

Day and Night Pedestrian Detection Using Cascade AdaBoost System

Pietro Cerri, Luca Gatti, Luca Mazzei, Fabio Pigoni
VisLab – Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma, ITALY
www.vislab.it
{cerri, lucag, mazzei, pigoni}@vislab.it

Ho Gi Jung
Central R&D Center
MANDO Corporation
Yongin-Si 446-901, Korea
hgjung@mando.com

Abstract—This paper presents the results of an all-day-long pedestrian classification system based on an AdaBoost cascade meta-algorithm. The underlying idea is to use a Haar-features-based AdaBoost together with an ad-hoc-features-based AdaBoost system in order to reach a better pedestrian classification. A specific night-time pedestrian classification is developed in order to obtain a system that can be used also in poorly illuminated environments. These classifiers are joined together using a cascade AdaBoost system that uses the output of the previous classifiers to obtain a final classification for the area. In the paper the night time and the ad-hoc features systems are presented together with the cascade classification and quantitative results.

I. INTRODUCTION

Pedestrian detection using computer vision is by no means trivial: people wear clothes which have different colors and shapes and it is difficult to distinguish pedestrians from the surrounding environment. Moreover, a pedestrian may be located in various positions and frequently carries objects (bags, umbrellas, boxes, etc.) which change his/her shape.

Different methods are developed to reach the best classification: these approaches are based on the full search for pedestrians in the whole area in front of the vehicle. Potential candidates are located using pedestrian characteristics [1] such as shape, symmetry, texture, motion, periodicity of human legs motion. Then the candidates are validated using different classification methods.

It is important to note that another step might be needed: i.e. the removal of pedestrians that do not represent danger. This step needs to assess the position and direction of the pedestrian [2] with respect to vehicle motion, the probability for the pedestrian to change direction and velocity, and other *environmental* and *behavioral* characteristics that are intrinsically tied to the pedestrian nature and the environment in which the pedestrian itself is moving.

The approach described in [3] is different from conventional ones, because it is focused on the detection of situations of clear danger only, by limiting the search to specific areas. Using a laser scanner it is possible to reconstruct the environment, locate dangerous areas, and search for pedestrians in such areas. This article describes a classifier specifically developed to be used in this project.

Even if the approach is innovative, the classification is anyway a critical aspect of the problem. Different methods can be used to reach a good classification. Methods based

on AdaBoost meta-algorithm [4] and Support Vector Machine [5] are the most used. Both of these methods are based on a previous training stage. In order to train the classifiers, a number of pedestrians and non-pedestrians are extracted from images of different sequences: these images are used to extract the best weak classifier at each step of the training and to provide a weight to it. For the development of the proposed classifier, a specific training phase is performed: areas provided by laser scanner-based environmental analysis are used as input, after a manual classification.

The first version of this classifier was based on Haar features only [6]: however in order to reach a better classification the algorithm mixes Haar features and ad-hoc features classification using AdaBoost. 12 specific features have been developed. Features are extracted from properly pre-processed bounding boxes with an edge extraction and binarization function and from the edges histograms.

These weak features are created in order to use a less generic alternative to Haar features to be used with AdaBoost meta-algorithm. Moreover, adding a new classifier is a way to improve the performance of the pedestrian collision avoidance system in an urban environment.

To develop this kind of features a number of images extracted from different sequences were examined. The considered sequences were recorded in urban environments with different lighting conditions (sun, rain, morning, afternoon, night, fog, urban boulevards etc.) and rural environments. A lot of acquisitions were necessary to obtain a wide variety of possible cases. An example of processed images is shown in figure 1.

Images obtained during night, using a NIR illuminator are really different from day time ones; a specific method to detect pedestrians during night and in very low illumination condition is developed. This method is based on pedestrian shape detection.

In this paper the Haar features method is not described, even if the one developed is slightly different from traditional one, because it does not only classify between pedestrians and non-pedestrians, but into partially occluded pedestrians as well: it will not provide two opposite votes, but three independent votes. Ad-hoc features are presented in section II together with night time method in section III and a cascade algorithm to merge all the results in section IV.

II. AD HOC FEATURES

In this section ad-hoc features developed for pedestrian detection are described. These features are based on a typical pedestrian shape and body parts aspect. To detect a pedestrian, the algorithm provides a vote to each feature of the following list:

- 1) Open Legs shape
- 2) Left step legs, little mask
- 3) Right step legs, little mask
- 4) Left step legs, big mask
- 5) Right step legs, big mask
- 6) Head top, little mask
- 7) Head top, big mask
- 8) Head, with circle formula
- 9) Pedestrian's M function
- 10) Horizontal border function
- 11) Tree and pole search function

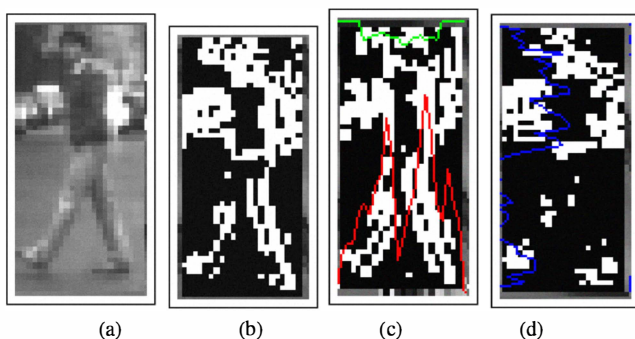


Figure 1: (a) bounding box framing a pedestrian, (b) Sobel image, (c) vertical Sobel with vertical histogram, (d) horizontal Sobel with horizontal histogram.

Figure 1 shows a bounding box containing a pedestrian (a), and images obtained by subsequent processing (Sobel filter and binarization) (b) to (d): ad-hoc features were built considering these pre-processed images. The Sobel image (b) is used to search for features based on pattern matching, in order to detect the head and legs, corresponding to the classifiers from 1 to 8. Vertical and horizontal edges images, (c) and (d), are used to compute the histograms, which, in the figure 1, are overlapped onto the images: histograms are used to calculate the last three features listed in the above table.

1) *Legs shape on maximal aperture*: The legs shape of a pedestrian crossing the road in front of the vehicle is a quite strong feature. At the time of the step change, human legs draw the profile of a V upside-down. The algorithm searches for this feature by computing a match between the image and a mask, overlapping and shifting it towards the bottom part of the bounding box.

Figure 2 (a) shows the mask used for comparison with the edge binarization. This mask has a size of 10×9 .

2) *Forward step legs shape*: The use of a single mask is not enough as the shape of the legs changes while walking. Other features are based on pattern matching with mask that

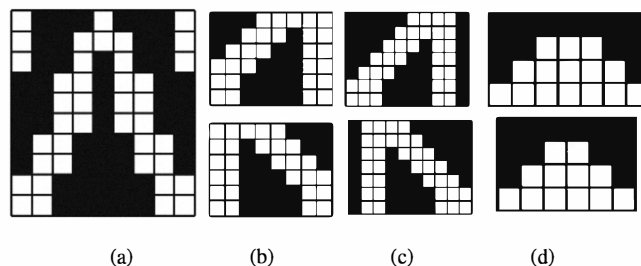


Figure 2: Masks collection for features matching. (a) Open legs. (b) Little mask for step. (c) Big mask for step. (d) Head mask.

represent legs in a step to the right or to the left. Four different types of masks are used (see figure 2 (b) and (c)). These masks are of two types: small, with size 8×6 pixels, and large, with size 10×7 pixels. Even in this case masks are overlapped and shifted towards the bottom part of the bounding box to calculate the number of matches between them and the image.

3) *Head Presence*: An unfailing characteristic of a pedestrian is the head. It seems very easy to locate, but in pictures with a weak contrast and a very poor detection of edges it is not so simple: pedestrians illuminated only by vehicle headlights are often very bright in the lower part (legs) and darker in the upper part (head and trunk). Even in daylight, the head might be not visible because hidden by umbrellas, hats, or other occluding objects. However, in most cases the head is cleanly visible, so it is useful to insert features for its search within the bounding box. We use two masks to search for the top of the head and a feature to search for a circular shape. Used masks are illustrated in figure 2 (d), and they have size 6×4 and 7×4 pixels. Only the upper part of the head is searched using these masks, because a match with a complete circular pattern can provide a too high number of false detections. A complete circle is searched as well; the circle has a radius of 3 pixels with a total number of pixels of 16; the search for these features is made in the first upper third of the box.

4) *M shaped pedestrian histogram*: In the vertical edges histogram of the image including a pedestrian who has a well-defined shape, it is possible to notice the presence of a shape similar to an 'M'.

Figure 3 (a) shows pedestrian images and the corresponding Sobel extracted images. The red lines show the shape of the 'M'. Considering different images in different sequences, it is possible to compute reference values, which describe the regularity of the M-shape; the position and value of left and right maxima and local minima are computed. The comparison of each computed value with the reference value is used to get a final value in the $[0, 1]$ interval.

5) *Horizontal borders*: The histogram of the horizontal edges obtained from the binarization of horizontal edges (represented in figure 1 (d) where the histogram is drawn as a blue line) is used to compute the arithmetic average value of the higher third of the histogram.

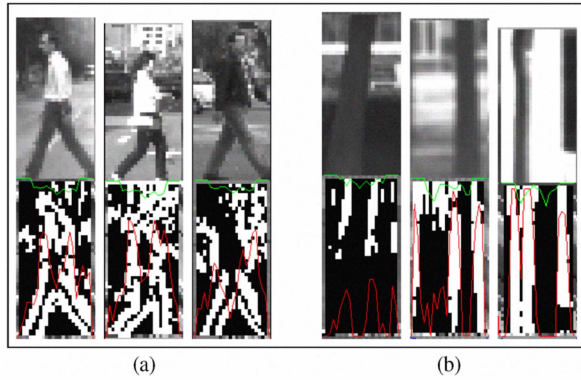


Figure 3: Examples of bounding boxes with (a) pedestrian and (b) tree.

This feature allows to discard all bounding boxes that do not show horizontal edges in the upper part of the bounding box: this could happen in case a tree or a pole is framed. On the contrary vehicles have a high presence of horizontal borders.

6) *Tree*: Often, classifiers might confuse trees and poles with pedestrians because they have some common characteristics: they are both vertical objects and fill the bounding box vertically. Figure 3 (b) shows some examples of bounding boxes containing trees, together with their corresponding vertical histogram (red line).

With this information we obtain an additional feature that indicates the possibility of having a tree or a pole, in a bounding box.

In the case of presence of two maxima in the vertical edges histogram, as shown in the first image of figure 3 (b), the minimum between the peaks is searched: in this case the likelihood of the presence of a tree is computed. Otherwise, if a single maximum is present, the presence of a pole is checked. In both the cases a vote between 0 and 1 is assigned, considering peaks and minimum values.

A. Features validation

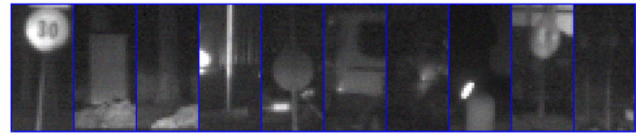
Trying to find a suitable features set is a complex work, but it is possible to obtain an information of the validity of this set by a statistical computation. The One Way ANOVA analysis was used to investigate it. This analysis was performed on our own pedestrian training sets formed by daily and nightly pedestrian and non-pedestrian images. One Way ANOVA was applied to every feature. The described features, analyzed with this method, are able to discriminate the above classes, both in night and day sequences. A correct classification is not possible using a single feature or by an analysis of the image with a subset of them. An AdaBoost meta-algorithm is used to reach a correct classification using these weak classifiers. An implementation of AdaBoost is developed: for each feature selected by AdaBoost a weight is assigned, together with threshold values and threshold types. 4 types of threshold are used:

- minimum (1 threshold)

- maximum (1 threshold)
- included (2 thresholds)
- excluded (2 thresholds)

III. NIGHT TIME

In this section a pedestrian localization system specifically designed to work at night in poorly illuminated areas, is described. Figure 4 shows some examples of images that will be analyzed to detect the presence of a pedestrian.



(a) Images without pedestrians that will be discarded



(b) Pedestrians

Figure 4: Some examples of regions of interest to be analyzed

The algorithm, described in the following, is based on the assumption that, thanks to NIR illuminators, pedestrians and other standing objects represent the brightest areas in the image with the exception of light sources. Therefore, after removing light sources, it is possible to detect obstacles with a binarization process and classify them according to their shape.

The evaluation process of the input image is divided into several phases. First of all, preprocessing phases are performed on each region of interest provided by a previous laser scanner analysis; this phase is composed by: light sources filter, contrast stretching, median filter, and binarization. After these operations, the procedure continues to check for pedestrian bodies and legs by respectively analyzing the upper and lower half of the image. Both body and closed legs search algorithm can be condensed in a single phase, while the search for open legs is more complex and is divided into several stages: skeletonization, straight line finder, and analysis of correspondence with models. The various operations performed by the complete algorithm are shown in figure 5.

A. Preprocessing

The first step of the algorithm consists in detecting light sources filter. It is complex to isolate light sources and distinguish them from bright pedestrians. Therefore, only very bright areas, with aspect ratio close to 1 and with a high brightness in the center decreasing on the borders are removed from the image. In figure 6 two examples of filter application are shown.

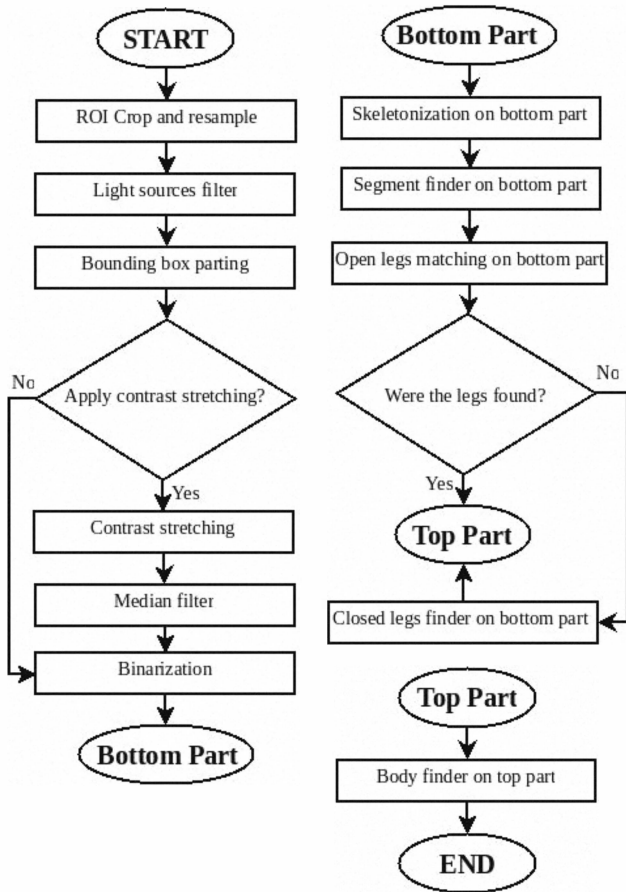


Figure 5: Algorithm flow chart

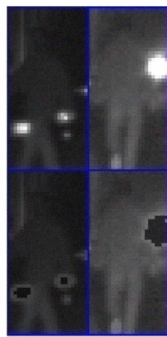


Figure 6: Light sources filter

The following step is contrast stretching. This operation is performed under certain conditions:

- 1) the difference between the value of the lighter and darker pixel must be under a certain threshold
- 2) the histogram should have two peaks at the extremes

The contrast adjustment makes the picture appear noisy, therefore a median filter is also inserted, in order to remove the impulsive noise, making the image more uniform and easier to use in the following phases of the algorithm. The following step is binarization, which is critical, since the result obtained from this phase is the input of the final part of pedestrians search. The percentage method is used, i.e. the pixels that

become white are a fixed percentage of the total, respectively for the lower and upper part of the bounding box.

B. Open legs search

The first part of the open legs search consists of striped skeletonization, during which the image is analyzed row by row starting from the bottom, in order to get a result like the one shown in Figure 7.



Figure 7: Binarization and skeletonization result for open legs search

The subsequent curves search algorithm is divided into two phases: the first phase consists in finding the first point of a curve, the second phase is repeated iteratively to join all pixels that are vertically connected until the last in the curve is found. Considering the first and the last point of the curve, a line can be interpolated: the output of this process is a set of lines identified by their lengths and their angular coefficients: these data are easily manageable and will be used for comparisons with pre-defined open legs templates. The final step consists in measuring how much the legs are far apart; it is measured using the length of the legs and the distance between the feet.

C. Bodies and closed legs search

These two processes are very similar, differing only in some details and thresholds. The captured images contain a very low level of detail and it is often inadequate for a complex analysis. For this reason a simple procedure is implemented that tries to make the most of the few details that characterize the pedestrian.

The algorithm operates computing integral of vertical histogram to detect the area where a peak can be detected. In this area, a horizontal histogram is computed, and a peak is searched for. Figure 8 shows the steps of the body search.

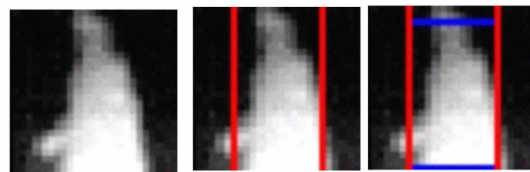


Figure 8: Steps of body search algorithm

Finally, the figure is selected vertically, using the horizontal histogram. Obviously, the histograms of the body and legs are analyzed in different ways. Concerning the body, for example, it is required to have its representation start line at the bottom of the above bounding box and a simple algorithm is used to validate the shape, which should be wider at the bottom and narrower at the top, meanwhile, the figure that represents closed legs must start from the topmost row of the bottom bounding box and it must have a more or less constant width.

D. Votes management

At the end of the algorithm four rates are assigned, which range from 0 to 1. The votes are:

- Vote of the body according to the correspondence with a pedestrian body model and the dimension of the possible body found
- Vote of the legs according to the correspondence with a model of open or closed legs and the dimension of the possible legs found
- Open legs rate which is obtained with the ratio between feet position distance and maximum feet distance
- Pedestrian vote which is obtained making an average of the body and legs vote. The vote of the legs has more weight if they are open because it is a dominant characteristic of a pedestrian.

All these ratings will be analyzed at later stages of the application to properly determine if the figure is recognized as a pedestrian.

IV. CASCADE

To increase the potential and the recognition performance, an additional stage, called AdaBoost cascade, was developed. In this stage the votes from the classifier implemented with Haar features and with ad-hoc features are merged and combined together to obtain an additional level of classification. The votes provided by the previously described algorithms are used as input to make a second kind of AdaBoost classifier.

As the number of outputs from previous classifier is very low (2 for the Haar method and 1 for the ad-hoc method), and some tests pointed out that they can not be used to reach an efficient classification process by their own, non linear combinations of these values are added as input of AdaBoost algorithm.

According to the classic voting approach, a final hypothesis resulting formula is implemented: final vote is computed as the weighted sum of the classifiers selected by AdaBoost training stage.

Different combinations are developed and used as input to the classifier. the firsts three votes correspond to the previous classifier outputs mentioned above. The other votes are weighted average of input (with different weights) and weighted combination normalized. A total of 33 values are computed.

Some tests were performed on night sequences, adding night time algorithm results as input. Classification performance were not increased. Night images acquired in illuminated environments, are processed using AdaBoost cascade. Night images acquired in not illuminated environments, are processed using night time specific algorithm only.

V. RESULTS

After the training phase, results of developed algorithms are assessed with test sequences. Ground truth on the sequences was previously manually collected. Three different test sets are considered: one for day time scenes, one for night time scenes

with poorly illuminated environments, and one for night time scene with properly illumination.

Figure 9 shows ROC curves for ad-hoc features, Haar features, and cascade method. The algorithms based on Haar features reach more precise results than algorithms based on ad-hoc features, but it is important to note that the cascade method, that merges the two methods, can reach a good classification rate.

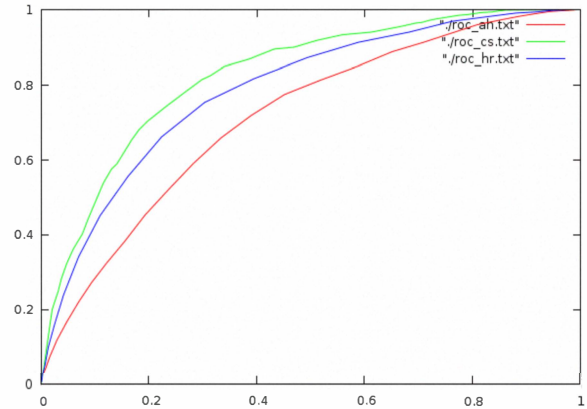


Figure 9: Day-time ROC. Red line refers to ad-hoc features results, blue line to Haar features results, and green line to cascade results

Results obtained on night time scenes with proper environmental lighting are similar to the previous one: some tests also pointed out that the use of specific night time algorithms produce worse results.

Otherwise, specific night time algorithms reach very good results with poorly illuminated environments. Figure 10 shows ROC curves of night time algorithm. Adding a cascade phase, results are slightly increased, but the little improvement does not justify the use of another classification phase.

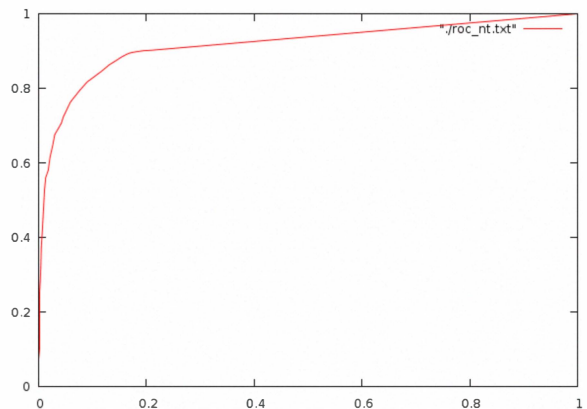


Figure 10: Night time ROC.

Frame processing time changes according to the chosen method and to the number of regions of interest which are analyzed: anyway the execution time on a good performing

PC is always lower than 70 ms/frame, considering also the laser scanner preprocessing.

VI. CONCLUSIONS

Results obtained with the proposed system are good: the system works both during day and night with specific algorithms for the two situations. Proposed ad-hoc features reach promising results, and they can be used to enforce the results of other classifiers, as shown in the results section. Night time results are good, and the processing time of night time system is very low. Some tests on new specific features can be performed: the current set is a base for an alternative set of features to be broaden and strengthen in the future.

VII. ACKNOWLEDGMENTS

The authors gratefully thank Alberto Broggi for his support and review of this project.

REFERENCES

- [1] T. Gandhi and M. M. Trivedi, "Pedestrian Protection Systems: Issues, Survey, and Challenges," *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 413–430, Sep. 2007.
- [2] T. Gandhi and M. Trivedi, "Image based estimation of pedestrian orientation for improving path prediction," in *Procs. IEEE Intelligent Vehicles Symposium 2008*, Eindhoven, Netherlands, Jun. 2008, pp. 506–511.
- [3] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. G. Jung, "A New Approach to Urban Pedestrian Detection for Automatic Braking," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 594–605, Dec. 2009, ISSN: 1524-9050.
- [4] L. Leyrit, C. Chateau, C. Tournayre, and J.-T. Lapresté, "Association of AdaBoost and Kernel Based Machine Learning Methods for Visual Pedestrian Recognition," in *Procs. IEEE Intelligent Vehicles Symposium 2008*, Eindhoven, Netherlands, Jun. 2008, pp. 67–72.
- [5] A. Broggi, M. Bertozzi, M. Del Rose, M. Felisa, A. Rakotomamonjy, and F. Suard, "A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2007*, Seattle, WA, USA, Sep. 2007, pp. 144–148.
- [6] P. Cerri, A. Broggi, L. Gatti, P. Grisleri, and H. Jung, "Scenario-Driven Search for Pedestrians aimed at Triggering non-reversible Systems," in *Procs. IEEE Intelligent Vehicles Symposium 2009*, Xi'an, China, Jun. 2009, in press.