

PublisherInfo	
PublisherName	: Springer International Publishing
PublisherLocation	: Cham
PublisherImprintName	: Springer

Pedestrian Validation in Infrared Images by Means of Active Contours and Neural Networks

ArticleInfo									
ArticleID	: 2925								
ArticleDOI	: 10.1155/2010/752567								
ArticleCitationID	: 752567								
ArticleSequenceNumber	: 267								
ArticleCategory	: Research Article								
ArticleCollection	: Advanced Image Processing for Defense and Security Applications								
ArticleFirstPage	: 1								
ArticleLastPage	:								
ArticleHistory	: <table> <tr> <td>RegistrationDate</td> <td>: 2009-11-30</td> </tr> <tr> <td>Received</td> <td>: 2009-11-30</td> </tr> <tr> <td>Accepted</td> <td>: 2010-3-31</td> </tr> <tr> <td>OnlineDate</td> <td>: 2010-5-10</td> </tr> </table>	RegistrationDate	: 2009-11-30	Received	: 2009-11-30	Accepted	: 2010-3-31	OnlineDate	: 2010-5-10
RegistrationDate	: 2009-11-30								
Received	: 2009-11-30								
Accepted	: 2010-3-31								
OnlineDate	: 2010-5-10								
ArticleCopyright	: The Author(s)2010 This article is published under license to BioMed Central Ltd. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.								
ArticleGrants	:								
ArticleContext	: 136342010201011								

Massimo Bertozzi^{Aff1}
Corresponding Affiliation: [Aff1](#)
Email: bertozzi@ce.unipr.it

Pietro Cerri^{Aff1}
Email: cerri@ce.unipr.it

Mirko Felisa^{Aff1}
Email: felisa@ce.unipr.it

Stefano Ghidoni^{Aff2}
Email: ghidoni@ce.unipr.it

Michael Del Rose^{Aff3}
Email: delpose@tacom.army.mil

^{Aff1} VisLab, Dipartimento di Ingegneria dell'Informazione, Università di Parma, 43124 Parma, Italy
^{Aff2} IAS-Lab, Dipartimento di Ingegneria dell'Informazione, Università di Padova, 35131 Padova, Italy
^{Aff3} Vetronics Research Center, U. S. Army TARDEC, MI 48397, USA

Abstract

This paper presents two different modules for the validation of human shape presence in far-infrared images. These modules are part of a more complex system aimed at the detection of pedestrians by means of the simultaneous use of two stereo vision systems in both far-infrared and daylight domains. The first module detects the presence of a human shape in a list of areas of attention using active contours to detect the object shape and evaluating the results by means of a neural network. The second validation subsystem directly exploits a neural network for each area of attention in the far-infrared images and produces a list of votes.

1. Introduction

During the last years, pedestrian detection has been a key topic of the research on intelligent vehicles. This is due to the many applications of this functionality, like driver assistance, surveillance, or automatic driving

systems: moreover, the heavy investments made by almost all car manufacturers on this kind of research prove that particular attention is now focused on improving road safety, especially for reducing the high number of pedestrians being injured every year. Also the U.S. Army is actively developing systems for obstacle detection, path following, and anti-tamper surveillance, for its robotic fleet [1, 2].

Finding pedestrians from a moving vehicle is, however, one of the most challenging tasks in the artificial vision field, since a pedestrian is one of the most *deformable* object that can appear in a scene. Moreover, the automotive environment is often barely unstructured, incredibly variable, and apparently moving, due to the fact that the camera itself is in motion; therefore, really few assumptions can be made on the scene.

This paper describes two modules for pedestrian validation developed for integration into a vision-based obstacle detection system to be installed on an autonomous military vehicle. This system is able to detect all obstacles appearing in the scene and is based on the simultaneous use of two stereo camera systems: two far-infrared cameras and two daylight cameras [3]. The first stages of this system provide a reliable detection of image areas that potentially contain pedestrians; following stages are devoted to refine and filter these rough results to validate the pedestrians presence. The validation is based on a multivote system: several approaches are independently used to analyze areas of attention, and each subsystem outputs a vote describing how much the obstacle is likely to be a pedestrian. Then, a final validation is done, based on all votes.

This paper describes two of the intermediate validation modules. The first one has been developed and, in an initial stage, extracts objects shape by means of active contours [4], then provides a vote using a neural network-based approach. The second validation stage directly exploits a neural network for evaluating the presence of human shapes in far-infrared images.

This paper is organized as follows. Section 2 describes related work in pedestrian detection systems based on artificial vision. The pedestrian detection system is discussed in Section 3. The module for active contours-based shape detection algorithm is detailed in Section 4 while Section 5 describes the neural network-based validation step. Finally, Section 6 ends the paper presenting few results and remarks on the system.

2. Related Work

For the U.S. Army the use of vision as a primary sensor for the detection of human shapes is a natural choice since cameras are noninvasive sensors and therefore do not emit signals.

Vision-based systems for pedestrian detection have been developed exploiting different approaches, like the use of monocular [5, 6] or stereo [7, 8] vision. Many systems based on the use of a stationary camera employ simple segmentation techniques to obtain foreground region; but this approach fails when the pedestrians have to be detected from moving platforms. Most of the current approaches for pedestrian detection using moving cameras treat the problem as a recognition task: a foreground detection is followed by a recognition step to verify the presence of a pedestrian. Some systems use motion detection [7, 9] or stereo analysis [10] as a means of segmentation.

Other systems substitute the segmentation step with a focus-of-attention approach, where salient regions in feature maps are considered as candidates for pedestrians. In the GOLD system [11], vertical symmetries are associated with potential pedestrians. In [12] the local image entropy directs the focus-of-attention followed by a model-matching module.

For what concerns the recognition phase, recent researches are often motion based, shape based, or multicut based. Motion-based approaches use the periodicity of human gait or gait patterns for pedestrian detection [7, 12]. These approaches seem to be more reliable than shape-based ones, but they require temporal information and are unable to correctly classify pedestrians that are still or have an unusual gait pattern.

Shape-based approaches rely on pedestrians appearance; therefore both moving and stationary people can be detected [11, 13]. In these approaches, the challenge is to model the variations of the shape, pose, size and appearance of humans, and their background. Basic shape analysis methods consist in matching a template with candidate foreground regions. In [14], a tree-based hierarchy of human silhouettes is constructed and the matching follows a coarse-to-fine approach. In [15, 16], probabilistic templates are used to take into account the possible variations in human shape. As a final step of the recognition task, some systems also exploit pattern-recognition techniques based on the use of classifiers, or in combination with a shape analysis with gait detection [14, 17].

For the task of human shape classification, the most common classifiers are support vector machine [18], adaboost [19], and neural networks. Concerning the systems adopting the neural networks approach, most of them first extract features from images, and then use these features as the input of the classifier. In [10], foreground objects are first detected through foreground/background segmentation, and then classified as pedestrian or nonpedestrian by a trained neural network. Conversely, other systems are based on the direct use of neural network on images. As an example, in [20], convolutional neural networks are used as feature extractor and classifier.

3. System Description

The algorithms described in this work have been developed as a part of a tetra-vision-based pedestrian system [3, 21]. The whole architecture is based on the simultaneous use of two far-infrared and two daylight cameras. Thanks to this approach, the system is able to detect obstacles and pedestrians when the use of infrared devices is more appropriate (night, low-illumination conditions, etc.) or, conversely, in case visible cameras are more suitable for the detection (hot, sunny environments, etc.).

In fact, FIR images convey a type of information that is very different from those in the visible spectrum. In the infrared domain the image of an object depends on the amount of heat it emits, namely, it is generally related to its temperature (see Figure 1). Conversely, in the visible domain, objects appearance depends on how the surface of the object reflects the incident light as well as on the illumination conditions.

Figure 1 Examples of typical scenarios in FIR and visible images.

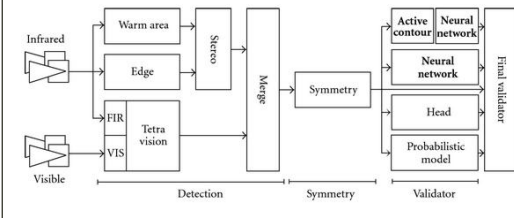


Since humans usually emit more heat than other objects like trees, background, or road artifacts, the thermal shape can be often successfully exploited for pedestrian detection. In such cases, pedestrians are in fact brighter than the background. Unfortunately, other road participants or artifacts emit heat as well (cars, heated buildings, etc.). Moreover, infrared images are blurred and have a poor resolution and the contrast is low compared with rich and colorful visible images.

Consequently, both visible and far-infrared images are used for reducing the search space.

Figure 2 depicts the overall algorithm flow for the complete pedestrian system. Different approaches have been developed for the initial detection in the two image domains: warm areas detection, vertical edges detection, and an approach based on the simultaneous computation of disparity space images in the two domains [3, 21].

Figure 2 Overall algorithm flow.



These first stages of detection output a list of areas of attention in which pedestrians can be potentially detected. Each area of attention is labelled using a bounding box. A symmetry-based approach is further used to refine this rough result in order to resize bounding boxes or to separate bounding boxes that can contain more pedestrians.

These two steps in the processing, barely, take into account specific features of pedestrians; in fact, only symmetrical and size considerations are used to compute the list of bounding boxes. Therefore, independent validation modules are used to evaluate the presence of human shapes inside the bounding boxes. These stages exploit specific pedestrian characteristics to discard false positives from the list of bounding boxes. In the following paragraphs the two validators shown as bold in Figure 2 are described and detailed.

A final decision step is used to balance the votes of validators for each bounding box.

4. Active Contour-Based Validator

As previously discussed, the pedestrian validation step is composed by several validators, each one supplying a vote that is then provided to the final evaluation step. The validator detailed in this section is based on the analysis of a pedestrian shape, which can be extracted using the well-known active contour models, also known as snakes.

4.1. Active Contour Models

Active contour models are widely used in pattern recognition for extracting an object shape. First introduced by [22], this topic has been extensively explored also in the last years. Basically, a snake is a curve described by the parametric equation $\mathbf{v}(s) = (x(s), y(s))$, where s is the normalized length, assuming values in the range $[0, 1]$. This continuous curve becomes, in a discrete domain, a set of points that are pushed by some energies that depend on the specific problem being addressed. Indeed, on the image domain, over which a snake moves, energy fields are defined, which affect the snake movements. Such energy fields depend on the original image, or on an image obtained by processing the original one, in order to highlight those features by which the snake should be attracted.

The points of the contour then move according to both these external forces and other forces that are said to be internal to the snake, that is, that control the way each snake point influences its neighbors.

The two challenges when dealing with snakes are, on one hand, a good choice of the external forces, in order to efficiently guide the snake toward the desired image features, and on the other hand, a correct decision on the snake internal parameters that should provide to the snake the desired "mechanical" properties.

Regarding external forces, it should be noted that they must generate something similar to an energy field: it is therefore not enough to choose the important features, but rather, a method must also be defined, in order to create the field: the snake behavior should be affected by the features also at a certain distance—this, after all, is the meaning of force field.

Every point composing the snake reaches a local energy minimum: this means that the active contour does not find a global optimum position: rather, since it is based on local minimization, the final position strongly depends on the initial condition, that is, the initial snake position.

Because initial stages of the pedestrian detection system provide a bounding box for each detected object, the snake initial position can be chosen as the bounding box contour: then, a contracting behavior should be impressed, to force the snake to move inside the bounding box. Other energies must also be introduced to make the snake stop when the object contour is reached.

It was said that there are two kinds of forces, and associated energies that control snake movements and that can be divided into two different categories: internal and external. Because internal energy comes from interactions between points, it depends only on the topology of the snake, and controls the continuity of the curve derivatives: it is evaluated by the equation

$$E_{int} = \alpha(s) |v_s(s)|^2 + \beta(s) |v_{ss}(s)|^2, \quad (1)$$

where $v_s(s)$ and $v_{ss}(s)$ are, respectively, the first and second derivatives of $v(s)$ with respect to s . The first contribution appearing in the sum represents the tension of the snake that is responsible for the elastic behavior; the second one gives the snake resistance to bending: $\alpha(s)$ and $\beta(s)$ are weights.

Therefore, internal energy controls the snake mechanical properties, but is independent of the image; external energy, on the contrary, causes the snake to be attracted to the desired features, and should therefore be a function of the image.

Analytically, the snake will try to minimize the whole energy balance, given by the equation

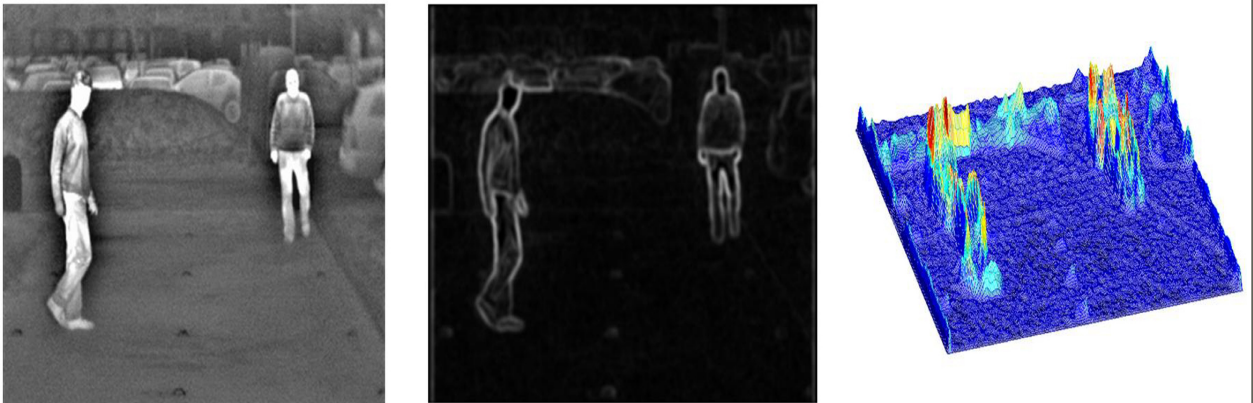
$$\bar{E}_{snake} = \int_0^1 (\bar{E}_{int}(v(s)) + \bar{E}_{ext}(v(s))) ds. \quad (2)$$

Because energies are the only way to control a snake, a proper choice of both internal and external energies should be made. In particular, the external energy depending on the image must decrease in the regions where the snake should be attracted. In the following, the energies adopted to obtain an object shape are described.

As previously said, the initial snake position is chosen to be along the bounding box contour. In this system both visible and far-infrared images are available, but the latter seem much more convenient when dealing with pedestrians, due to the thermal difference between a human being and the background [3].

To extract a pedestrian shape, the Sobel filter output is a useful starting point: moreover, the edge image is needed also by previous steps of the recognition algorithm; therefore it is already available. A Gaussian smoothing filter is then applied to enlarge the edges, and consequently the area capable of influencing the snake behavior, that is, the area where the field generated by external forces is sensible. The resulting image is then associated with an energy field that pushes the snake towards the edges: for this reason, the brighter a pixel in that image, the lowest the associated energy: in this way, snake's (the points into which the snake is discretized) are attracted by the strongest edges; see Figure 3.

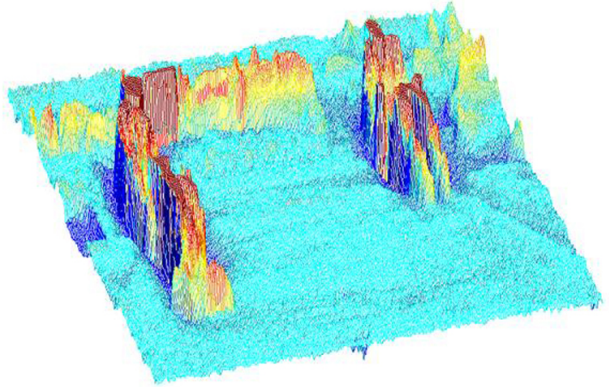
Figure 3 Energy field due to edges: (a) original image, (b) edge image obtained using Sobel operator and gaussian smooth, and (c) edge energy functional with inverted sign, to obtain a more effective graphical representation.



Bright regions of the original FIR image are also considered. In fact, smoothed edges do not accurately define the object contour (mainly because they are smoothed): snake contraction has to be arrested by bright regions in the FIR image that can belong to a portion of a human body (see Figure 4). This method lets the snake

correctly adapt to a body shape in a lot of situations, and it should also be noticed that this mechanism works only if there are hot regions inside the bounding box: a useful side effect, then, is an excessive snake contraction when there are not warm blobs inside a bounding box.

Figure 4 Energy field due to the image: (a) original image and (b) intensity energy functional with inverted sign, to obtain a more effective graphical representation.



The minimum energy location is found by iteratively moving each snaxel, following an *energy minimization algorithm*. Many of them were proposed in the literature. For this application, the greedy snake algorithm [23], applied on 5×5 neighborhood, was adopted.

During the initial iterations, the snake tends to contract, due to the elastic energy; this tendency stops when some other energy counterweights it, for instance, the presence of edges or a light image region. While adapting to the object shape, the snake length decreases, as well as the mean distance between two adjacent snaxels. Since this mean distance is a value that affects the internal energy, in order to keep almost constant the elastic property also during strong contraction, the snake is periodically resampled using a fixed step; in this way some unwanted snaxels accumulation can be avoided.

Due to the iterative nature of the snake contraction, computational times are not negligible. On a Core2 CPU working at 2.13 GHz the algorithm needs a time that is below 20 ms for each snake, and sensibly lower for small targets. This computational load makes the use of this technique feasible in a system that is asked to work at several frames per second, like the one being described.

4.2. Double Snake

The active contour technique turned out to be effective, but it showed some weaknesses when adapting to concave shapes, like those created by a pedestrian when his legs are open. In this case, the active contour needs to sensibly extend his length while wrapping around the concave shape, but this process is usually not complete because of the elastic energy. Moreover, the initialization, that is, the initial configuration of the snake, strongly influences the shape extracted at the end of the process. To increase the capability of adapting to concave shapes, and to partially solve the dependence on the initialization, the study in [24] proposed a technique based on two snakes: a snake external to the shape to recover, like the one previously discussed, and a new one, placed inside the pedestrian shape, that tends to adapt from inside, driven by a force that makes the snake expand, instead of contracting. Moreover, the two snakes do not evolve independently, but rather interact: how they do that is a key point in the development of this technique. The simplest interaction is obtained by adding in (2) a contribution that depends on the position of the other snake, so that each one tends to move towards the other.

Note, however, that there is no guarantee that the two snakes will get very close, as there can be strong forces that make the two snakes remain far from each other: for this reason, the tuning of the parameters in the energy calculation should be carefully performed, so that the force between the two contours can balance the other components. This task turns out to be particularly difficult when dealing with images taken in the automotive scenario, which usually present a huge amount of details and noise; it is in fact very difficult to find a set of parameters providing a good attraction between the two snakes, and, at the same time, letting them free of moving towards the desired image features.

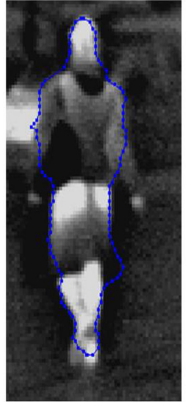
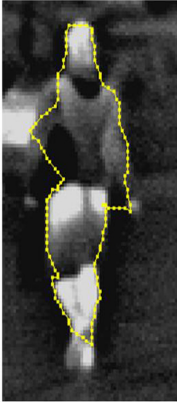
Alternatively, the snake evolution can be controlled by a new behavior that ensures that the two snakes will get very close to each other. Such behavior is based on the idea that, at each iteration, every snaxel should move towards the corresponding snaxel on the other snake. Snaxels are therefore coupled, so that each snaxel in one snake has a corresponding one in the other contour. Then, during the iteration process, snaxels couples are considered: for each of them, one of the points is moved towards the other one, the latter remaining in the same position; the moving point is chosen so that the energy of the couple is minimized. In general, the number of points is different for the two snakes, this means that a snaxel of the shorter contour can be included in more than one couple: such points have a greater probability of being moved, but this effect does not jeopardize the shape extraction.

In this approach the energy balance is still considered, but here it has a slightly different meaning, because it is used to choose which snaxel in the couple should move. This gives a great power to the force that attracts the two snakes, and the drawback is that they can therefore neglect the other forces, namely, the features of the image that should attract them. To mitigate this power, every two iterations with the new algorithm, an iteration with the classical greedy snake algorithm is performed, so that the snakes are better influenced by the image and by the internal energy. This solution turned out to be the most effective one.

Some examples and performance comparisons of contour extraction are presented in Figure 5: in the left column, a simple case is presented: the contour of the same pedestrian is extracted using the single snake technique (a) and the double snake (c). Then, in (b) and (d) a more complex scene is considered: together with a

pedestrian, some other obstacles are detected in the frame: all of the contours are extracted for the classification. In this case it can be analyzed the behavior of the shape extractor when dealing with obstacles other than pedestrians that are usually colder than a human being: as a result, in the FIR images they will appear dark, and will therefore lack the features that attract the snakes. In this situation, contours extracted using the double snake algorithm (d) tend to become similar to a square, and are clearly different from the shape of a pedestrian; this difference is not so high using the single snake technique, as can be seen in (b).

Figure 5 Examples of shape extraction. In (a), the contour of a pedestrian is extracted using the single snake algorithm, while (c) shows the result when the double snake technique is used; it can be seen that the contour is smoother in the latter case. In (b) a more complex situation is analyzed using the single snake technique, and (d) presents the same scene analyzed by the double snake algorithm (the red contour is the inner one, while the green snake is the outer one).



4.3. Neural Network Classification

Once the shape of each obstacle is extracted, it has to be classified, in order to obtain a vote to provide to the final validator. Obstacles shapes extracted using the active contour technique are validated using a neural network.

Prior to be validated, extracted shapes should be further processed: the neural network needs a given number of input data, but each snake has a number of points that depend on its length. For this reason, each snake is resampled with a fixed number of points, and the coordinates are normalized in the range $[0, 1]$. The neural network has 60 input neurons, two for each of the 30 points of the resampled snake, and only one output neuron that provides the probability that the contour represents a pedestrian; such probability will be, again, in the range $[0, 1]$.

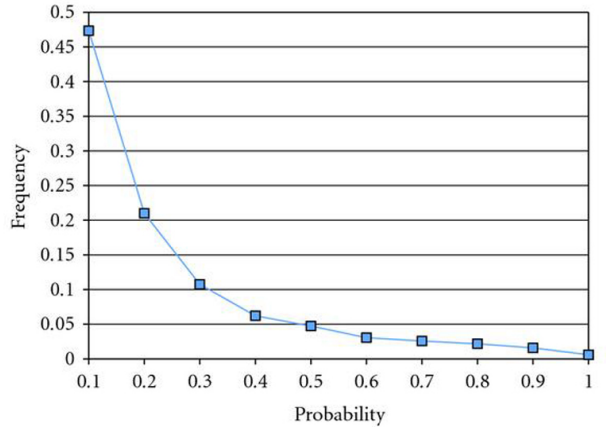
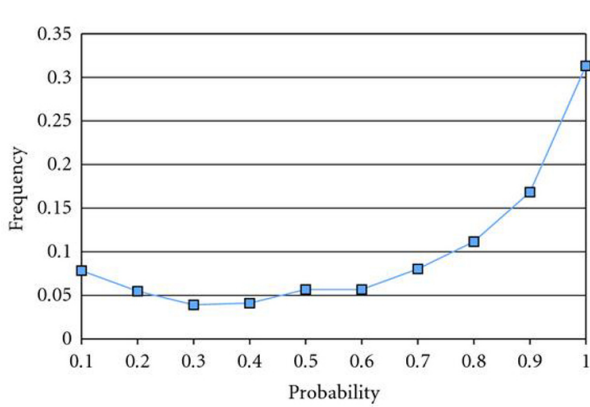
For the training of the network, a dataset of 1200 pedestrian contours and roughly the same number of contours of other objects has been used. They have been chosen in a lot of short sequences of consecutive frames, so that each pedestrian appeared in different positions, but avoiding to use too many snakes of the same pedestrian. During the training phase, the target output has been chosen as 0.95 and 0.05 for pedestrians and nonpedestrians, respectively; extreme values, like 0 or 1, have been avoided, because they could have produced some weighting parameters inside the network to assume a too high value, with negative influence on the performance.

This classifier was tested on several sequences. Recall that the output of the neural network is the probability that an obstacle is a pedestrian; it is therefore interesting to analyze which values are assigned

to pedestrians and other objects on the test sequences. Output values of the network are shown in Figure 6 (a) which represents the output values distribution when pedestrians are classified, while (b) is the distribution when contours of objects that are not pedestrians are analyzed.

Figure 6 Distribution of the neural network output values. On

[\[MediaObjects/13634_2009_Article_2925_I Eq13_HTML\]](#) the x-axis are plotted the probability values given by the neural network, while on the y-axis is reported the occurrence of each probability value when the shapes of pedestrians (a) and other objects (b) are analyzed.



It can be seen that classification results are accurate, and this classifier was therefore included in the global system depicted in Figure 2. Moreover, the performance was evaluated also considering this classifier by itself, and not as a part of a greater system. A threshold was therefore calculated to obtain a hard decision: the best value turned out to be 0.4, which provided a correct classification of 79% of pedestrians and 85% of other objects.

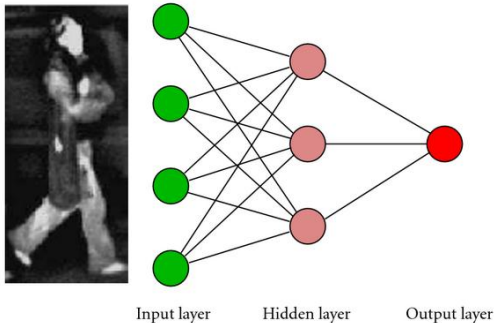
The computational time of a neural network can be neglected, since it is anyway below 1 ms.

5. Neural Network-Based Validator

This section describes the neural network-based validator, shown in Figure 2. A feed-forward multilayer neural network is exploited to evaluate the presence of pedestrians in the bounding boxes detected by previous stages. Since neural networks can express highly nonlinear decision surfaces, they are especially appropriate to classify objects that present a high degree of shape variability, like a pedestrian. A trained neural network can implicitly represent the appearance of pedestrians in various poses, postures, sizes, clothing, and occlusion situation.

In the system described here, the neural network is directly trained on infrared images. Generally, neural network-based systems, working on daylight images, do not exploit directly the image; in fact, it is not appropriate for encoding the pedestrian features, since pedestrians present a high degree of variability in color and texture and, moreover, intensity image is sensitive to illumination changes. Conversely, in the infrared domain the image of an object depends on its thermal features and therefore it is nearly invariant to color, texture, and illumination changes. The thermal footprint is a useful information for the neural network to evaluate the pedestrian presence and, therefore, it is exploited as a direct input for the net (Figure 7). Since a neural network needs a fixed-sized input ranged from 0 to 1, the bounding boxes are resized and normalized.

Figure 7 A three-layer feed-forward neural network: each neuron is connected to all neurons of the following layer. The infrared bounding boxes are exploited as input of the network.



The net has been designed as follows: the input layer is composed by 1200 neurons, corresponding to the number of pixels of resized bounding boxes (20×60). The output layer contains a single neuron only and its

output corresponds to the probability that the bounding box contains a pedestrian (in the interval $[0,1]$). The net features a single hidden layer. The number of neurons in the hidden layer has been computed trying different solutions; values in the interval 25–140 have been considered.

The network has been trained using the back-propagation algorithm. The training set is generated from the results of the previous detection module that were manually labelled. Initially, a training set, composed by 1973 examples, has been created. It contains 902 pedestrians, and 1071 nonpedestrians examples ranging from traffic sign poles, vehicles, to trees. Then, the training set has been expanded to 4456 examples (1897 of pedestrian and 2559 of nonpedestrian) in order to cover different situations and temperature conditions and to avoid the overfitting. Moreover, an additional test set has been created in order to evaluate the performance of the validator.

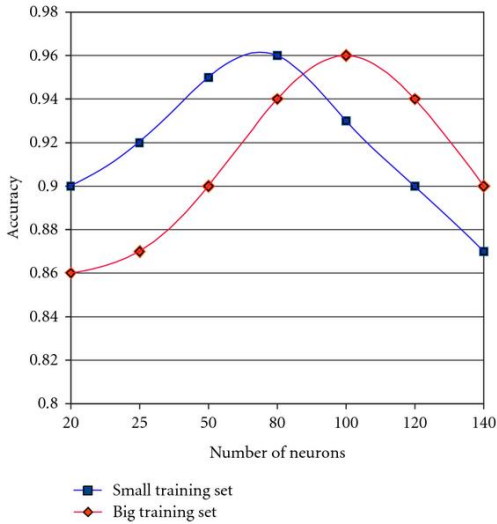
The network parameters are initialized by small random numbers between 0.0 and 1.0, and are adapted during the training process. Therefore, the pedestrian features are learnt from the training examples instead of being statically predetermined. The network is trained to produce an output of 0.9 if a pedestrian is present, and 0.1 otherwise. Thus, the detected object is classified thresholding the output value of the trained network: if the output is larger than a given threshold, then the input object is classified as a pedestrian, otherwise as a nonpedestrian.

A weakness of the neural network approach is that it can be easily overfitted, namely, the net steadily improves its fitting with the training patterns over the epochs, at the cost of diminishing the ability to generalize to patterns never seen during the training. The overfitting, therefore, causes an error rate on validation data larger than the error rate on the training data. To avoid the overfitting, a careful choice of the training set, the number of neurons in the hidden layer, and the number of training epochs must be performed.

In order to compute the optimal number of training epochs, the error on validation dataset is computed while the network is being trained. The validation error decreases in the early epochs of training but after a while it begins to increase. The training session is stopped if a given number of epochs have passed without finding a better error on validation set and if the ratio between error on validation set and error on training set is greater than a specific value. This point represents a good indicator of the best number of epochs for training and the weights at that stage are likely to provide the best error rate in new data.

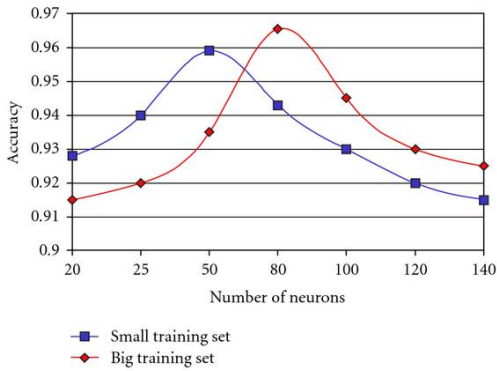
The determination of number of neurons in the hidden layer is a critical step as it affects the training time and generalization property of neural networks. Using too few neurons in the hidden layer, the net results inadequate to correctly detect the patterns. Too much neurons, conversely, decreases the generalization property of the net. Overfitting, in fact, occurs when the neural network has so much information processing capacity that the limited amount of information contained in the training set is not enough to train all of the neurons in the hidden layer. In Figure 8, the accuracy of the net on validation set depending on the number of neurons in hidden layer is shown. With a larger training set, a bigger number of neurons in the hidden layer are required. This is caused by the bigger complexity of the training set that contains pedestrians in different conditions. Therefore, a net with more processing capacity is needed.

Figure 8 The accuracy of the net on validation set depending on the number of neurons in hidden layer. The optimal neurons number is a tradeoff between underfitting and overfitting.



The trained nets have been tested on the test set that is strictly independent to the training and validation set. It contains examples of pedestrians and nonpedestrians in various poses, shapes, sizes, occlusion status, and temperature conditions. In Figure 9, the accuracy of the net on test set varying the number of neurons in hidden layer is shown. The performance of the nets, trained on the big training set, is greater than that trained on the small set. This is caused by a higher completeness of the training set. The performance of nets is similar to that performed on validation set (Figure 8); but the optimal number of neurons in the hidden layer is lower. The net having 80 neurons in the hidden layer and trained on big training set is the best one, achieving an accuracy of 96.5% on the test set.

Figure 9 The accuracy of the net on test set depending on the number of neurons in hidden layer.



6. Discussion

The developed system has been tested in different situations using an experimental vehicle equipped with the tetra-vision system (see Figure 10).

Figure 10 The tetra-vision far-infrared and daylight acquisition system installed on board of the test vehicle.

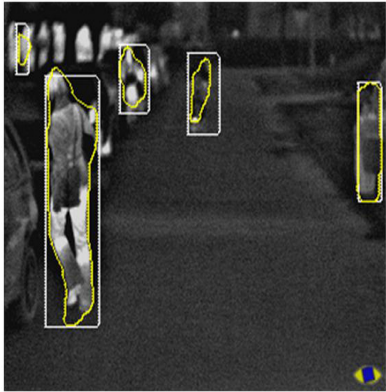


Tests were performed on both validation techniques separately, in order to understand the strong and weak points of each of them: such a knowledge is needed by the final validator in order to properly adjust the weights of the soft decisions. The discussion will therefore focus on results given by both neural networks, one working on shapes extracted by the active contours technique and the other one directly on the regions of interest found by the algorithm early stages.

As previously described, the approach chosen for the classification of pedestrians contours is based on a neural network, an approach that gives good results when the problem description turns out to be complex. A neural network suitable for the classification of pedestrians contours was developed, which provided good results, as can be seen in Figure 6.

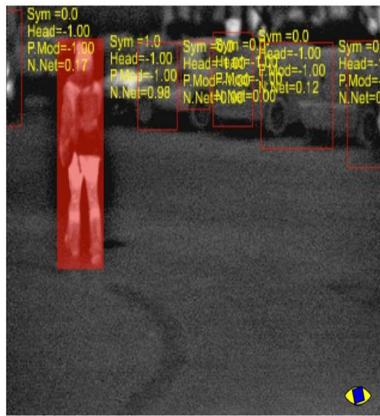
In Figure 11 some examples of the contraction mechanism are reported: the white lines are the snakes in the initial position, that is, on the bounding box contour, while the snakes after energy minimization are drawn in yellow. Some examples are presented for a close pedestrian, Figure 11(a), and for a distant pedestrian and a motorbike, Figure 11(b). In Figure 11(c) the importance of the initial snake position is highlighted: the head is not detected because it is outside of the initial snake position (in white). Some shape extraction results are presented when the FIR images are not optimal, like those acquired in summer, under heavy direct sunlight: in this condition, many objects in the background become warm, and the assumption that a pedestrian has a higher temperature than the background is not satisfied. This causes some errors in the contraction process, so that the snake in the final position does not completely adhere to the pedestrian contour, but also includes some background details (Figure 11(d)).

Figure 11 Results: in (a) and (b), shape extraction of a close and distant pedestrian, respectively; the white snake represents the initial position, while the yellow one is the final configuration. In (c), a typical issue connected with a wrong initial snake disposition is shown: the head is outside the extracted shape because it was also outside the bounding box. In (d) some results in a difficult working condition are presented, that is, during summer, when a lot of background objects appear bright, due to the high temperature.



In Figure 12 some classification results of the neural network that analyzes pedestrians shapes are shown. In Figure 12(a), a lot of potential pedestrians are found by the obstacle detector of previous system stages, but only one is classified as a pedestrian, with a vote of 0.98, while all the other obstacles received a vote not greater than 0.17. In Figure 12(b) a scene with a lot of pedestrians is shown and two obstacles: the latter received votes not exceeding 0.19, while one of the pedestrians received a vote of 0.44, and all the others votes greater than 0.85. In Figure 12(c), a distant pedestrian is correctly classified with a vote of 0.84; in Figure 12(d) two pedestrians are present, at different distances, and are correctly classified, with votes of 0.87 and 0.77.

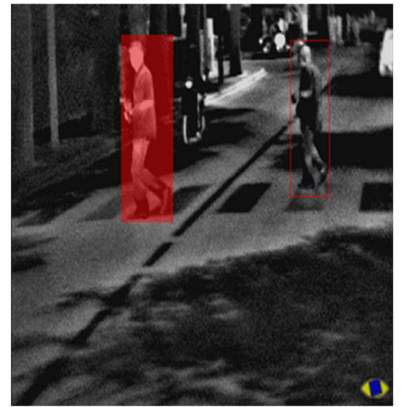
Figure 12 Classification results of the neural network analyzing pedestrians shapes. Bounding boxes that are filled are classified as pedestrians, while a red contour is put around obstacles that are classified as nonpedestrians. Output values are also printed on the image.



Concerning the neural network-based validator, a feed-forward multilayer neural network is exploited to evaluate the presence of pedestrians in the bounding boxes detected by previous stages of the tetra-vision system. The neural network is trained on infrared images in order to acknowledge the thermal footprint of pedestrians. The training set has been generated from the results of the previous detection modules that were manually labelled. Such set contains a large number of pedestrian and nonpedestrian examples, like traffic sign poles, vehicles, and trees, in order to cover different situations and temperature conditions. Different neural nets have been trained to understand which is the optimal number of training epochs, neurons in the hidden layer of the net, and training examples and, therefore, to avoid the overfitting. The test set containing also pedestrians partially occluded or with missing parts of the body has been generated in order to evaluate the performance of net. Experimental results show that the system is promising, achieving an accuracy of 96.5% on the test set.

Figure 13 shows some results of the neural network validator. The validated pedestrians are shown using a superimposed solid red box. Conversely, the empty rectangles represent the bounding boxes generated by previous steps and classified as nonpedestrians. Figures 13(a) and 13(b) depict examples of pedestrians and nonpedestrians correctly classified. In Figure 13(c), an area of attention is not correctly validated because it contains multiple pedestrians, and they are not in the typical pedestrian pose. Some false positives are presented in Figures 13(d) and 13(e).

Figure 13 Neural network results: validated pedestrians are shown using a superimposed red box; the white rectangles represent the discarded bounding boxes.



BodyRef	
FileRef	: BodyRef/PDF/13634_2009_Article_2925.pdf
TargetType	: OnlinePDF

Acknowledgment

This work has been supported by the European Research Office of the U. S. Army under contract number N62558-07-P-0029.

References

1. Del Rose M, Frederick P: **Pedestrian detection.** *Proceedings of the Intelligent Vehicle Systems Symposium, 2005, Traverse, Mich, USA*
2. Kania R, Del Rose M, Frederick P: **Autonomous robotic following using vision based techniques.** *Proceedings of the Ground Vehicle Survivability Symposium, 2005, Monterey, Calif, USA*
3. Bertozzi M, Broggi A, Caraffi C, Del Rose M, Felisa M, Vezzoni G: **Pedestrian detection by means of far-infrared stereo vision.** *Computer Vision and Image Understanding 2007*,**106**(2-3):194-204. 10.1016/j.cviu.2006.07.016
4. Bertozzi M, Binelli E, Broggi A, Del Rose M: **Stereo vision-based approaches for pedestrian detection.** *Proceedings of the IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum, June 2005, San Diego, Calif, USA*
5. Shashua A, Gdalyahu Y, Hayun G: **Pedestrian detection for driving assistance systems: single-frame classification and system level performance.** *Proceedings of the IEEE Intelligent Vehicles Symposium, June 2004, Parma, Italy* 1-6.
6. Zhao L: *Dressed human modeling, detection, and parts localization, Ph.D. dissertation.* Carnegie Mellon University; 2001.
7. Cutler R, Davis LS: **Robust real-time periodic motion detection, analysis, and applications.** *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2000,**22**(8):781-796. 10.1109/34.868681
8. Shimizu H, Poggio T: **Direction estimation of pedestrian from multiple still images.** *Proceedings of the IEEE Intelligent Vehicles Symposium, June 2004, Parma, Italy*
9. Polana R, Nelson RC: **Detection and recognition of periodic, nonrigid motion.** *International Journal of Computer Vision* 1997,**23**(3):261-282. 10.1023/A:1007975200487
10. Zhao L, Thorpe CE: **Stereo- and neural network-based pedestrian detection.** *IEEE Transactions on Intelligent Transportation Systems* 2000,**1**(3):148-154. 10.1109/6979.892151
11. Bertozzi M, Broggi A, Fascioli A, Sechi M: **Shape-based pedestrian detection.** *Proceedings of the IEEE Intelligent Vehicles Symposium, October 2000, Detroit, Mich, USA* 215-220.

12. Curio C, Edelbrunner J, Kalinke T, Tzomakas C, von Seelen W: **Walking pedestrian recognition.** *IEEE Transactions on Intelligent Transportation Systems* 2000,1(3):155-162. 10.1109/6979.892152
13. Beymer D, Konolige K: **Real-time tracking of multiple people using continuous detection.** *Proceedings of the IEEE International Conference on Computer Vision, 1999, Kerkyra, Island*
14. Gavrila DM: **Pedestrian detection from a moving vehicle.** *Proceedings of the European Conference on Computer Vision, July 2000* 2: 37-49.
15. Nanda H, Davis L: **Probabilistic template based pedestrian detection in infrared videos.** *Proceedings of the IEEE Intelligent Vehicles Symposium, June 2002, Paris, France*
16. Stauffer C, Grimson WEL: **Similarity templates for detection and recognition.** *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2001* 1: 221-228.
17. Philomin V, Duraiswami R, Davis L: **Pedestrian tracking from a moving vehicle.** *Proceedings of the IEEE Intelligent Vehicles Symposium, October 2000, Detroit, Mich, USA* 350-355.
18. Broggi A, Bertozzi , Del Rose M, Felisa M, Rakotomamonjy A, Suard F: **A pedestrian detector using histograms of oriented gradients and a support vector machine classifier.** *Proceedings of the IEEE International Conference on Intelligent Transportation Systems, September 2007, Seattle, Wash, USA* 144-148.
19. Overett G, Petersson L: **Boosting with multiple classifier families.** *Proceedings of the IEEE Intelligent Vehicles Symposium, June 2007, Istanbul, Turkey* 1039-1044.
20. Zarvas M, Yoshizawa A, Yamamoto M, Ogata J: **Pedestrian detection with convolutional neural networks.** *Proceedings of the IEEE Intelligent Vehicles Symposium, June 2005, Las Vegas, Nev, USA* 224-229.
21. Bertozzi M, Broggi A, Felisa M, Vezioni G, Del Rose M: **Low-level pedestrian detection by means of visible and far infra-red tetra-vision.** *Proceedings of the IEEE Intelligent Vehicles Symposium, June 2006, Tokyo, Japan* 231-236.
22. Kass M, Witkin A, Terzopoulos D: **Snakes: active contour models.** *International Journal of Computer Vision* 1988,1(4):321-331. 10.1007/BF00133570
23. Williams DJ, Shah M: **A fast algorithm for active contours and curvature estimation.** *CVGIP: Image Understanding* 1992,55(1):14-26. 10.1016/1049-9660(92)90003-L