Stereo-Based Pedestrian Detection for Collision-Avoidance Applications

Sergiu Nedevschi, *Member, IEEE*, Silviu Bota, and Corneliu Tomiuc

***Abstract*—Pedestrians are the most vulnerable participants in urban traffic. The first step toward protecting pedestrians is to reliably detect them. We present a new approach for standing- and walking-pedestrian detection, in urban traffic conditions, using grayscale stereo cameras mounted on board a vehicle. Our system uses pattern matching and motion for pedestrian detection. Both 2-D image intensity information and 3-D dense stereo information are used for classification. The 3-D data are used for effective pedestrian hypothesis generation, scale and depth estimation, and 2-D model selection. The scaled models are matched against the selected hypothesis using high-performance matching, based on the Chamfer distance. Kalman filtering is used to track detected pedestrians. A subsequent validation, based on the motion field's variance and periodicity of tracked walking pedestrians, is used to eliminate false positives.**

***Index Terms*—Collision avoidance, optical flow, pattern matching, pedestrian detection, stereo vision, urban traffic.**

I. Introduction

RECOGNIZING humanoid shape is very easy for humans, yet it is very difficult, at the moment, for computer vision systems. This is particularly true in highly cluttered urban environments and using moving cameras. The high variance in appearance, occlusions, and different poses and distances present difficult problems in pedestrian detection. The aim of our work was the development of a real-time pedestrian-detection algorithm, exploiting 2-D and 3-D information that is capable of detecting pedestrians in urban scenarios. Our system is designed to work as a precrash sensor on board road vehicles. The sensor will provide information for driver-warning systems and actuators.

The architecture of our classification system is presented in Fig. 1. Our pedestrian detector uses two grayscale cameras arranged in a stereo configuration. Our cameras supply 512×383 images. The baseline of our stereo system is 320 mm. The focal length is 380 pixels, giving a field of view of 68°. More information about the system can be found in [1].

In all subsequent equations, we will refer to the following reference frames: 1) the image reference frame, having the origin in the top-left corner, the Oy axis pointing in the up–down direction, and the Ox axis pointing from left to right; 2) the left camera's reference frame, having the origin in the optical center of the left camera, the Ox axis pointing from left to right, the Oy axis pointing along the up–down direction, and the Oz axis pointing away from the camera; and 3) the ego vehicle's reference frame, which is similar to the left camera's reference frame but has the origin on the ground, in the middle of the front end of the ego vehicle.

A hardware stereo-reconstruction system named "TYZX" [2] is used to generate a range image (from a disparity map) by stereo matching the two intensity images. From the range image, a set of reconstructed 3-D points is generated. An original point-grouping algorithm based on "density maps" is applied on the set of reconstructed 3-D points to generate pedestrian hypotheses. A model type and a scaling factor are determined for each hypothesis, based on the 3-D information associated with it. By projecting the 3-D hypothesis in the left camera's image plane, a 2-D candidate window is generated. A set of 2-D edges is extracted from each 2-D candidate window, and the edges are filtered based on their associated depth information. A "distance transform" is performed on the edges. The set of edge features associated with the selected model type at the selected scale is then elastically matched to the distance-transformed edge image. Objects that do not have a high-enough matching score are filtered out. The remaining objects are strong pedestrian hypotheses. A Kalman-filter-based tracking algorithm is used to track these remaining strong pedestrian hypotheses. If the tracked pedestrians are determined to be moving (walking), a subsequent test based on motion is used to reject false positives. The motion test is based on detecting the variance of the 3-D motion field vectors associated with the body parts of the presumed pedestrians. We call this variance a "motion signature." Pedestrians are expected to display a large motion signature, which is caused by the different directions in which body parts move during walking. Another powerful test we use is to determine if the motion signature is a periodic function of time. The periodicity is caused by the swinging of the pedestrian's arms and legs while walking.

A. Related Work

There are various methods for object detection and pedestrian hypothesis generation. Some are based on 2-D information, for example, the detection of image regions where there are a significant number of vertical edges [3]. Other methods are based on some type of additional information like infrared (IR) images [3] or depth information [3], [4]. Most pedestrian detection methods, which use depth information, rely on the disparity map and make some kind of segmentation on this map to detect objects [4] or use a v-disparity approach [3]. However,

Manuscript received December 14, 2007; revised May 6, 2008 and November 4, 2008. First published February 2, 2009; current version published September 1, 2009. The Associate Editor for this paper was U. Nunes.

The authors are with the Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, 400020 Cluj-Napoca, Romania (e-mail: Sergiu.Nedevschi@cs.utcluj.ro; Silviu.Bota@cs.utcluj.ro; Corneliu.Tomiuc@cs.utcluj.ro).

Digital Object Identifier 10.1109/TITS.2008.2012373

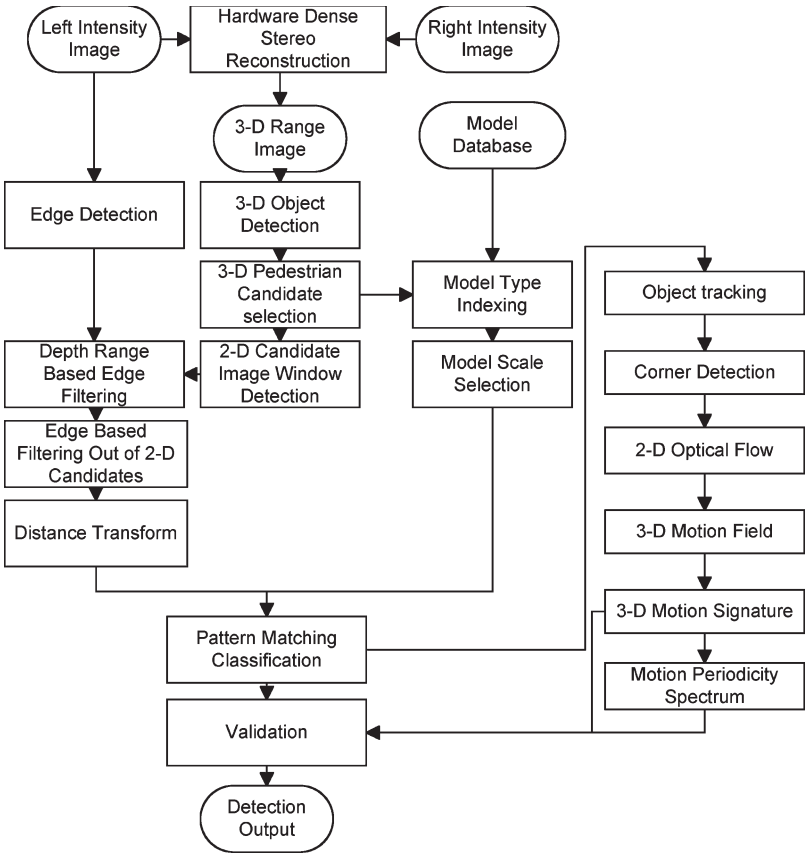

Fig. 1. System architecture.

although the approaches based on disparity maps are faster compared with approaches based on full 3-D information, these approaches heavily rely on a dense and error-free disparity map, which is hard to obtain in real-life scenarios. Some methods to reduce errors are cumulation using v-disparity or the generalized Hough transform paradigm. A similar approach to our own, which uses a criterion based on reconstructed 3-D points density, is described in [5]. The authors divide the ground plane into cells and compare the number of reconstructed points with the expected number. However, this approach may be unable to accurately detect such small objects as pedestrians.

Object detection and object classification based on pattern matching are traditionally limited to 2-D image intensity information [6]. Sometimes, techniques such as Adaboost with illumination-independent features are used [7]. The advantage of using the 2-D information consists of the fact that all the information has a high degree of accuracy and a high level of trust since the image represents an accurate projection of the real scene (without taking into account the image noise, which is not a significant factor for high-quality video cameras; of course, a production system would use lower quality cameras and should apply some filtering to remove noise before processing). The disadvantage of using only 2-D image information is that we do not have any additional spatial information about where the objects are and what their size is. A detection system based on pattern matching using only the intensity image will usually try to fit all the models using all the positions and scales that are plausible to find a match in the image. This generates an extremely large search space, which cannot be reduced because the 3-D information is missing. Pattern-matching approaches based on methods similar to the distance transform allow a limited degree of difference between the model and the features of the matched object but still require a large number of scales and positions for each model.

The 3-D information generated by a stereo reconstruction system provides depth for objects that make up the scene. There are some classification methods that directly use this information [8], [9]. The classification solely based on 3-D information is difficult as the 3-D reconstruction systems do not provide sufficiently dense and reliable 3-D data to allow the reliable extraction of 3-D shapes and surfaces (however, for a promising approach in surface extraction, see [10]).

The use of pattern matching in conjunction with 3-D information has not been extensively explored, mainly because real-time dense stereo reconstruction systems have not been available until recently. Some approaches aimed at pedestrian detection have used dense 3-D information but only as a validation method [11]. The 3-D data generated by these real-time dense stereo reconstruction devices are still noisy and have a much lower degree of confidence than intensity data. However, by careful calibration of the stereo rig and careful filtering and usage of the 3-D data, it is now possible to extract quality results from the 3-D data.

Another important feature for walking-pedestrian detection is their walking pattern. There are a number of works, e.g., [12] and [13], that have used motion cues for pedestrian detection.

A typical approach for motion-based pedestrian detection is to determine if the motion associated with the presumed pedestrian is periodic. However, in cluttered environments, it is usually very hard to distinguish object motion from background motion. Furthermore, due to high car speed, close range, and a low frame rate, it is possible that objects will move many pixels from one frame to the next, thus making local methods for motion detection infeasible. Also in motion-based pedestrian detection, 3-D features are not extensively explored. The main advantage of a 3-D approach in motion analysis is the possibility of correctly segmenting the foreground and the background in complex scenarios and with moving cameras. In addition, having 3-D information means that the true scale of the motion can be recovered.

B. System Characteristics

According to [14], pedestrian collision-avoidance systems are classified by field of view, angular resolution, detection range, range resolution, illumination type, hardware cost, and algorithmic complexity. Our system has a wide field of view of 68°. The angular resolution is medium, at about 8 of arc. The detection range is medium, at 20 m. Above this range, because of disparity errors, it becomes very difficult to detect such small objects as pedestrians. The range resolution is high, due to the use of 3-D information (the depth information is computed from stereo rather than inferred from mono). Expected errors are about $4\%Z$. We use no active illumination techniques, which is an advantage. The hardware cost is medium as normal cameras are used, but a hardware stereo vision machine [2] is used for depth information extraction. The algorithmic complexity is medium to low, as using 3-D information speeds up the matching process. Motion signature computation too is performed only when necessary and only on corner points belonging to presumed pedestrians. Our system works under all urban traffic conditions, illumination conditions permitting.

C. Contributions

The novelty of our system consists particularly in the powerful combination of 2-D intensity information, 3-D depth information, and motion features. This combination uses all the possible information provided by our stereo sensor. Furthermore, no other sensors such as a gyroscope, radar, and laser scanner, no active illumination, and no IR cameras are used.

Object detection using density maps is a novel and resilient object-detection algorithm.

The pattern-matching algorithm rejects all edges that do not have a correct depth. A similar approach is used for optical flow computation, where all corner points that do not have correct depth are eliminated. In addition, the correct scale of the model used for matching is directly inferred using the 3-D information available.

The motion-based approach also has many novel aspects. The motion field computation is done in 3-D and not in 2-D and is thus able to detect pedestrians walking in any direction. We also introduce a new feature called the "motion signature" feature.

D. Paper Structure

The next section presents the object-detection algorithm using density maps. Section III describes the pattern-matching-based classification. Sections IV and V describe the motion signature and motion periodicity validation tests. In Section VI, we describe the fusion of the pattern-matching-based classification with the motion-based classification. In Section VII, we show some experimental results obtained using our classifier, and in Section VIII, we draw conclusions and present possible future work.

II. Object Detection Using Density Map

We propose a method for object detection and pedestrian hypothesis generation based on full 3-D information (array of 3-D points with x-, y-, and z-coordinates computed based on the disparity map and the parameters of the calibrated stereo-acquisition system).

Our method for object detection relies on the fact that objects have a higher concentration of reconstructed 3-D points than the road surface (see Fig. 2). Furthermore, vertical structures are usually very well reconstructed by the 3-D reconstruction engines when the stereo camera system has a horizontal baseline (our case also). This results in a high density of 3-D points in the 3-D space occupied by objects in general and pedestrians in particular. By trying to determine those positions in space where there is a high density of 3-D points, possible positions for objects can be determined.

The characteristics of a collision avoidance safety system require a detection area of 20 m (longitudinal) and 10 m (lateral). Our camera system is designed to give the best results in these range. Therefore, we select, for the purpose of object detection, a box-shaped subspace of the scene, having a 20-m length, a 10-m width, and a 2-m height. The height restriction is necessary to avoid spurious detection of tree foliage, suspended objects, etc.

To cope with projection errors caused by errors in the cameras' extrinsic parameters, we use a very precise calibration procedure, which produces exact results up to a range of 35 m [15]. The precise calibration procedure allows stereo reconstruction to be performed in principle up to the specified distance without significant outliers caused by incorrect image rectification or undistortion. The pitch and roll of the ego vehicle are estimated online [16]. However, some errors are still present, caused by imperfect stereo matching. These errors are filtered out by the accumulation (averaging) effect of the density map and do not significantly alter the performance of our system in the detection area given above. There is also an increase in reconstruction error with distance. The stereo reconstruction, which is performed by the TYZX hardware, computes disparities with subpixel accuracy, and because of this, the errors are not too serious for the considered range.

A density map is constructed from the 3-D points located within this limited subspace. The 3-D points are projected on the xOz (road) plane. In this limited area that we have considered for detection, the road can be considered flat; therefore, it is no problem that a plane is used.

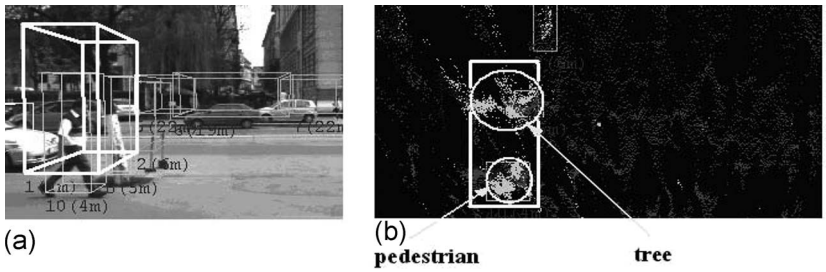

Fig. 2. (a) Grayscale image and objects. (b) Three-dimensional points projected on the xOz plane (the big box represents the result of a different object detection algorithm [1].

The density map is an accumulation buffer. Each projected point adds a value to the accumulation buffer. A cell in the accumulation buffer covers an area of 50 mm $\times$ 50 mm on the ground plane. The weights that the point adds to the density map have a Gaussian form, with the maximum at the center cell and decreasing in the neighboring cells. Because points become sparser as we move away from the camera, the size of the patch increases with the distance. The diameter of the patch is one cell (50 mm) at the nearest range and increases up to six cells (300 mm) at the far range (20 m). The amount by which the patch increases was empirically determined by testing the system in various scenarios. A better approach would be to consider a probabilistic error model for stereo reconstruction and to compute the required patch size from it. Furthermore, at the far end of the detection range, there exists the risk of multiple persons being grouped together, because of the large patch size.

Because the influence of the 3-D points on the density map is cumulative, the density map will contain large values in areas with a high density of 3-D points.

Segmentation is performed on the density map, using a region-growing algorithm to determine possible object candidates. The region-growing threshold is based on the total amount of 3-D reconstructed points. This allows the segmentation to adapt to the situations where the reconstruction is less than perfect.

The result of the segmentation is a list of object hypotheses on the density map. Three-dimensional boxes are generated based on the position and size of the object hypotheses in the density map. Once the list of potential candidates is determined, the pattern-matching algorithm is applied, as described in Section III.

The detector's resilience to occlusions mainly depends on the amount of occluded area. Usually, if at least 50% of the pedestrian is visible, then the density map approach is able to detect it. Of course, it is rather difficult, by any imaginable approach, to completely detect occluded pedestrians. The tracking module is able to propagate the pedestrian hypotheses for a few frames if they become temporarily occluded.

In the area considered for detection, we found that we are also able to detect small children.

Usually, our detector is able to segment individual pedestrians from pedestrian groups. If properly segmented, the pedestrians will be individually detected, using shape pattern matching and motion features. If pedestrians are multiply grouped, then the pattern matching will usually fail, but the motion features may be able to supply enough information for correct classification.

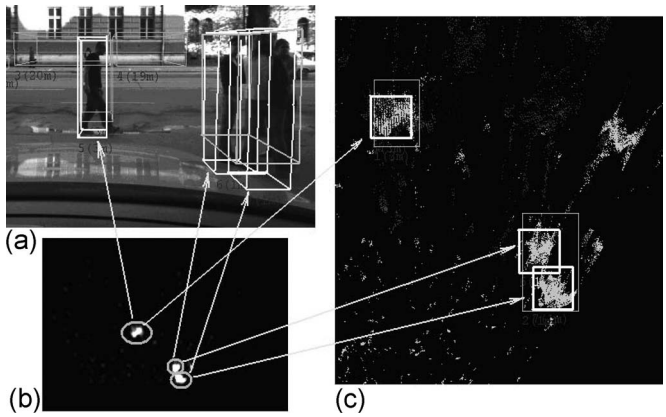

Fig. 3. Object detection using density map. (a) Grayscale image. (b) Density map. (c) Three-dimensional points projected on the xOz plane.

Fig. 3 shows the results of density-map-based object detection. The top-right cluster is caused by points that are outside the considered region. Because of this, they are not considered when the density map is built. In this particular scenario, these points were caused by bad reconstruction (they "slipped" to the far range). The large dark-gray boxes are the result of a different object detection algorithm that is suitable for large objects such as cars [1].

III. Pattern-Matching-Based Classification

This section presents the shape-based pedestrian–nonpedestrian classification. The first step is to select a number of candidates from the detected objects (see the previous section). Next, the edge features used for pattern matching are extracted from the left 2-D image and filtered based on their 3-D positions to be coherent with the position of the object box. Models are selected based on the aspect of the 3-D box containing the object, and their scale is determined. A distance transform is applied on the edge-feature image. Finally, the selected model is matched to the feature image. The following sections will describe each of these steps in detail.

A. 3-D Candidate Selection

A fast classification algorithm is applied on the 3-D boxes that contain the objects. Candidate objects are classified based on 3-D size and position. This type of classification only

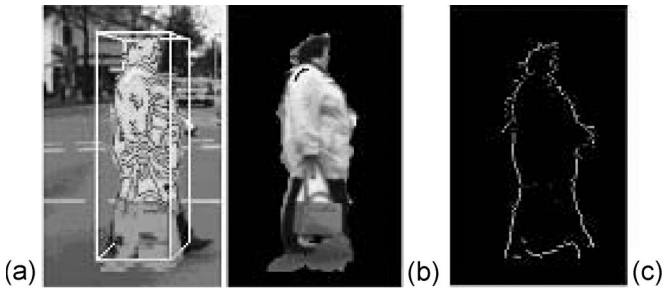

Fig. 4. (a) Original image (reconstructed object points are colored). (b) Object separation. (c) Feature selection

provides rough information and is only able to distinguish between classes of objects that are very different in size, like trucks, cars, and pedestrians. The size features for 3-D boxes are their height and "base radius." The height is obtained by analyzing the 3-D point distribution along the vertical axis it will be discarded. We have determined the right attribute values using Bayesian learning, and the acceptance or rejection is made by a naive Bayesian classifier. We found that the acceptable height is between 0.9 and 2.2 m, and the base radius is between 0.25 and 1 m. Of course, each height and base radius value has its own likelihood of being a valid hypothesis. With these limits, groups with up to three persons and small children are accepted. The position refers to the position of the hypothesis as related to the current driving lane. If the driving lane is detected and if the object's position is not on or near the current lane, then we discard it (this validation is optional).

The candidates are also filtered based on 3-D information, taking into account that the objects need to have a good distribution of reconstructed 3-D points to be good candidates. If not enough 3-D information is present, the classification will have a low degree of certainty, and the result will not be very useful. As a result, if an object does not have a continuous distribution of 3-D points along the vertical axis, it will be eliminated. This filter will eliminate false positives that are small objects but have a high enough density of reconstructed 3-D points to surpass the density threshold.

B. 2-D Candidate Processing

Once a 3-D hypothesis has been determined to be a good candidate for classification, a 2-D hypothesis is generated by projecting the 3-D object on the left image plane. The 2-D hypothesis is a window containing the hypothetical pedestrian and is described by its bordering edges. In Fig. 4(a), the window of the 2-D hypothesis is shown. In Fig. 4(b), only reconstructed points that are located within the hypothesis' 3-D box are shown. In Fig. 4(c), the edges belonging to the object's outline are shown.

The set of edge features that will be used for pattern matching are extracted from this window using the Canny edge detector algorithm [17]. A clear separation of the edges belonging to the object against the other background edges is required to increase the accuracy of the pattern matching. For that, a depth coherency constraint is applied, exploiting the available 3-D information. The edges in the selected image window have associated depth information, and only edges situated in the volume of space determined by the 3-D box of the object hypothesis are used, as opposed to all the edges in that region.

The edges we use for classification are occlusion edges, caused by the pedestrian's outline covering the background. These edges cause problems for some stereo-reconstruction algorithms. This is because they represent discontinuities in the intensity image and may be viewed from slightly different angles by the left and right cameras. However, this is not the case with the stereo vision machine we use. It usually pulls the reconstructed points to the foreground and thus generates correct depth for foreground edges. Furthermore, because of the short baseline, the cameras view the same object from approximately the same angle, and thus, there are no large errors around reconstructed edges.

Subsequently, the 2-D hypotheses are filtered out based on the absence of significant (long) edge features in the image window. This is because objects lacking significant edges and, thus, implicitly significant corner points would be useless for further shape- and motion-based classification steps. From our test scenarios, we observed that we never eliminated a true pedestrian by this filter. Usually, the false positives eliminated here are objects with a high density of 3-D points, vertically distributed and belonging to small elements, e.g., foliage or other strongly textured structures.

C. Pattern Matching

In this step, we match a set of human shape (outline) models with the edges present in the 2-D image window. A shape model is a set of points forming a pedestrian outline. To reduce the computational time and to optimize the classification process, the set of templates is grouped into a hierarchical structure, as presented in [18]. The hierarchy is constructed by determining the similarity between each pair of models. The initial set of models is segmented into a number of groups, based on similarity. A prototype model (the model that is most similar to all other models in its group) is selected for each group. The process is then repeated at the next level, using the prototypes from the previous level, until a single model remains. This model has the highest similarity compared to all other models (it is the most general), and it will be situated at the root of the template tree. The children of each node in the template tree are the nodes from among which it was selected as a prototype. Therefore, the templates become more particular along each path going from the root of the tree to a leaf. The similarity matrix and the structure of the tree are determined offline.

The 3-D information related to the object provides the exact distance to the object and its dimensions. Using the projection equations, we can determine the size of the object in the 2-D image. As a result, the height information is used to determine the model type, and the depth information is used to determine the scaling factor for the models used in the pattern matching.

To perform the pattern matching, a distance transform is applied on the hypothesized image window (see Fig. 5). The distance transform is applied on the edge image rather than on the model image because the model has less information and can be reduced to a vector of image positions. This vector will only contain those positions where information is present in

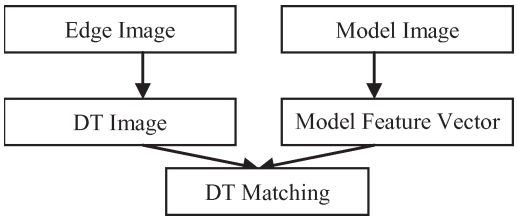

Fig. 5. Distance transform matching.

the model image. This parameterization of the model provides a significant speedup in the matching stage. The result of applying a distance transformation [19] to an edge image is a bidimensional map with the same size as the image, and each element of this map has a value proportional with the distance from the corresponding image point to the closest edge point.

The distance between the model and the distance-transformed image is then computed, and the model receives a score based on this distance. Two main distance metrics can be used to measure the similarity between two sets of points: the average distance (or Chamfer distance [20]) and the average truncated distance (or Hausdorff distance [21], [22]). The Hausdorff distance is more robust (can deal with partial object occlusions) than the Chamfer distance, but it has an increased computational complexity. In our system, we chose to use the average distance (Chamfer distance) for matching since the system needs to fulfill real-time processing constraints. We found that the errors generated by the use of the Chamfer distance are usually small.

To reduce the number of models and levels in the classification tree, we used models that represent only the torso and head of the pedestrians since these features seem to be the ones that change the least in the process of walking. The head and torso of the hypotheses are considered to lie within its upper body part. This approach induces some false positives but also decreases the amount of false negatives, improving the overall correct detection rate.

An alternative approach when matching would have been to use scale-invariant descriptors such as shape context. However, these methods have a much larger computational complexity and do not take advantage of the fact that we already know the correct scale (by stereo).

IV. Motion-Based Validation for Walking Pedestrians

In this section, we present the motion-validation method used to eliminate false positives among walking pedestrians. This step is applied only to moving hypotheses. If a hypothesis is not moving, it will be classified based on shape only.

A. Hypotheses Tracking

The first step for the motion-based validation is to track pedestrian hypotheses across multiple frames. The purpose of tracking is twofold. First, it is used to determine if the pedestrian hypothesis is moving relative to the fixed scene (the reference frame attached to the road). Second, the tracking supplies associations between objects in the previous frame and the current frame, which are important for the computation of motion-based features.

Our multiple-tracking frame consists of the following steps. First, the ego vehicle motion is estimated, using a yaw rate sensor (the serial production sensor incorporated in the ESP system), a speed sensor (again, the serial production sensor), and knowledge of the time stamps of the previous and current frames. As the yaw rate sensor is noisy, we track its output using a Kalman filter for more stable results.

We are thus able to obtain a rotation matrix and a translation vector, which together describe the way in which the ego vehicle has moved relative to the fixed reference frame, from the time of the previous frame to the time of the current frame. Alternatively, the translation vector and the rotation matrix describe how objects (considered fixed relative to the fixed scene) are moving relative to the ego vehicle.

Second, we consider that each pedestrian hypothesis moves in a straight line at constant speed. We are thus able to predict their position relative to the ego vehicle. Of course, pedestrians do not always move in straight lines, but this is a good first-order approximation.

The association phase is the most difficult and sensitive step. To make the association more resilient, we used image-based validation for matching targets from the previous frame with targets in the current frame. The images used for matching are depth masked (i.e., only the points for which 3-D information is available and which, according to this 3-D information, belong to the object's foreground are used). Furthermore, because of the 3-D information available, we are able to scale the images (to compensate for the zoom-in effect caused by the ego vehicle motion). Image validation is performed on depth-masked scaled-down object images, using the sum of absolute differences as a similarity measurement.

Unfortunately, most association errors occur when object trajectories cross. In this case, one object usually occludes the other, and we are left with very few unoccluded pixels to perform image-based validation. Furthermore, pedestrians tend to be similar in appearance. As a further improvement, we used the method of bipartite graph matching, which is described in [23], to find globally optimal associations.

If, according to the tracking information, the object seems to be *moving relative to the fixed scene*, we consider it a walking-pedestrian hypothesis and validate it using motion.

A motion signature, based on the 3-D motion field associated with the presumed pedestrian is computed, and objects with a low value for this motion signature are discarded. Finally, we test to see if the motion signature is periodic across multiple frames and discard objects with nonperiodic motion signature. The following sections present each step in detail.

B. Depth Masking and 2-D Optical Flow Computation

In urban environment scenarios, there are a number of difficulties associated with optical flow extraction.

1) Frame Rate: The frame rate is relatively low, as compared with the velocities of the objects in the scene. The average frame rate in the sequences we used was 20 ft/s, which means that an object that is sufficiently close by could move by many

pixels from one frame to the next. For example, a pedestrian situated at 2 m in front of the camera moving with a speed of 2 m/s on a direction parallel to the camera would generate an image motion of

$$\Delta x = \frac{f_x V_X}{Z} \Delta t \tag{1}$$

which equals 19 pixels in our $f = 380$-pixel camera. A speed of 2 m/s is equivalent to 7.2 km/h, which is not very large for a running pedestrian. To cope with the relatively low frame rate, we use the output from the tracker to estimate the global motion of the object. We extract the object's image from the previous frame based on its previous location and the object's image in the current frame based on the location predicted by the tracker. The size of the extracted image is given by the minimum rectangle (bounding box) that encloses the projected 3-D cuboid associated with each object. To simplify the optical flow computation, we equalize the sizes of the previous and current object image, considering the largest size.

2) Occlusions: Objects passing in front of each other cause occlusions. The problem of occlusions must be addressed, because if we ignore it, a moving object passing in front of a stationary one will cause spurious optical flow components associated with the object in the background. To eliminate from objects' images the pixels that are not associated with true object parts, we only consider a "slice" of the image. We compute the minimum and maximum depth values of the cuboid associated with the considered object, as expressed in the camera's coordinate system. We then filter out the points for which the depth estimate computed by the TYZX system lies outside the minimum and maximum depth interval. *Having dense stereo information is crucial for this step, as a sparse set of points would not capture the true extent of the object's shape*. Even with a dense set of stereo-reconstructed points, the masked image sometimes contains "holes," particularly if the pedestrian's clothing texture is uniform. This does not pose big problems, because in areas with uniform texture, we would not be able to extract the optical flow anyway. Another problem is that some parts of the pedestrian's feet are linked with the road. This too does not seem to influence the result of the optical flow computation.

3) Optical Flow Variability: We tried various methods for computing the optical flow, based on brightness constancy constraints such as those described in [24]–[26] and block matching. We also tried methods based on both brightness and depth, as described in [27]. Unfortunately, when the ego vehicle is nonstationary, the radial optical flow field components generated by the motion of the ego vehicle greatly vary from the center of the image to its edges. Furthermore, because of imperfect tracking, global object motion cannot be totally eliminated. The moving parts of the human body also generate a large optical flow variance when imaged from a close range. However, it is imperative to compute a correct 2-D optical flow field, as this motion-based detection scheme solely relies on it.

The methods described in [24] and [25] do not yield good results because they are unable to estimate sufficiently large motion vectors that are caused by a low frame rate and large motions (ego vehicle and other objects). The method described in [27] does not seem to increase the precision of the optical flow computation, because the range data used to form the linear depth constancy equation are too smooth (lacking corners or edges) to be useful.

Consequently, only two methods for optical flow computation are useful for our environment: block matching and the pyramidal approach described in [26]. Block matching gives good results but is prohibitively computationally expensive. It is also unable to generate a sufficient number of optical flow vectors, because it uses fixed-size large blocks. Therefore, we used the pyramidal approach described in [26]. This approach has the advantage that it works across a large range of displacements. It also computes the optical flow only where it can be exactly recovered, i.e., at image corner points. The number of corner points is relatively small. The fact that we track the global motion of each object further increases the working range of this optical flow extraction method, as it only needs to detect local motion displacements. Because we perform corner detection only on the masked object images and only corner points are tracked, optical flow computation does not have a very high computational expense.

To summarize, the steps we perform for optical flow extraction are (see Fig. 6):

1) tracked object's image extraction;
2) depth mask computation, which eliminates wrong points from the object's image (i.e., points that belong to the background because of their associated depth);
3) pyramidal 2-D optical flow computation:
   a) corner detection (based on eigenvalues);
   b) Gaussian pyramid generation;
   c) pyramidal optical flow computation;
4) elimination of optical flow vectors that have ends that do not fall onto points with correct depth.

C. 3-D Motion Field Computation

In this section, we discuss the computation of the true 3-D motion of objects in the scene, based on the 2-D optical flow and the range image. As explained in the previous section, we only compute 2-D optical flow vectors that start and end in points for which the hardware TYZX system is able to supply the range value. Let $p_1(x_1, y_2)$ denote the start of the optical flow vector $\vec{v}$ in the previous frame and $p_2(x_2, y_2)$ denote the end of the optical flow vector (in the current frame). In addition, let z_1 and z_2 be the depth values supplied by the TYZX system at points p_1 and p_2, respectively. Then, the 3-D relative motion vector (expressed in the left camera's coordinate system) associated with the 2-D optical flow vector $\vec{v}$ is

$$\vec{V} = \begin{pmatrix} \frac{(x_2 - x_1) z_1}{f_x} \\ \frac{(y_2 - y_1) z_1}{f_y} \\ z_2 - z_1 \end{pmatrix} \tag{2}$$

where f_x and f_y are the focal lengths of the left camera, which is expressed in horizontal and vertical pixel units, respectively.

Because the camera's position is arbitrary, we would like to express the 3-D motion vector into the more suitable reference

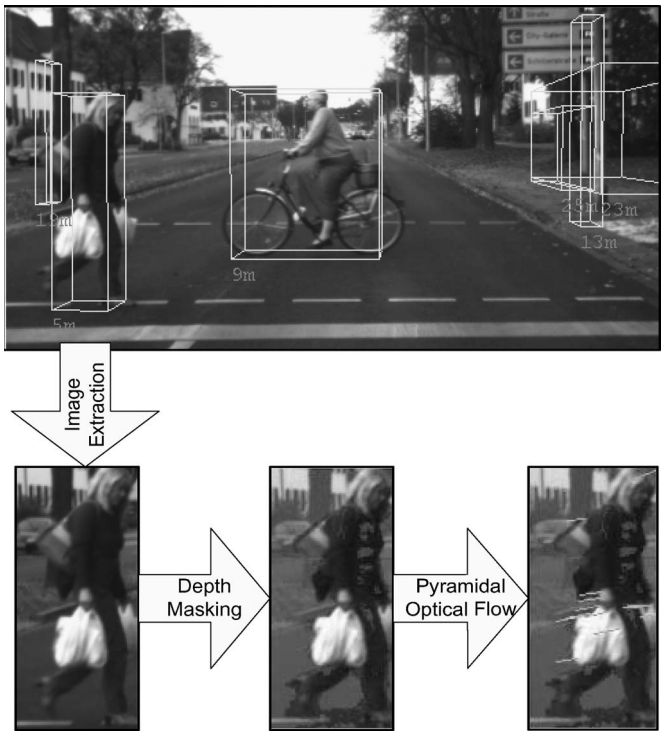

Fig. 6. Image extraction, depth masking, and optical flow computation. The masked pixels are grayed. Optical flow vectors are represented using lines.

frame (one attached to the vehicle). We have

$$\vec{V}^w = R^T(\vec{V} - \vec{T}) \tag{3}$$

where $\vec{T}$ and R are the translation vector and the rotation matrix from the vehicle's reference frame to the left camera's coordinate system, as determined by the calibration process.

After performing these transformations, we end up with a set of $\vec{V}^w$ vectors for each tracked object. The first step is the elimination of objects that lack a sufficiently high number of motion vectors. Our experiments determined that objects for which there are less than five motion vectors are very unlikely to be walking pedestrians. These objects are mainly poles or other stationary objects. Therefore, in the next steps, we will only consider objects for which more than five motion vectors have been computed.

The 3-D motion vectors cannot, by themselves, serve as a discriminating feature between pedestrians and other objects. Because of imperfect tracking, objects tend to still have a global motion, even after the global displacement predicted by tracking is eliminated. We solve this problem by subtracting the average motion

$$\mu_V = \frac{\sum_{i=1}^{n} V_i^W}{n} \tag{4}$$

from each motion vector. Another problem is caused by objects such as vertical poles. Because they lack horizontal edges (high-frequency components along the vertical direction), such objects may present spurious vertical motion components. We tried various approaches to eliminate such spurious motion vectors:

1) considering the ratio of horizontal/vertical motion components;
2) considering the average angle between the horizontal plane and the motion vector;
3) considering only the horizontal components of the motion vectors.

Although all the above approaches yield better results (higher discriminating power) than considering only the magnitudes of the motion vectors, they are all rather sensitive to noise. A much better and stable approach, i.e., the principal component analysis, is described in the next section.

D. Principal Components Analysis and Thresholding

While experimenting with the different features extracted from the 3-D motion field presented in the previous section, we observed that both the magnitudes and the orientations of the motion field vectors are important features for our walking-pedestrian detector. We would like to find the main direction along which most motion takes place. Furthermore, we are not interested in the motion itself but rather in its variability. For example, while walking, a foot moves forward, while the other moves backward (relative to the global body motion); in addition, the arms tend to have the same motion pattern (if not carrying large bags). A measure of the motion variance can be obtained by performing principal component analysis. The covariance of the motion vectors is

$$C = \frac{1}{n}\sum_{i=1}^{n}(V_i^w - \mu_V)(V_i^w - \mu_V)^T. \tag{5}$$

The 3×3 matrix C represents a covariance matrix. Let $\lambda_{\max}$ be the largest eigenvalue of matrix C. The eigenvector $\vec{V}_{\max}$ associated with $\lambda_{\max}$ represents the direction of the principal variance of the vector field $\vec{V}$. The standard deviation along the direction $\vec{V}_{\max}$ is $\sqrt{\lambda_{\max}}$. Pedestrians move mainly in the horizontal xOz plane. Therefore, we eliminate the vertical (y) motion components and consider only the projection of $\vec{V}_{\max}$ on the xOz plane. We call the fraction of $\lambda_{\max}$ corresponding to this projection as λ_{xz}. As the experimental results will show, the value of the new standard deviation $\sigma = \sqrt{\lambda_{xz}}$ is a good feature for discriminating walking pedestrians from other objects. Moreover, because of the fact that it does not consider vertical motion, it eliminates some errors caused by the absence of horizontal features on objects such as poles, which sometimes generate spurious vertical motion vectors.

The next step is to determine how to use the values of λ_{xz}^t to determine if an object is a walking pedestrian or something else. It is obvious that the values of λ_{xz}^t will be larger for walking pedestrians than for rigid objects. We manually labeled 400 images of both pedestrian and nonpedestrian objects and found that the best threshold is $\lambda_{xz}^t = 5(\mathrm{m}^2/\mathrm{s}^2)$. The procedure used for determining the threshold was an expectation–maximization algorithm.

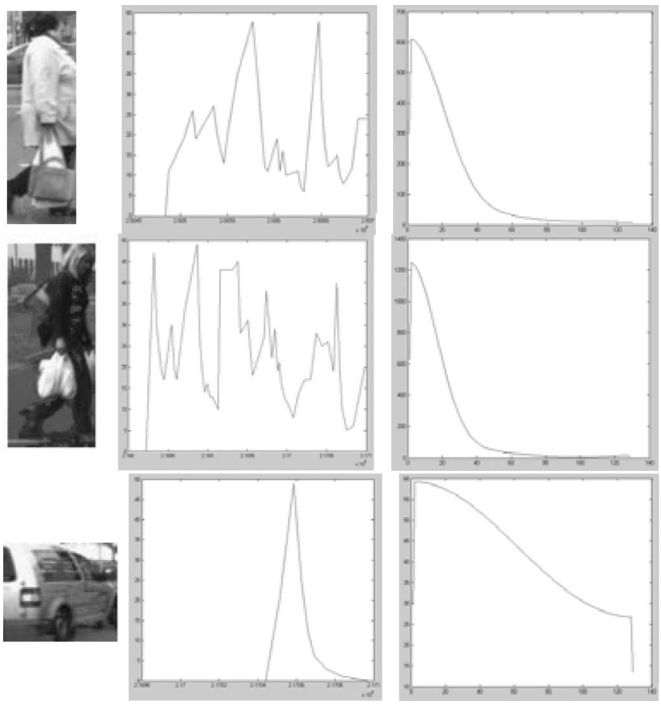

Fig. 7. (Left) Objects. (Middle) Variation with time of the motion signature. (Right) Motion signature frequency spectrum.

V. Motion Signature Periodicity

A problem with the motion signature feature is that occasionally, because of errors in stereo reconstruction and optical flow computation, a nonpedestrian object displays a high motion signature. However, these errors usually last for one or two frames.

Because we track pedestrian hypotheses, we can record and analyze their motion signature history. We have determined that for pedestrians, the motion signature variation with time is approximately periodic. Although nonpedestrian objects occasionally display some spike noise in their motion signature, this is caused by errors and is not periodic.

To characterize the periodic versus nonperiodic nature of the motion signature generated by an object, we use Welch's averaged modified periodogram method of spectral estimation [28] to compute the frequency spectrum. Frequency spectra corresponding to periodic pedestrian motion are typically band limited, while for other types of objects, they are not. This is because noise occurs with equal probability in all the spectra, while the periodic motion caused by swinging of pedestrians' arms and legs during walking occurs at some fixed frequencies.

Fig. 7 shows some examples of the motion signature variation and their spectra for two pedestrians and one car. In the middle column, the motion signature magnitude is plotted against time. In the left column, the frequency spectrum of the motion signature is shown. For the two pedestrians, this spectrum is limited to a few low frequencies. For the car, the spectrum of the motion signature is extended to high frequencies.

The classifier analyzes the frequency spectrum of motion signature to identify the periodical motion of the pedestrian's arms and legs. The cutoff frequency is used as a feature.

A major disadvantage of the motion periodicity feature is the fact that it can only be computed after the pedestrian hypothesis has been tracked for a number of frames (we used 1 s in our system). This is because we must capture the periodic motion caused by pedestrians' walking to analyze the frequency spectrum.

Another disadvantage of both motion signature and motion periodicity is that their accuracy decreases if the predominant motion direction approaches the line of sight, because it relies more and more on noisy 3-D data. The results are optimal if the predominant direction of motion is perpendicular to the line of sight.

VI. Classification Result Fusion

In this section, we describe the process by which we fuse the results of the pattern matching with those of motion signature and motion periodicity. The aim of this fusion is to reduce the number of false positives as much as possible while not rejecting correct positives.

The first observation is that while the pattern matching always gives a classification score, this is not the case for motion-based features. The first requirement for the motion-based features is that the object must be tracked. This means that at least one-frame delay is incurred from the initial hypothesis generation to the first motion signature result. The second requirement is that the object must be nonstationary (relative to the fixed scene). If the object is stationary, we do not expect any motion signature. The third requirement, this time for the motion periodicity, is that the object be tracked for a sufficient number of frames.

Therefore, we use the following scheme for fusion.

1) The pattern-matching score is computed on all pedestrian hypotheses.
2) If an object is not tracked, then only the pattern-matching score is considered, albeit with a reduced weight.
3) If an object is tracked and is moving relative to the fixed scene, then its motion signature is computed. The motion signature is then normalized with regard to the object's speed. This is because we expect a larger magnitude for the motion signature as the pedestrian's speed increases.
4) Finally, if the pedestrian hypothesis was tracked for a sufficient number of frames and the motion periodicity feature can be computed, then it is taken into account and has a major influence on the final result.
5) Another component of the fusion is the propagation of the object's class by tracking. If an object was classified as a pedestrian in the previous frame, the classification result will increase the probability that the object will also be classified as a pedestrian in the current frame and *vice versa*, toward the pedestrian in the current frame.

Let $\mathcal{A}$ be the pattern-matching score (Chamfer distance between model and hypothesis edges), $\mathcal{B} = (\lambda_{xz}/v_{xz})$ be the motion signature score normalized with speed, and $\mathcal{C} = f_{\text{cutoff}}$ be the motion periodicity score. In addition, let $\mathcal{S}_{t-1}$ be the previous score (if the object was tracked). $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are always positive values. For each of the scores, a threshold must be determined. The thresholds were determined in a

similar manner for each feature by minimizing the classification error probability on a manually labeled training set. Let N be the number of hypotheses detected in the training set, N_p be the number pedestrian among these hypothesis, and N_n be the number of nonpedestrians. Of course, $N = N_p + N_n$. Let also $\mathcal{P}$ be the subset of pedestrians in the training set and $\mathcal{N}$ be the subset of nonpedestrians in the training set. For the given threshold T and for a given feature extractor $\mathcal{X}$ applied to hypothesis H, we consider the object classified as pedestrian if $\mathcal{X}(H) > T$ and as nonpedestrian otherwise. Let N_{fp} be the number of misclassified nonpedestrians (false positives) and N_{fn} be the number of misclassified pedestrians (false negatives). We have

$$N_{\mathrm{fp}} = \sum H \in \mathcal{N} \wedge \mathcal{X}(H) > T \tag{6}$$

$$N_{\mathrm{fp}} = \sum H \in \mathcal{P} \wedge \mathcal{X}(H) \leq T. \tag{7}$$

Therefore, the best threshold is

$$T = \arg\min_{T} \left(\frac{N_{\mathrm{fp}}}{N_n} + \frac{N_{\mathrm{fn}}}{N_p} \right). \tag{8}$$

Of course, another way to choose the threshold would be to minimize a utility function. If, for example, the false negatives are considered more dangerous, we could assign them a higher weight when choosing the threshold. Of course, this will increase the number of false positives.

The classification score is the weighted sum of thresholded $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ values and of the previous classification score $\mathcal{S}_t$ if the object is tracked

$$\mathcal{S} = \frac{\alpha(\mathcal{A} - T_A) + \beta(\mathcal{B} - T_B) + \gamma(\mathcal{C} - T_C) + \theta\mathcal{S}_{t-1}}{\alpha + \beta + \gamma + \theta}. \tag{9}$$

If any of these scores are missing, then it is considered equal to the threshold minus one small ϵ value, which acts as a penalty for the missing feature (therefore, a missing value will have a small negative weight, which biases the system toward nonpedestrians).

To determine the relative magnitude of α, β, γ, θ, and ϵ, we determined the detection rate of each single-feature classifier (pattern matching, motion signature, and motion periodicity). This approach is not 100% correct, because the features are not really independent. However, currently, we do not have a training set large enough to permit capturing the dependencies between the features. We determined the following magnitudes: $\alpha = 0.2$, $\beta = 0.2$, $\gamma = 0.3$, and $\theta = 0.3$. We also empirically determined that a good value for ϵ is between 0.01 and 0.08.

VII. Experimental Results

Our system was tested in many different crowded urban traffic scenarios using cameras mounted on a moving road vehicle. The detection rate is high, and our system proves to be reliable. There are few false positives, and pedestrians are detected early enough to permit taking active measures for collision avoidance. Fig. 8 shows some correctly detected pedestrians.

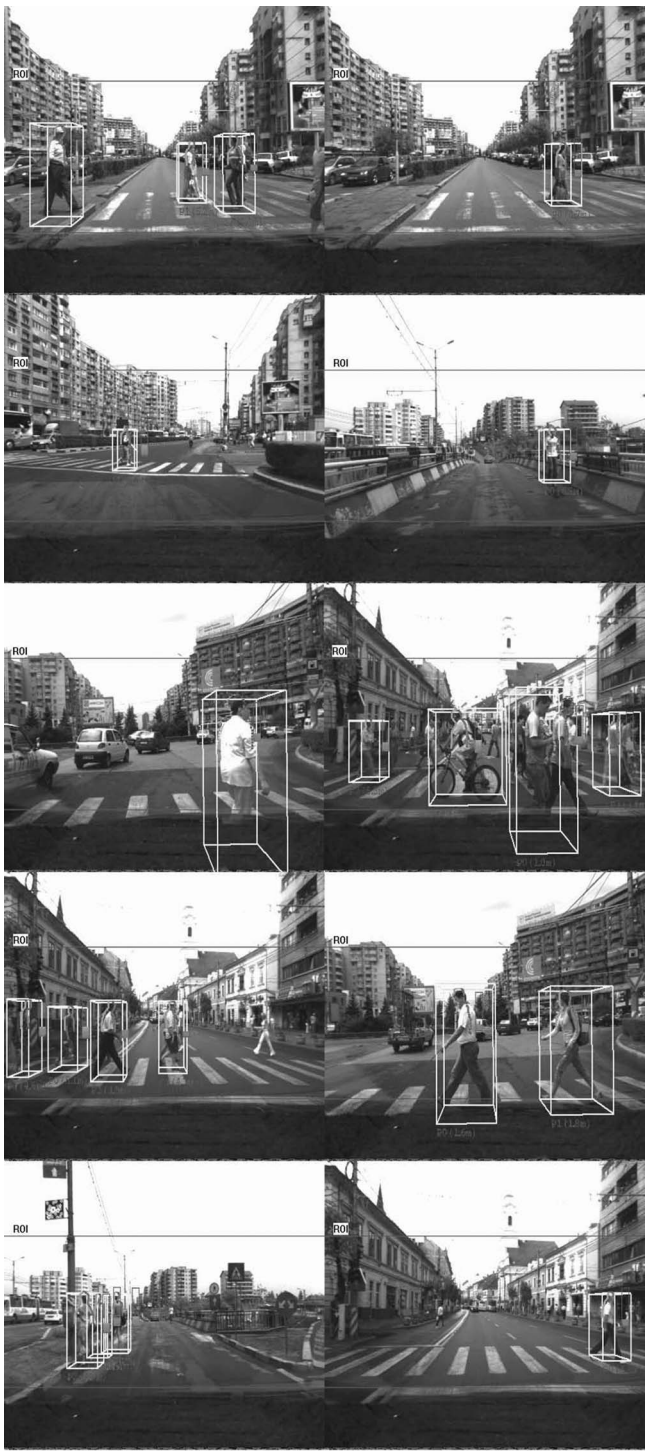

Fig. 8. Pedestrian detection results.

To obtain a quantitative measure of our system's accuracy, we used a set of 2600 manually labeled frames from various scenarios with different complexities. The detection algorithms were tested against these ground-truth data. The results are summarized in Table I. The "total objects" row represents the number of hypotheses (objects) detected using the density map. The "pedestrians" row represents the ground-truth pedestrians (manually labeled). The "others" row represents the ground-truth other objects (cars, trees, and poles). The "false

TABLE I
EXPERIMENTAL RESULTS

Scenario		Results		
Low Complexity Scenario 300 frames		Total Objects	720	100%
		Pedestrians	610	85.7%
		Others	110	15.3%
		False Pedestrians	0	0%
		Undetected Pedestrians	0	0%
		Incorrect Detection	0	0%
		Correct Detection	720	100%
Medium Complexity Scenario 800 frames		Total Objects	1003	100%
		Pedestrians	721	71.9%
		Others	282	28.1%
		False Pedestrians	23	2.3%
		Undetected Pedestrians	20	2%
		Incorrect Detection	43	4.3%
		Correct Detection	960	95.7%
High Complexity Scenario 1500 frames		Total Objects	5923	100%
		Pedestrians	1231	21.8%
		Others	3692	79.2%
		False Positives	411	6.9%
		False Negatives	392	6.6%
		Incorrect Detection	803	13.6%
		Correct Detection	5120	86.4%

pedestrians" row represents the objects that the system incorrectly classified as pedestrians but which were not pedestrians. The "undetected pedestrians" row represents the total number of objects the system classified as others, even though they were pedestrians. "Incorrect detection" is the sum of "false pedestrians" and "undetected pedestrians." "Correct detection" represents all cases of correctly classified pedestrians and other types of objects.

We concluded that all steps of our algorithm, i.e., object detection, pattern matching, and motion-based validation, are accurate and together yield a high rate of pedestrian detection. However, this detection rate is not yet sufficient for a production system, and it will have to be improved in the future. Most false-pedestrian errors occur because of trees, poles, and car pillars (particularly the D pillar). Most undetected pedestrians occur when pedestrians are too close to other objects or are severely occluded.

We also found that our detector has no problem running in real-time (we can achieve 25 ft/s on an Intel Core 2 Duo 2.6-GHz processor).

VIII. CONCLUSION AND FUTURE WORK

We developed a real-time pedestrian-detection system, exploiting 2-D and 3-D information, that is capable of detecting pedestrians in urban scenarios. Our system was designed to work as a precrash sensor on board road vehicles. The sensor will provide information for driver-warning systems and actuators.

The novelty of our system particularly consists of the powerful combination of 2-D intensity information, 3-D depth information, and motion features. Object detection using density maps is a novel and resilient pedestrian hypothesis-generation algorithm. The pattern-matching algorithm rejects all edges that do not have the correct depth. A similar approach is used for optical flow computation, where all corner points that do not have the correct depth are eliminated. Furthermore, the correct scale of the model used for matching is directly inferred using the 3-D information available. A powerful motion-based validation is used for walking pedestrians, consisting of motion signature extraction and the analysis of the periodicity of this motion signature.

Possible future work in the detection area includes using more features, such as texture, and combining all the features into a Bayesian framework. We also wish to integrate our system into a generic driving-assistance system, which, based on the information given by our pedestrian-collision sensor, will actively act to reduce the risk of potential collisions and minimize the effects of unavoidable collisions.

REFERENCES

[1] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, and M. A. Obojski, "A sensor for urban driving assistance systems based on dense stereovision," in *Proc. Intell. Vehicles*, Istanbul, Turkey, Jun. 2007, pp. 276–283.

[2] J. I. Woodfill, G. Gordon, and R. Buck, "Tyzx deepsea high speed stereo vision system," in *Proc. IEEE Comput. Soc. Workshop Real Time 3-D Sensors Their Use, Conf. Comput. Vis. Pattern Recog.*, Jun. 2004, p. 41.

[3] M. Bertozzi, E. Binelli, A. Broggi, and M. D. Rose, "Stereo vision-based approaches for pedestrian detection," in *Proc. IEEE Comput. Soc. Conf. CVPR Workshops*, 2005, p. 16.

[4] P. Kelly, E. Cooke, N. E. O'Connor, and A. F. Smeaton, "Pedestrian detection using stereo and biometric information," in *Proc. Int. Conf. Image Anal. Recog.*, 2006, pp. 802–813.

[5] H. A. Haddad, M. Khatib, S. Lacroix, and R. Chatila, "Reactive navigation in outdoor environments using potential fields," in *Proc. Int. Conf. Robot. Autom.*, Leuven, Belgium, May 1998, pp. 1232–1237.

[6] D. M. Gavrila, "Pedestrian detection from a moving vehicle," in *Proc. Eur. Conf. Comput. Vis.*, 2000, pp. 37–49.

[7] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, "Vehicle detection combining gradient analysis and Adaboost classification," in *Proc. IEEE Intell. Transp. Syst.*, 2005, pp. 66–71.

[8] D. Huber, A. Kapuria, R. Donamukkala, and M. Herbert, "Parts-based 3D object classification," in *Proc. IEEE Conf. CVPR*, 2004, pp. II-82–II-89.

[9] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D models with shape distributions," in *Proc. Shape Modeling Int.*, Genova, Italy, May 2001, pp. 154–166.
[10] F. Oniga, S. Nedevschi, M. M. Meinecke, and T. B. To, "Road surface detection based on elevation maps from dense stereo," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Oct. 2007, pp. 859–865.
[11] D. M. Gavrila, J. Giebel, and S. Munder, "Vision-based pedestrian detection: The PROTECTOR system," in *Proc. Intell. Vehicles Symp.*, Jun. 2004. pp. 13–18. [Online]. Available: http://www.gavrila.net/iv04_protector.pdf
[12] L. Havasi, Z. Szlávik, and T. Szirányi, "Pedestrian detection using derived third-order symmetry of legs," in *Proc. Int. Conf. Comput. Vis. Graph.*, 2004, pp. 733–739.
[13] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: Single-frame classification and system level performance," in *Proc. IEEE Intell. Vehicle Symp.*, Jun. 2004, pp. 1–6.
[14] T. Gandhi and M. Trivedi, "Pedestrian collision avoidance systems: A survey of computer vision based recent studies," in *Proc. IEEE ITSC*, 2006, pp. 976–981.
[15] T. Marita, F. Oniga, S. Nedevschi, T. Graf, and R. Schmidt, "Camera calibration method for far range stereovision sensors used in vehicles," in *Proc. IEEE Intell. Vehicles Symp., (IV)*, Tokyo, Japan, Jun. 2006, pp. 356–363.
[16] S. Nedevschi, C. Vancea, T. Marita, and T. Graf, "Online extrinsic parameters calibration for stereovision systems used in far-range detection vehicle applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 651–660, Dec. 2007.
[17] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[18] D. M. Gavrila and V. Philomin, "Real-time object detection for smart vehicles," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kerkyra, Greece, 1999, pp. 87–93.
[19] J. C. Russ, *The Image Processing Handbook*, 3rd ed. Boca Raton, FL: CRC, 1999.
[20] M. A. Butt and P. Maragos, *Optimum Design of Chamfer Distance Transform*. Atlanta, GA: School Elect. Comput. Eng., Georgia Inst. Technol., 1996.
[21] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, *Comparing Images Using the Hausdorff Distance*. Ithaca, NY: Dept. Comput. Sci., Cornell Univ., 1993.
[22] W. Rucklidge, "Locating objects using the Hausdorff distance," in *Proc. Int. Conf. Comput. Vis.*, 1995, pp. 457–464.
[23] M. Rowan and F. Maire, "An efficient multiple object vision tracking system using bipartite graph matching," in *Proc. FIRA*, Busan, Korea, Oct. 2004.
[24] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.
[25] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th IJCAI*, 1981, pp. 674–679.
[26] J.-Y. Bouguet (2000). *Pyramidal Implementation of the Lucas Kanade Feature Tracker*. [Online]. Available: http://robots.stanford.edu/cs223b04/algo_tracking.pdf
[27] M. Harville, A. Rahimi, T. Darrell, G. Gordon, and J. Woodfill, "3D pose tracking with linear depth and brightness constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, pp. 206–213.
[28] P. D. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, no. 2, pp. 70–73, Jun. 1967.

Sergiu Nedevschi (M'98) received the M.S. and Ph.D. degrees in electrical engineering from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, Romania, in 1975 and 1993, respectively.

He was a Researcher with the Research Institute for Computer Technologies, Cluj-Napoca, from 1976 to 1983. In 1998, he was appointed as a Professor of computer science and founded the Image Processing and Pattern Recognition Research Laboratory at TUCN, where he was the Head of the Computer Science Department from 2000 to 2004 and is currently the Dean of the Faculty of Automation and Computer Science. His research interests include image processing, pattern recognition, computer vision, intelligent vehicles, signal processing, and computer architecture.

Silviu Bota received the M.S. degree in computer science from the Technical University of Cluj-Napoca, Cluj-Napoca, Romania, in 2005. He is currently working toward the Ph.D. degree in computer science with the Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, specializing in stereovision systems for intelligent vehicles.

His research interests include pedestrian recognition, motion detection, stereovision, and image processing.

Corneliu Tomiuc received the M.S. degree in computer science from the Technical University of Cluj-Napoca, Cluj-Napoca, Romania, in 2004. He is currently working toward the Ph.D. degree in computer science with the Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, specializing in stereovision systems for intelligent vehicles.

His research interests include pedestrian recognition, shape-based recognition, stereovision, and image processing.