RADAR-VISION FUSION FOR VEHICLE DETECTION BY MEANS OF IMPROVED HAAR-LIKE FEATURE AND ADABOOST APPROACH

Anselm Haselhoff 1, Anton Kummert 1, and Georg Schneider 2

1Faculty of Electrical, Information, and Media Engineering, University of Wuppertal
Rainer-Gruenter-Str. 21, 42097 Wuppertal, Germany
email: {haselhoff, kummert}@uni-wuppertal.de
web: http://www.uni-wuppertal.de/

2Elektronik Vorentwicklung, Audi Electronics Venture GmbH
Sachsstr. 18, 85080 Gaimersheim, Germany
email: georg2.schneider@audi.de
web: http://www.audi.de/

ABSTRACT
This work describes a vehicle detection system that uses fusion of vision and radar data. The radar provides a first estimation of the lateral position of vehicle candidates and the related information. This information is used to define a region of interest (ROI) that is subject to verification. A video camera is used for the verification purpose. The projection of the ROI onto the image plane is scanned via an AdaBoost object detection algorithm, and thus radar detection can be verified and more specific data of the vehicle’s 3D position and width can be given.
Moreover, the distance information provided by radar is used to choose optimal parameters during the visual detection process, e.g. properties of the scan window and parameters for fusing detections.
In addition, mutual information for haar-like feature selection is used to increase detection rates.

1. INTRODUCTION

More than 50 years ago, automotive industry started to develop methods and systems to assist vehicle drivers. Today’s vehicles contain driver assistance systems, which observe the environment of the vehicle to provide road traffic with more comfort, safety, and efficiency.

The automotive industry develops comfort systems that are mainly radar-based, for example adaptive cruise control systems. Radar-based sensors show potential for optimization in robustness and reliability. A major challenge in designing radar-based systems is the suppression of ghost targets that are caused by reflections. Furthermore, none vehicle objects like guard rails can trigger radar sensors as well. A data fusion with further sensor technology is worthwhile to solve these ambiguities. Amongst others, video-radar fusion is a subject of current research in the vehicle detection context [3], since video sensors enable sophisticated object classification.

The weakness of radar-based systems is the lack in classification ability and that the radar reflex could be received from an arbitrary spot of the vehicle; in worst case, one of the vehicle side panels. Thus important information for driver assistance systems is missing, like accurate location of the vehicle center and the vehicle width. The video sensor can be used to extend the system with information about accurate lateral position, object width, and classification ability for verifying radar detection.

The video camera facilitates application of various object detection algorithms. Appearance-based methods are state-of-the-art in vehicle detection and generally applicable to object detection problems. These methods learn the characteristics of vehicle appearance from a set of training images which capture the variability in the vehicle class [11]. Different combinations of feature extraction methods and learning algorithms are proposed.

Schneider et al. [8] use a biologically inspired hierarchical vision model that determines optimal features and nonlinearities of the visual hierarchy by evolutionary optimization. The classifier is based on the simple nearest-neighbor concept. Sun et al. [10] follow a more common approach by using haar wavelets and gabor filters for feature extraction and a support vector machine for classification. The system developed by Ponsa et al. [7] is an adaptation of the Viola & Jones object detection system that uses haar-like features and an AdaBoost learning algorithm. They determine 3D information of vehicles with a single camera by estimating, where the horizon is projected onto the image plane. Feasible image regions for searching vehicles are determined by assuming a flat road.

In this work we generally follow the approach of Ponsa et al. [7]. Instead of determining search areas by assuming a flat road we use the regions of interest defined by radar. In addition, radar enables improved measurement of the distance and position information. In combination with mutual information for haar-like feature selection, introduced by Shen et al. [9], we achieve significantly improved results. To gain invariance under illumination conditions we propose contrast instead of gray-level variance normalization. Moreover, we developed a new method of fusing overlapping detections based on distance information.

2. GENERAL OVERVIEW OF SENSORS AND ALGORITHMS

The input of our vehicle detection system is provided by four sensors. Three radar sensors are placed at the front of the car and a monochrome camera with a resolution of 752 × 404 pixel and 43.5°/23.5° horizontal/vertical field of view is mounted behind the rear-view mirror. The camera resolution enables an object detection up to a distance of approximately 60 meters.

The visual object detection presented in this paper operates on a frame by frame basis. That means, radar informa-
tion is allocated to each frame and the positions of the radar reflexes are projected onto the image plane. A region of interest is defined as a rectangular area of approximately five meter width and four meter height around the projected radar reflex. To avoid mutual occlusion, at first the closest ROIs are examined, followed by a hidden surface determination.

Each ROI is passed to the visual detection system, which is based on the approach of Ponsa et al. [7]. To learn the characteristics of vehicle appearance, vehicle rear views and negative-samples are used as training images. Some samples are shown in figure 1. These images are subject of feature extraction. The new image representation or feature vector is obtained by applying several filters to different positions on the image.

This representation is used for training of the classifier by means of an AdaBoost algorithm. AdaBoost performs a feature selection and combines the selected features as simple weak classifiers to a strong one. In each iteration step of the AdaBoost algorithm a weak classifier with the smallest weighted classification error is selected. Each weak classifier is just dependent on one component of the feature vector and the classification is done via a simple threshold comparison.

The feature selection process is very important with respect to computation time of the final classifier. The computation time can be optimized by organizing strong classifiers of progressively increasing complexity in a cascade. The idea is to reject sub-windows containing no vehicles with just a few operations (features) [12]. Only sub-windows containing vehicle-like objects are passed through the entire cascade by means of an AdaBoost algorithm. AdaBoost performs a weighted classification error is selected. Each weak classifier, \( \alpha_t \) is the weight of the \( t \)th weak classifier and \( T \) is the number of features selected. The weak classifiers are combined by a weighted majority vote to a strong classifier \( H \).

After the offline learning process only a few selected features must be calculated for online classification of a ROI. The trained classifier is applied to reasonable sub-windows of the ROI only. Due to the distance information of the radar we can introduce some constraints of minimal and maximal detector size. For each sub-window the classifier gives a vote whether a vehicle is present or not. Since the classifier is invariant against small changes in translation and scale, generally more than one detection occurs around a present vehicle. These overlapping detections must be fused.

Once a vehicle is detected, the width in meter and accurate position can be estimated by fusing radar and vision data.

3. FEATURE EXTRACTION

3.1 Preprocessing

To gain invariance to different illumination conditions and camera influences, we tested several gray-level transformations. Amongst others, histogram equalization [4], gray-level variance [12], and contrast normalization were tested. With respect to our test images, we achieved the best results performing contrast normalization on the whole scene, which has the effect that the whole gray-level range is covered.

The general idea of the contrast normalization of the image \( I(x, y) \) to the range \([0, 1]\) can be expressed as

\[
I'(x, y) = \frac{I(x, y) - \min[I(x, y)]}{\max[I(x, y)] - \min[I(x, y)]}.
\]

Another necessary preprocessing step for the training process is a normalization of the image size, e.g. to a resolution of \(24 \times 18\). This is required, because images of different resolution have a different number of computable features. For example, we tested bilinear transform to guarantee the same image size. However, it turned out that it is a better way to apply the filters just to certain sampling points and use all points of the original image for calculation of the filter response. This approach leads to improved true positive rate of more than four percentage points. Figure 2 illustrates this approach. Ponsa et al. [2] perform an aspect ratio and a size normalization on the training data. In contrast, we retain the original image size and adjust the filter position and scale respectively.

3.2 Rectangle features

In the object detection system developed by Viola & Jones [12] haar-like features are proposed, called rectangle features. The advantage of these features is a very fast computation due to the use of a new image representation, the integral image. These features encode characteristic vehicle properties like edge and symmetry information.

For the training process, an exhaustive set of features is used from which the AdaBoost algorithm can select the most...
Figure 3: Five basic types of rectangular filter masks.

important ones. The feature values are obtained by applying the filters in different scales to varying positions on an image. The five basic types of rectangular filter masks [7] are shown in figure 3 and are defined as follows:

- Two rectangular regions (horizontal/vertical): The response is the difference between the sum of pixel values within the two regions.
- Three rectangular regions (horizontal/vertical): The response is the sum within the two outside regions subtracted from the sum in the center region.
- Four rectangular regions: The response is the difference between the rectangular pairs.

4. USE OF MUTUAL INFORMATION

The idea of AdaBoost was to train a number of weak classifiers, which are combined to form a single strong classifier. In each iteration of the AdaBoost algorithm the weak classifier with smallest weighted error is selected. No guarantee can be given that this choice is optimal for the combined strong classifier. It is likely that features encoding redundant information are selected.

To avoid these redundant features, Shen et al. [9] propose the use of mutual information (MI) for redundancy elimination. Each time a new weak classifier is selected the maximal MI to one of the previously selected is determined. The classifier with smallest weighted error is accepted, if a certain value of MI is not exceeded.

Mutual information for two random variables \( X \) and \( Y \) is defined as:

\[
MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}
\]

where \( p(x) \) and \( p(y) \) are probability density functions and \( p(x,y) \) is a joint density function. The density functions can be estimated by examining the training data. To apply the MI measure, the weak classifiers \( h_i \in \{0,1\} \) have to be considered as random variables. Afterwards, the mutual information can be used to measure the dependency of two classifiers \( h_i \) and \( h_j \).

Shen et al. use an MI measure to improve the AdaBoost learning in context of gabor feature selection for face recognition. We tested this approach in rectangle feature selection for vehicle detection.

5. DETECTION PARAMETERS AND FUSION OF OVERLAPPING DETECTIONS

The detection of vehicles requires to scan certain sub-windows of the ROI. Each region is passed to the classifier, which gives a prediction, whether a vehicle is present or not. The resulting detections are subject to fusion. The following will explain how we define the scanning regions and how we fuse the overlapping detections.

5.1 Detection parameters

First, we define a scan window with minimal/maximal width and height, e.g., we bound the maximal window width to 2.55 meters and the height to 2.00 meters. With the distance information provided by radar, we can convert the width from meters into a pixel value. This window is moved across the ROI with a step size \( \Delta \). \( \Delta \) determines the number of pixels to shift the window. Optimal values for \( \Delta \) can be determined by using the distance as well.

It is obvious that for vehicles in short distance a greater value for \( \Delta \) can be chosen, without the risk of missing a sub-window containing a vehicle. Likewise to maximal window width we set a fixed step size of 10 centimeters and calculate \( \Delta \) in pixel values dependent on the distance.

Another parameter is the increasing factor \( \Lambda \) that defines, how the size of the scan window is changed in the range of maximal and minimal dimension. After the detector has scanned all windows of width \( w \), the window width is increased to \( w' = w \cdot \Lambda \). With respect to distance, \( \Lambda \) values in the range of \([1.01, 1.25]\) are used.

5.2 Fusion of detections

The classifier has the property of providing multiple detections around a present vehicle. A clustering algorithm has to be applied to join these detections. For this purpose, we follow the approach of Ponsa et al. [7], where a confidence weighted average of disjoint regions is built. Each cluster \( \vartheta_i \) of regions is characterized by:

- number of elements \( n_i \),
- sum of confidence values \( c_i \), and
- width of the weighted average region \( w_i \).

In contrast to Ponsa et al. our detection system has to find one vehicle in the ROI only. Hence, we can optimize the fusion algorithm to provide the detection that fits most to the radar information. Since our ROI is quite large, it is possible that more than one vehicle or a few vehicle parts are present and trigger a detection.

In general, the prospect vehicle has the property to raise a cluster with the most elements \( n_i \) and the maximal \( c_i \) value. Furthermore, we use the information that we search for a vehicle that has on average a width close to 2.00 meters converted to pixel value (\( w_{avg} \)). We define two vectors \( d_i = (n_i, c_i, w_i) \) and \( d_{ref} = (n_{max}, c_{max}, w_{avg}) \). The confidence value \( c_i \) for each detection can be determined with \( \| \sum_{i=1}^{L} \alpha_i h_i(x) - \frac{1}{2} \sum_{i=1}^{L} \alpha_i \| \), where \( \| \cdot \| \) denotes the Euclidean norm.

For each ROI the algorithm provides just one detection that is obtained as follows:
1. Build clusters \( \vartheta_i \) of disjoint regions;
2. Determine the confidence weighted average of each cluster;
3. Choose \( \vartheta_i \)th cluster by \( \min_i \| d_i - d_{ref} \| \).

6. RESULTS AND CONCLUSIONS

We tested our system on a data set of 1833 images, containing 343 vehicles and 1490 negative samples. One half of the sample images is used for training and testing respectively. As mentioned above, we used vehicle rear views and negative samples like road signs, trees, and guard rails (cp. figure 1). This hand labeled data was recorded in a German city.
The scenes were collected with changing cameras during different seasons and daytime. Thus, a wide range of illumination conditions is given.

For evaluation, we trained four single layered (monolithic) classifiers with 150 features. Since we only want to evaluate properties of different feature extraction methods, it is not necessary to train a cascade. For each classifier a test is performed, where the number of selected features is increased from 1 to 150 and the influence on true positive rate (TP), false positive (FP) rate and accuracy (AC) is analyzed. All classifiers are trained with a discrete AdaBoost [12] algorithm.

Table 1 shows that using contrast normalization (CN) instead of variance normalization (VN) results in slightly better detection rates. The true positive rate is increased about 1.27 percentage points and the false positive rate is lowered from 0.32 % to 0.16 %. For further improvement we examined mutual information in combination with contrast normalization. Bounded mutual information can increase the accuracy of the classifiers. In our system a strong MI threshold (e.g. MI < 0.25) causes a very low false positive rate of 0 % with a decreased true positive rate of 88.54 %. In contrast, slightly bounded MI (e.g. MI < 0.5) leads to an increased true positive rate of 93.63 %, while the false positive rate is still 0 %. The choice of an MI threshold depends on the training data and features. Examining a classifier with unbounded MI showed that more than one fourth of the 150 selected features had an MI value greater than 0.5. This could explain, why we obtain better results with a relatively high MI threshold. If we choose a lower MI threshold, it is likely that the algorithm tries to select features with low mutual information, but the weighted classification error is too high to produce good results. Thus, the more features are selected the more the difficulty rises to find features with low MI and low classification error. A solution for this problem is suggested by Shen et al. [9]. For example, an adaptive MI threshold or a cross-validation set could be used.

In figure 4 the progress of successively using more features for a classifier with unbounded MI and an MI threshold of 0.5 is shown, respectively. Since the first 23 features have MI values smaller than 0.5, the two algorithms select the same features. This effect is reflected by the accuracy. Bounded mutual information can increase the accuracy of the classifiers. In our system a strong MI threshold (e.g. MI < 0.25) causes a very low false positive rate of 0 % with a decreased true positive rate of 88.54 %. In contrast, slightly bounded MI (e.g. MI < 0.5) leads to an increased true positive rate of 93.63 %, while the false positive rate is still 0 %. The choice of an MI threshold depends on the training data and features. Examining a classifier with unbounded MI showed that more than one fourth of the 150 selected features had an MI value greater than 0.5. This could explain, why we obtain better results with a relatively high MI threshold. If we choose a lower MI threshold, it is likely that the algorithm tries to select features with low mutual information, but the weighted classification error is too high to produce good results. Thus, the more features are selected the more the difficulty rises to find features with low MI and low classification error. A solution for this problem is suggested by Shen et al. [9]. For example, an adaptive MI threshold or a cross-validation set could be used.

Table 1 shows that using contrast normalization (CN) instead of variance normalization (VN) results in slightly better detection rates. The true positive rate is increased about 1.27 percentage points and the false positive rate is lowered from 0.32 % to 0.16 %. For further improvement we examined mutual information in combination with contrast normalization. Bounded mutual information can increase the accuracy of the classifiers. In our system a strong MI threshold (e.g. MI < 0.25) causes a very low false positive rate of 0 % with a decreased true positive rate of 88.54 %. In contrast, slightly bounded MI (e.g. MI < 0.5) leads to an increased true positive rate of 93.63 %, while the false positive rate is still 0 %. The choice of an MI threshold depends on the training data and features. Examining a classifier with unbounded MI showed that more than one fourth of the 150 selected features had an MI value greater than 0.5. This could explain, why we obtain better results with a relatively high MI threshold. If we choose a lower MI threshold, it is likely that the algorithm tries to select features with low mutual information, but the weighted classification error is too high to produce good results. Thus, the more features are selected the more the difficulty rises to find features with low MI and low classification error. A solution for this problem is suggested by Shen et al. [9]. For example, an adaptive MI threshold or a cross-validation set could be used.

In figure 4 the progress of successively using more features for a classifier with unbounded MI and an MI threshold of 0.5 is shown, respectively. Since the first 23 features have MI values smaller than 0.5, the two algorithms select the same features. This effect is reflected by the accuracy. Bounded mutual information can increase the accuracy of the classifiers. In our system a strong MI threshold (e.g. MI < 0.25) causes a very low false positive rate of 0 % with a decreased true positive rate of 88.54 %. In contrast, slightly bounded MI (e.g. MI < 0.5) leads to an increased true positive rate of 93.63 %, while the false positive rate is still 0 %. The choice of an MI threshold depends on the training data and features. Examining a classifier with unbounded MI showed that more than one fourth of the 150 selected features had an MI value greater than 0.5. This could explain, why we obtain better results with a relatively high MI threshold. If we choose a lower MI threshold, it is likely that the algorithm tries to select features with low mutual information, but the weighted classification error is too high to produce good results. Thus, the more features are selected the more the difficulty rises to find features with low MI and low classification error. A solution for this problem is suggested by Shen et al. [9]. For example, an adaptive MI threshold or a cross-validation set could be used.

Table 1: Results of 150-feature classifiers.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>TP</th>
<th>FP</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>VN, unbounded MI</td>
<td>88.54%</td>
<td>0.32%</td>
<td>97.35%</td>
</tr>
<tr>
<td>CN, unbounded MI</td>
<td>89.81%</td>
<td>0.16%</td>
<td>97.81%</td>
</tr>
<tr>
<td>CN, MI &lt; 0.5</td>
<td>93.63%</td>
<td>0.00%</td>
<td>98.76%</td>
</tr>
<tr>
<td>CN, MI &lt; 0.25</td>
<td>88.54%</td>
<td>0.00%</td>
<td>97.68%</td>
</tr>
</tbody>
</table>

Nevertheless, there are some aspects that should be enhanced or included in future work:
1. Evaluation of a cascaded classifier
2. Search for an efficient optimization strategy to determine optimal mutual information threshold for training of a cascaded classifier
3. Inclusion of other features, e.g. rotated rectangle features proposed by Barczak et al. [1] or Lienhart and Maydt [6]
4. Integration of an object tracking method, e.g. the feature-based tracking developed by Leung et al. [5]

Figure 4: Development of different classifiers increasing the number of features.

Figure 5: Region of interest and object distance is passed to the classification system. After the fusion of detections the accurate object position and width can be estimated.

The scenes were collected with changing cameras during different seasons and daytime. Thus, a wide range of illumination conditions is given.

For evaluation, we trained four single layered (monolithic) classifiers with 150 features. Since we only want to evaluate properties of different feature extraction methods, it is not necessary to train a cascade. For each classifier a test is performed, where the number of selected features is increased from 1 to 150 and the influence on true positive rate (TP), false positive (FP) rate and accuracy (AC) is analyzed. All classifiers are trained with a discrete AdaBoost [12] algorithm.

Table 1 shows that using contrast normalization (CN) instead of variance normalization (VN) results in slightly better detection rates. The true positive rate is increased about 1.27 percentage points and the false positive rate is lowered from 0.32 % to 0.16 %. For further improvement we examined mutual information in combination with contrast normalization. Bounded mutual information can increase the accuracy of the classifiers. In our system a strong MI threshold (e.g. MI < 0.25) causes a very low false positive rate of 0 % with a decreased true positive rate of 88.54 %. In contrast, slightly bounded MI (e.g. MI < 0.5) leads to an increased true positive rate of 93.63 %, while the false positive rate is still 0 %. The choice of an MI threshold depends on the training data and features. Examining a classifier with unbounded MI showed that more than one fourth of the 150 selected features had an MI value greater than 0.5. This could explain, why we obtain better results with a relatively high MI threshold. If we choose a lower MI threshold, it is likely that the algorithm tries to select features with low mutual information, but the weighted classification error is too high to produce good results. Thus, the more features are selected the more the difficulty rises to find features with low MI and low classification error. A solution for this problem is suggested by Shen et al. [9]. For example, an adaptive MI threshold or a cross-validation set could be used.

In figure 4 the progress of successively using more features for a classifier with unbounded MI and an MI threshold of 0.5 is shown, respectively. Since the first 23 features have MI values smaller than 0.5, the two algorithms select the same features. This effect is reflected by the accuracy. Bounded mutual information can increase the accuracy of the classifiers. In our system a strong MI threshold (e.g. MI < 0.25) causes a very low false positive rate of 0 % with a decreased true positive rate of 88.54 %. In contrast, slightly bounded MI (e.g. MI < 0.5) leads to an increased true positive rate of 93.63 %, while the false positive rate is still 0 %. The choice of an MI threshold depends on the training data and features. Examining a classifier with unbounded MI showed that more than one fourth of the 150 selected features had an MI value greater than 0.5. This could explain, why we obtain better results with a relatively high MI threshold. If we choose a lower MI threshold, it is likely that the algorithm tries to select features with low mutual information, but the weighted classification error is too high to produce good results. Thus, the more features are selected the more the difficulty rises to find features with low MI and low classification error. A solution for this problem is suggested by Shen et al. [9]. For example, an adaptive MI threshold or a cross-validation set could be used.

Table 1: Results of 150-feature classifiers.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>TP</th>
<th>FP</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>VN, unbounded MI</td>
<td>88.54%</td>
<td>0.32%</td>
<td>97.35%</td>
</tr>
<tr>
<td>CN, unbounded MI</td>
<td>89.81%</td>
<td>0.16%</td>
<td>97.81%</td>
</tr>
<tr>
<td>CN, MI &lt; 0.5</td>
<td>93.63%</td>
<td>0.00%</td>
<td>98.76%</td>
</tr>
<tr>
<td>CN, MI &lt; 0.25</td>
<td>88.54%</td>
<td>0.00%</td>
<td>97.68%</td>
</tr>
</tbody>
</table>
REFERENCES


