a small difference in their intensities. This filter removes mostly vegetation parts from a traffic scene.

Another local filter analyzes the density of horizontal edges in a block. If the number of horizontal edges in the block is high (if we have more than 90% of horizontal edge points) and the standard deviation of edge intensity is low than that block is removed. This filter has the role of eliminating regions from the road. Figure 3 shows two vertical edges in a scanning window. This measure was also used by [15]. We analyze a group of adjacent windows of the same scale. For each group we keep the window that has the greatest density of connected vertical edges. Figure 4(c) shows how the number of scanning windows is reduced after the local maxima filter is applied.

We have experimented with several configurations of adjacent windows. For example a local maxima filter of two windows considers a window having the top left corner at position \((x, y)\) and its neighbor is at position \((x, y + stride)\). A local maxima filter of nine windows takes into account the scan windows at positions \((x, y), (x, y + stride), (x, y - stride), (x + stride, y), (x - stride, y), (x - stride, y - stride), (x + stride, y + stride), (x - stride, y + stride), (x + stride, y - stride)\).

Depending on the number of adjacent windows used the scan window local maxima filter prunes the search space with 20% up to 40%.

IV. EXPERIMENTAL RESULTS

We have applied the proposed scan window removal method on four reference pedestrian datasets. In these four datasets the annotations contain person, a group of persons, occluded person or groups and bicyclists and motorcyclists. We have performed our evaluation on two test cases: person and occluded person. We have chosen these two because a group is formed of several persons hence if the person is detected correctly the group will also be detected.

For each of the test cases we have measured the scan window reduction factor and the hit-rate of the pruning algorithm. We have moved the scan windows with a stride of \(4 \times 4\) pixels, and due to the framework in which we have integrated our pruning algorithm we have chosen an aspect ratio of 0.5 for the scan windows and 28 different scales. Each scan window is up-sampled with a scale factor of 1.05.

A. Scan window reduction factor

Table I shows the evolution of the removed scan windows by our pruning strategy. The localization filter removes about 37% of the scan windows. On the next column we have the localization filter and the edge based filter that together remove about 45% of the scan windows. On the next columns we have the localization filter plus the edge based filter and several configurations of the local maxima filter.

<table>
<thead>
<tr>
<th></th>
<th>spatial localization</th>
<th>edge filters</th>
<th>local max 2 windows</th>
<th>local max 4 windows</th>
<th>local max 9 windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent of scan windows removed</td>
<td>37.2%</td>
<td>45.2%</td>
<td>68.6%</td>
<td>64.34%</td>
<td>93%</td>
</tr>
</tbody>
</table>

TABLE I

SCAN WINDOW REMOVAL PERCENT

B. Accuracy of the localization

A ground truth bounding box \(BB_{GT}\) is hit by a scan window \(SW\) if their overlap factor is greater than a threshold.

\[
a_0 = \frac{\text{area}(SW \cap BB_{GT})}{\text{area}(SW \cup BB_{GT})} \geq T
\]
For measuring the hit rate we have considered only ground truth bounding boxes having a height greater than 50 pixels.

We have varied the values of the threshold $T$ in the range 0.5…0.95 in order to show how the accuracy of the localization depends on the overlap factor. From figure 5(a) one can notice that for a threshold from 0.5 to 0.7 the hit rate is more than 90% for the person instances. The lower hit rate when the threshold is increased can be motivated by the fact that we use a fixed aspect ratio of 0.5 and the aspect ratio of pedestrians in the four datasets varies from 0.2 to 0.8. We have chosen this fixed aspect ratio because of the detection framework in which we have integrated our pruning strategy. Nevertheless even if we choose an overlap threshold of 0.65 the detection rates of pedestrians are very good (see Figure 6).

For the person occluded case the hit rates are lower. For an overlap threshold equal to 0.55 the hit rate for Caltech occluded persons is 0.98, while for Daimler person occluded the hit rate is 0.72. These rates are due to the aspect ratios of scan windows. Usually occluded persons have aspect ratios that vary from 0.1 to even 1.5. With an overlap threshold of 0.55 the detection rate of occluded pedestrians are high when integrated in the ACF detection framework [1].

C. Localization combined with detection

We have integrated our pruning strategy in the framework provided by [1]. This framework uses Aggregated Channel Features (ACF) [3] for performing pedestrian detection. It provides an evaluation methodology for Caltech data set. From figure 6 and table II one may notice that the ACF combined with the localization filter has a better performance than the original ACF and runs faster.

The ACF combined with localization filters, edge filters and local maxima filter on a group of $2 \times 1$ neighboring scan windows has the performance the same as ACF and keeps the execution time.

If we consider more neighbors in the local maxima removal filter the performance is decreased (up to 53%), but as one may notice from table II the execution time is improved.
D. Time performance

Our experiments have been run on an i7–3770K CPU. All the images have a dimension of $640 \times 480$ pixels.

In the context of the ACF pedestrian detector [3] we have analyzed how the execution time is influenced by the pruning strategy we propose.

The ACF method runs at about 32fps. We have measured the influence of the filters contained by our scan window removal strategy on the execution time.

<table>
<thead>
<tr>
<th>Method</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF</td>
<td>32fps</td>
</tr>
<tr>
<td>ACF and Localization</td>
<td>34 fps</td>
</tr>
<tr>
<td>ACF and localization and edge based filter and local maxima for 2 windows</td>
<td>32 fps</td>
</tr>
<tr>
<td>ACF and localization and edge based filter and local maxima for 4 windows</td>
<td>34 fps</td>
</tr>
<tr>
<td>ACF and localization and edge based filter and local maxima for 9 windows</td>
<td>36 fps</td>
</tr>
</tbody>
</table>

One may notice that ACF plus the localization filter works faster. It works faster because we have not implied any additional feature computation for it, just the pyramids implied by ACF.

The computation of the features (energy, density of vertical edges) used in the scan window removal method takes about 6ms per frame. But, this additional execution time is compensated by a lowered execution time at the scan window procedure, as shown in table III.

<table>
<thead>
<tr>
<th>Method</th>
<th>Execution time of scan window procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF</td>
<td>12ms</td>
</tr>
<tr>
<td>ACF and Localization</td>
<td>10 ms</td>
</tr>
<tr>
<td>ACF and localization and edge based filter and local maxima for 2 windows</td>
<td>9 ms</td>
</tr>
<tr>
<td>ACF and localization and edge based filter and local maxima for 4 windows</td>
<td>6 ms</td>
</tr>
<tr>
<td>ACF and localization and edge based filter and local maxima for 9 windows</td>
<td>4 ms</td>
</tr>
</tbody>
</table>

TABLE III

Execution time of ACF combined with scan window pruning filters

V. Conclusions

We have presented an algorithm for scan window based pedestrian recognition methods improvement by search space and scale reduction. The algorithm is tuned for pedestrian detection in monocular images of traffic scenes. The method exploits the geometry of a traffic scene that induces some dimensionality constraint based on the position of the pedestrians in the image. The proposed method strives to determine the scale and dimension of relevant scanning windows that are very likely to overlap pedestrian regions. The method is refined by several edge based filters that have the role of removing homogeneous parts of the image such as the sky, uniformly varying areas such as vegetation, areas having a high density of horizontal edges such as the road and enhance the windows having a high density of connected vertical edges. We have included the presented filters in an existing pedestrian framework and we obtain a high search space reduction, we have a negligible loss in performance and an improvement the execution time.

Acknowledgments

This work was supported by the Romanian Ministry of National Education, UEFISCDI, project number PNII-PCCA 18/2012 SmartCoDrive.

References