Locally Tuned Edge Extraction With Artificial Neural Networks

Marco Accame and Francesco G.B. De Natale
Department of Biophysical and Electronic Engineering
University of Genoa
Via All'Opera Pia 11A
I-16145 Genova
ITALY

Abstract: This paper presents a new strategy that exploits Artificial Neural Networks (ANNs) for a direct selection of edge points from an image. First, the Canny spatial filtering is used to obtain a set of candidate edge points which turn out to be the local maxima of the filtered image. A preliminary smooth selection of these points that exploits neighbourhood information is made to produce a set of pseudo-edges. Some features are extracted from this set and are used to teach an ANN to classify whether or not a point belongs to a real edge. Since the selection works at the pixel level (even if on a strongly reduced subset of the whole image), the generation of training data is easy even with less expert users. Concerning performances, the ANN locally improves edge extraction where significant edges are missed by a selection criterion that is fixed all over the image (e.g. the classical hysteresis selection). The proposed method demonstrates how to provide an easy man-machine interface in those visual sensing systems used in autonomous applications that need a powerful and flexible edge extraction.

Keywords: Artificial Neural Networks, Edge Extraction, Image Processing, Visual Systems

1. Introduction

The extraction of visual primitives is a basic operation in many computer vision systems. It helps simplify image analysis by dramatically reducing the amount of processed data, while maintaining the semantics of a scene. In particular, high-frequency information related to image edges is often considered as one of the most meaningful and compact subsets oı an analyzed image [1].

An image edge is usually generated by a sharp discontinuity in the intensity function due to many possible reasons: different depths of the objects present in the scene, their reflectance properties or their lighting conditions. For this reason, an edge is related in some way to the image gradient whose intensity presents strong peaks near light discontinuities. However, in real images, there are many edges that are not related to the boundaries of an object, and that are caused by acquisition noise, surface textures, or irregular shapes. Even though such edges produce image discontinuities, they do not provide information about the semantics of a scene, like those produced by the boundaries of objects, and are usually seen as noisy data for high-level processing.

In many applications, it is necessary to process information on object boundaries in order to perform an analysis of a scene in a security surveillance context or to search for patterns for visual inspection problems, or even to find some targets to be used in autonomous navigation. A common approach is to develop an efficient Visual Sensing System (VSS) which translates the complexity of the raw image data into a simpler domain, and to use the reduced data as input to a high-level module for attaining the application goal. Such a Visual Sensing System is required to be able to discriminate and preserve only the contours of the objects that are meaningful for the scene description, and to reject those that can be seen as noisy data. For this reason, an efficient edge extraction inside the VSS is essential to any real application based on object recognition.

The most common edge detectors are multistage detectors and do not work directly with grey-level information, but take advantage of intermediate representations, and retain some particular points as candidate edges. Algorithms presented in the literature deal mainly with the first stage, which is essentially accomplished by filtering in spatial frequency [2, 3, 4] or by matching with edge templates [5, 6] and then by detecting the maxima as candidates for edge points. After this stage, most algorithms perform some form of threshold selection in order to discriminate the points belonging to features of interest from those points due to noise and false indications.

One of the most inportant edge-detection techniques is the one developed by Canny [4], often used as a comparison term for testing other algorithms. This method, apart from the optimality of the image filtering, takes advantage of a hysteresis selection of the resulting maxima which makes it possible to extract edges of better quality. However, performances are extremely sensitive to the setting of parameters: this fact involves that, to achieve a reliable detection on a wide class of images, a continuous parameter tuning is required. This can severely compromise the performances of autonomous applications, as each image requires ad hoc settings. Even if threshold selection is fundamental to achieve good results, only a few works deal with threshold setting, both with fixed values [7] and with values that are dependent on the considered image [8, 9].

As a sample for a potential industrial application, in previous works [10, 11, 12] we focused on the regulation of a Visual Sensing System used to extract visual world features for autonomous mobile robots which might adapt to the change of environmental conditions. The visual primitives are used for the recognition of some targets inside the environment (e.g. doors and other landmarks). Since this system is used in fast changing environments, a continuous correct tuning of the parameters of its modules is required (image acquisition, edge extraction and edge selection). Properly trained Artificial Neural Networks (ANNs) proved to be able to give the correct regulation, starting from some environmental features.

In particular, the edge extraction module was based on the Canny filtering, followed by a hysteresis selection whose thresholds were set by an ANN properly trained and fed by some image features. This method allows good adaptiveness to changes in environment conditions and produces edges of good quality by solving the problem of threshold setting. However, it requires many images to generate a suitable training set, since the image-pattern relation is an one-to-one relation, and the ANN requires many patterns for generalizing the association rule that maps image features into correct thresholds. Moreover, during the phase of training-set generation, it is not easy to choose the right hysteresis thresholds from among the many possible ones, even by an expert operator. Indeed, it is necessary to evaluate many different values before finding the right ones; and even the best ones may not be satisfactory enough, since the tuning process works globally inside an image, without considering local characteristics. This results in a long and difficult work that cannot give quality improvements over the classical Canny edge extraction with manual parameter setting, and only the automation of on-line extraction can yield good results.

In the authors' opinion, the solution for achieving an ANN-driven edge extraction that is efficient during on-line operation (in terms of good locally adaptive selection) and even during the training phase (in terms of a fast and easy preparation of the training set), is to perform directly edge-point selection. This way, the ANN can learn how to extract significant edges that could not be detected by using thresholds fixed all over an image because their local properties would be hidden by a global method. Moreover, since each point can generate a pattern, only a few images are needed to generate an adequate set of training features. And the generation of the related target is simpler, since the user just needs to establish whether each point of the pseudo-edge set is a true edge or not.

In the following, a new method is presented, that exploits the ANN capabilities of learning the local properties of an image for extracting edge points. First, we give an outline of the optimal Canny filtering with the hysteresis selection of the resulting candidate edge points. Then, a detailed description of the proposed method is reported: creation of a pseudo-edge set (*PES*) from the output of the Canny filtering, choice of the features able to characterize each local point of the *PES*, and use of the ANN for the classification of the edge points. Finally, we show how easily the training set is created and how the ANN improves edge extraction performances over other classical algorithms.

2. The Canny Algorithm

In [4], Canny solved the optimum edge detection by formalizing it as the problem of identifying a filter that produces the maximum response near an edge point, when some constraints were imposed:

- *Good detection:* the probabilities to miss the detection of an existing edge and to detect an edge that does not exist must be kept small.
- *Good localization:* the points belonging to a detected edge must be as near as possible to the real edge.
- *Single response:* the detector must give only one response to a single edge, or at least a fixed small number of responses.

For the one-dimensional case, let $G(x)$ be the function with the edge discontinuity in $x=0$; and let $f(x)$ be the impulse response of the searched filter, which is finite and bounded by $[-W, W]$. If we assume that the function $G(x)$ is affected by noise (additive white Gaussian noise with n_0 as its mean-squared amplitude per unit length), then the previous conditions can find a mathematical formulation with the following functional:

$$\max_{f(x)} \left\{ \left(\left| \int_{-W}^{+W} G(-x)\, f(x)dx \right| \; \left| \int_{-W}^{+W} G'(-x)\, f'(x)dx \right| \right) \Big/ \left(n_0 \int_{-W}^{+W} f^2(x)dx \right)^{-(1/2)} \; n_0 \int_{-W}^{+W} f'^2(x)dx \,)^{-(1/2)} \right) \right\}$$

which is maximized subject to the third constraint above by fixing the number of false responses R_n due to the noise.

Canny computed optimal filters $f(x)$ for some classes of edge functions, but in general, as the above functional is very difficult to solve, he found a good general approximation for any kind of edge function in the first derivative of a Gaussian which behaves only a little worse than the optimal solution:

$$f(x) = (\partial/\partial x)\, N(x), \text{with } N(x) = exp\,(-x^2/(2\,\sigma^2))$$

It is possible to extend this result to the two-dimensional case in the following way. An edge in an image $I(x,y)$ is a line whose points are characterized by a position and an orientation n which is the direction normal to the contour followed by the edge. A point of an edge of orientation n can then be found as a local maximum along n of the operator $N_n(x,y)$ (the Canny Filter) applied to the image $I(x,y)$:

$$(\partial/\partial n)\,(\,N_n(x,y)*I(x,y)\,)=0$$

where:

$$N_n(x,y) = (\partial/\partial n)\, N(x,y) = n\, \nabla N(x,y)$$

$$N(x,y) = exp\,(\,-(x^2+y^2)/(2\,\sigma^2)\,)$$

For the associativity of convolution, it is possible to locate an edge point, by first convoluting the image with $N(x,y)$, and then searching for the zeroes of any directional second derivative. Then, a point $P=(x_0,y_0)$ that satisfies the following condition:

$$(\partial^2/\partial n^2)\,(\,N(x,y)*I(x,y)\,)\Big|_{(x_0,y_0)} = 0$$

is a point of the maxima points set (*MPS*) and is a candidate to be an edge point in a direction normal to the orientation n and with intensity S:

$$n = \left[\,(\nabla(N(x,y)*I(x,y)))\,/\,|\nabla(N(x,y)*I(x,y))|\,\right]\Big|_{(x_0,y_0)}$$

$$S = \left|N_n(x,y)*I(x,y)\right|\Big|_{(x_0,y_0)} = \left|\nabla(N(x,y)*I(x,y))\right|\Big|_{(x_0,y_0)}$$

In order to better discriminate between the responses due to noise and those due to true edges, a threshold is used for discarding the points of the *MPS* with a low intensity *S*. A careful analysis of the properties of responses shows that the former responses are frequently characterized by low intensity, whereas the latter are more localized and with higher values of *S*. On the basis of this analysis it is possible to fix a value for the threshold.

To make sure that a real contour will not be split into several small segments, Canny proposed a hysteresis selection: if a candidate point has intensity that is above a higher threshold T_H, it is immediately selected, as well as its connected neighbours that are above a lower threshold $T_L<T_H$. This way, the probability of choosing an isolated edge point is reduced and the lengths of the connected chains of true edges are maximized. Figure 1 shows an example of hysteresis selection which succeeds in producing a closed edge, whereas a simple threshold selection fails.

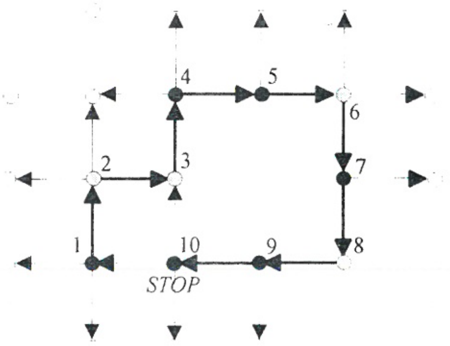

Figure 1. Scheme of Hysteresis Selection

The circles are points of the MPS. The black ones have gradient intensity above T_H, the grey ones have intensity between T_L and T_H. A point of the MPS is selected: (i) if its intensity is higher than T_H (points 1, 4, 5, 7, 9, 10), or (ii) if it is higher than T_L and has some previously selected neighbours (points 2, 3, 6, 8). The selected points *are numbered progressively and the followed connections are thicker. The dotted connections have not been considered since points 3 and 1 have already been selected. The propagation toward an initial direction stops if no neighbour point can be selected anymore.*

3. ANN Driven Direct Edge Point Extraction Using A Preliminary Canny Filtering

The hysteresis selection of the maxima points detected by the Canny filter (the points of the *MPS*) exhibits the main disadvantage that, if its threshold values are kept fixed, optimal results cannot be achieved. Thus, it is necessary to set the threshold values carefully for each image because this choice strongly influences the edge extraction in a manner different from image to image.

In a previous work [9], we solved this problem by teaching an ANN to choose the correct threshold values for a successful hysteresis selection. In particular, the ANN was fed with a number of features describing the statistics of the histogram reporting the occurrence probabilities for the edge strength of the points selected by the Canny Filter. After an initial training, performed on about 600 indoor images, the tests have shown a high percentage of success (95%) in extracting edges in a way similar to the expert's.

However, even if the results show the ability of the ANN to automatically find the correct calibration of the hysteresis thresholds, this approach exhibits the limitation that the hysteresis selection acts globally. For this reason, it cannot adapt to the local characteristics of particular image regions, thus losing some significant edges whose features are lost in the global context.

This limitation can only be solved by an approach different from the hysteresis selection. In this work, a new method is investigated that takes advantage of an ANN for a direct selection of the edge points. The ANN is trained to classify whether a point can be labelled as an edge point or not. The ANN is fed with features computed from a spatial reorganization of the points of the *MPS*, and works as an on-off classifier on each point of the reorganized data. This way, the ANN can be taught to perform a selection criterion which exploits the local properties of each point, and not only a global optimization, as in [9].

3.1 Spatial Organization of the Points in the Pseudo-Edge Set

The points of the *MPS* (the local maxima of the image after the Canny filtering) with a preliminary smooth threshold selection are used to create a set of pseudo-edges (*PES*). From this set, which is a reorganization of the points of the *MPS* in a connection context, some features are extracted that give local connection properties such as orientation and intensity information for each point and for each pseudo-edge. The *PES* contains points which are both edge points and noisy ones, and the target is to discriminate between them.

The method used for creating the *PES* works as follows. The local maxima of the output of the Canny filter are taken which are arranged in an image with a reduced number of sparse points that result to be the maxima of the gradient of the original image after a Gaussian filtering. These points belong to the *MPS* and are sparsely distributed, over the original image grid, thus having position information as well as an associated gradient vector. The points have no connection information among their possible neighbours, as the *MPS* set can be expressed as $MPS=\{(P_i(x,y), \boldsymbol{F}_i),\ i=1,\cdots,N\}$, where P_i is the generic point, x, and y, are its co-ordinates, and $\boldsymbol{F}_i$ is the associated gradient vector, with intensity S_i and orientation n_i.

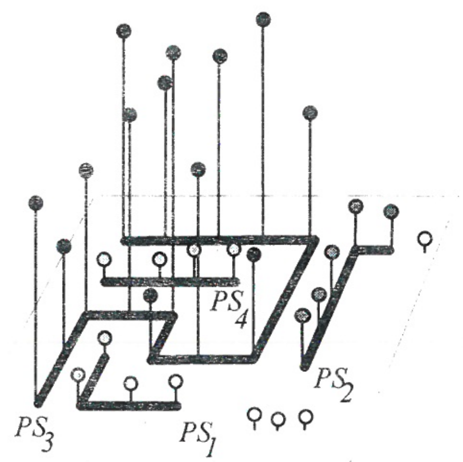

Figure 2. Example of Pseudo-edges Organization

The points of the MPS have been selected with a small threshold (white circles indicate some discarded points) and organized into four pseudo-edges. From this organization, some statistical features describing each point's properties are obtained.

The points are reordered into the *PES* which is made up of connected pseudo-edges, $PES=\{PE_j, j=1,\cdots,L\}$, where PE_j is a pseudo-edge represented as a list of connected points, $PE_j=\{P_{jk}, k=1,\cdots,M_j\}$, L is the number of pseudo-edges, and $\Sigma_j(M_j)<=N$ (i.e. the number of points in the *PES* is smaller than or equal to the number of points in the original set *MPS*). Each list is found by starting from an initial point P_i of the *MPS*. Any neighbour of the starting point P_i is recursively followed, its gradient intensity is evaluated and, if it is higher than a small threshold, it is added to the pseudo-edge PS_j and discarded from the initial set *MPS*. When there are no spatial connections anymore (or intensity values above the threshold), the propagation stops and another pseudo-edge PE_{j+1} is created, starting from a new point of the *MPS*. A very small threshold is used to partly reduce the amount of points in the *MPS* being mapped onto the *PES*, since many points with a very low intensity are mainly noise, and they are not representative at all as edges. However, the threshold has to be kept small enough in order to find edge points with both strong and small intensity values (see Figure 2).

statistics about it. Many efforts and experiments have shown the features in Table 1 as the most

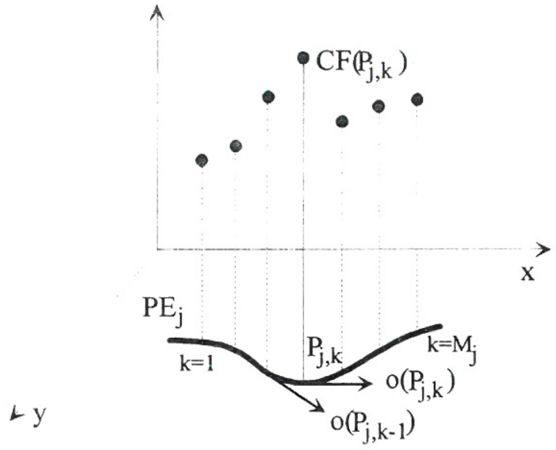

Figure 3. Examples of Some Features of A Point of the *PES*

The point P_{jk} belonging to PE_j of length $l(PE_j) = M_j$ has a gradient intensity $CF(P_{jk})$, and a differential orientation $\delta o(P_{j,k}) = o(P_{j,k}) - o(P_{j,k-1})$.

Parameter	Description	Property of
$CF(P_{jk})$	gradient intensity of the point P_{ik}, the k-th point in the PE_i	P
$\partial o(P_{jk})$	difference between the gradient orientation of the point P_{ik} and that of the previous point in PE_i	P
$\overline{CF}(PE_j)$	mean gradient intensity of the pseudo-edge PE_i	PE
$\overline{\partial o}(PE_j)$	mean difference in gradient orientation on the PE_i	PE
$CF_{max}(PE_j)$	max. gradient intensity of the pseudo-edge PE_i	PE
$\partial o_{max}(PE_j)$	max. difference in gradient orientation on PE_i	PE
$\sigma^2_{CF}(PE_j)$	variance of the gradient intensity of PE_i	PE
$\sigma^2_{\partial o}(PE_j)$	variance of the difference in gradient orientation on PE_i	PE
$l(PE_j)$	length of PE_i	PE

Table 1. The Features Set Derived from the PES Organization

A feature of the generic point P_{jk} (the k-th point of the pseudo-edge PE_j) can be a property of the point itself (property P) or of the pseudo-edge PE_j to whom the point belongs (property PE).

3.2 Features for the Description of the Points of the Pseudo-edge Set

Once the *PES* has been computed from the output of the Canny filter, some features are computed for each point of its pseudo-edges. Some features depend only on the properties of the point, whereas some others give information about the properties of the pseudo-edge using useful for our approach. See even Figure 3 for some examples of some features of a point of the *PES*.

3.3 Choice of the Target for Each Point of the Pseudo -edge Set

Using the current approach, the generation of the training set to be used to teach the ANN the correct discrimination between noise and edge points is an easy task. Indeed, once the features have been carefully designed, the training input patterns are computed automatically using some images acquired under different environmental conditions. Moreover, the association of the target with each pattern (the on/off information needed to train the ANN how to classify a point, given the associated features) is reduced to the problem of a pre-classification of the points of the *PES*.

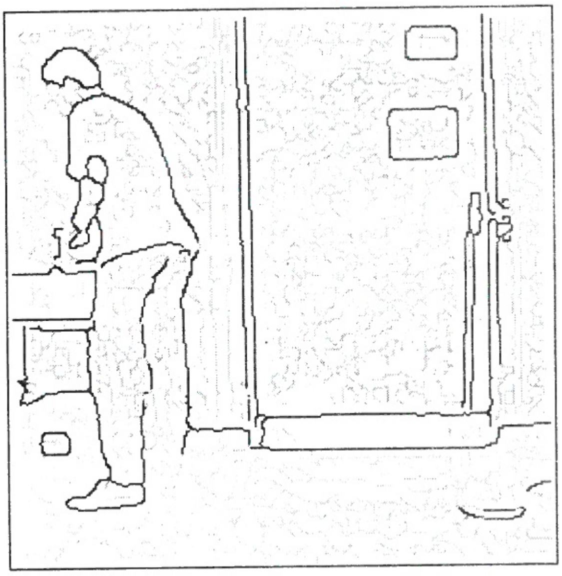

Figure 4. Example of Pixel Selection for A Training Image

The PES is the union of the points pre-classified as real edges (black) and those pre-classified as noise (grey). The user who wants to generate the targets for each point of the PES just needs to mark the chosen edge points with a graphical tool which automatically completes the task.

This pre-classification is a trivial task that can be done rather easily even by a user who is not expert in Image Processing. Indeed, with an appropriate tool, it is only required to delete from the image the points of the *PES* that are judged to be due to noise. It is a great step forward as compared with the preparation of the training data for the method developed in [9], where the user had to try several possibilities for each target (i.e. for each of the 600 images of the training set), and had to be expert, since the relation between a target and the resulting edge extraction is not so obvious as for the method presented (see Figure 5).

3.4 Characteristics of the ANN Used for the Classification of the Points

The ANN used for the classification task is a Multi-Layer Perceptron with one hidden layer. The final architecture of the network consists of 9 input neurons and 1 output neuron. Several experiments have shown that a hidden layer of 21 neurons gives the best results, aiming at generalizing the classification rule.

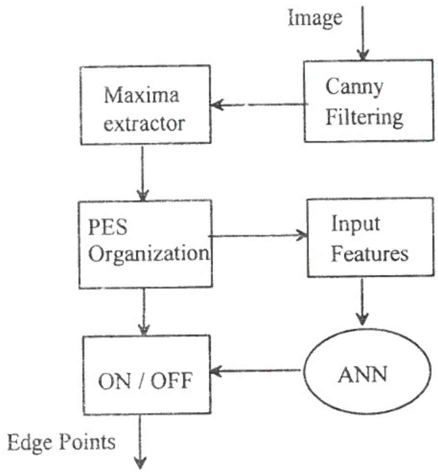

Figure 5. On-line ANN Extraction of Edge Points

An image is filtered with the Canny Filter and the features of the point of the PES are computed and given as input to the ANN that classifies each point as an edge or noise.

Concerning the off-line training strategy, the network uses the Error Back Propagation technique with an accelerated implementation [13] which features a good speed-up, as compared with the basic algorithm [13]. In our simulation, we have used a non-linear activation function for the neurons which is different from the one used in [Vo88]. It allows to manage both positive and negative values and can be expressed as follows:

$$f(z) = (e^{z}-e^{-z}) / (e^{z}+e^{-z}), \quad \text{with } f: R \rightarrow (-1, +1)$$

Let x be the input vector (x_i, $i=1,..,I$), h the hidden vector (h_j, $j=1,..,J$), and y the vector of the output values (y_k, $k=1,..,K$). $[U]$ and $[V]$ be the matrices of the input and output weights, respectively. The x, h and y vectors are related to one another by the following equations:

$$h_j = f(\Sigma_m (U_{mj} x_m)), \qquad m=1,..,I$$

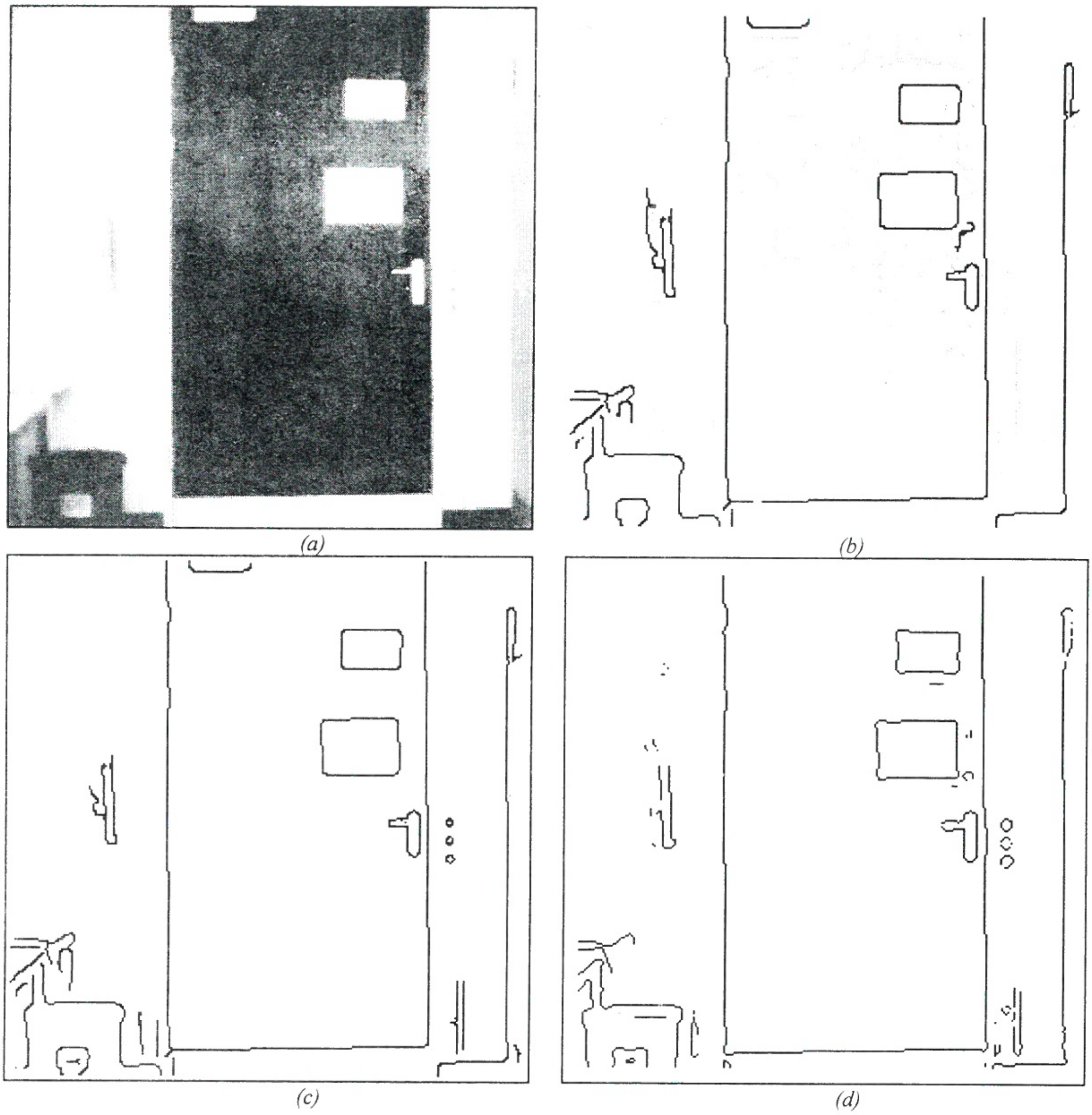

Figure 6. Comparison of Edge Extraction Methods:

(a) original test image;

(b) edge extraction by ANN selection at the pixel level (in black, selected points, in grey, rejected ones);

(c) edge extraction by ANN regulation of the hysteresis thresholds;

(d) Marr-Hildreth edge extraction with optimal parameters set by hand.

$$y_k=f(\Sigma_m (V_{nj}\, h_n)), \qquad n=1,\cdot\cdot,K$$

with $\quad i=1,\cdot\cdot,I$ *and* $k=1,\cdot\cdot,K$

The ANN is trained on a collection of input-output patterns (x_p - t_p, $p=1,\cdot\cdot,P$), P being the number of training patterns, i.e. in our case, it is the number of *PES* points for all the training images. The Error Back Propagation training strategy is briefly described in the following.

A pattern x_p is presented at the bottom layer of the network and the output y_p is produced. The achieved output is then compared with the target t_p and the discrepancies are collected into a single scalar value using a cost function.

$$E_p= (1/2)\, |y_p - t_p|^2 = E_p([U], [V])$$

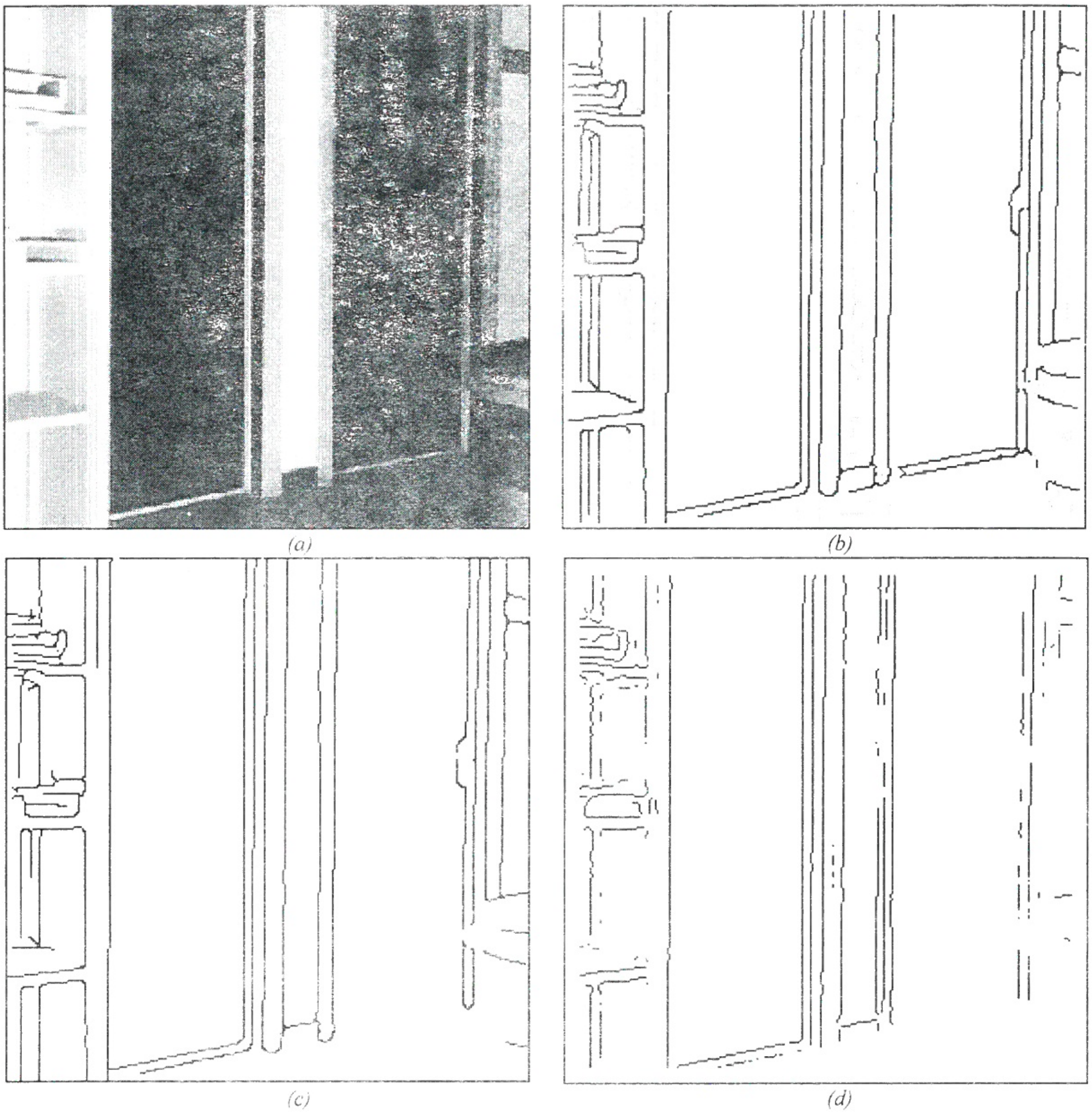

Figure 7. Comparison of Edge Extraction Methods:

(a) original test image;

(b) edge extraction by ANN selection at the pixel level (in black, selected points, in grey, rejected ones);

(c) edge extraction by ANN regulation of the hysteresis thresholds;

(d) Marr-Hildreth edge extraction with optimal parameters set by hand.

$$E = \sum_p E_p = E([U], [V]), \quad p=1,\cdots,P.$$

The contribution of each input-target pattern is considered and the global error function E is computed. E is regarded as a multi-dimensional potential surface which depends on the weights $[U]$ and $[V]$. Given the training set, the weights that minimize the global error E are computed using the steepest-descent algorithm. The iterative updating rules for the entries of the input and output weight matrices ($\boldsymbol{U}_{jk}$ and $\boldsymbol{V}_{ij}$, with $i=1,..,I;\ j=1,..,J;\ k=1,..,K$) are:

$$V_{jk}(t) = -\eta\, \Delta V_{jk}(t) + \alpha\, \Delta V_{jk}(t-1),$$

where:

$$\Delta V_{jk}(t) = \Sigma_p \, [\, (y_{k,p}-t_{k,p}\,) \; h_{j,p} \; (1-y^2_{k,p}) \,], \quad p=1,\cdot\cdot,P$$

$$U_{ij}(t) = -\eta \, \Delta U_{ij}(t) + \alpha \, \Delta U_{ij}(t-1),$$

where:

$$\Delta U_{ij}(t) = \Sigma_p \, [\, \delta_{j,p} \, (1-h^2_{j,p}) \, x_{i,p} \,], \quad p=1,\cdot\cdot,P$$

$$\delta_{j,p} = \Sigma_k \, [\, (y_{k,p}-t_{k,p}\,) \, (1-y^2_{k,p}) \; V_{jk} \,], \; k=1,\cdot\cdot,K$$

and where t stands for the iteration index.

In the above equations, the coefficient η is the speed gain and α is the coefficient of the momentum of inertia, used to avoid that the algorithm may converge to local minima of the energy function. Moreover, to speed up convergence, the following heuristic criteria are applied:

- The weights are updated according to the above equations.
- If the global error at iteration (t) turns out to be smaller than the previous one at iteration (t-1), then $\eta=\rho\eta$, with $\rho>>1$, and the weights are updated.
- Otherwise, if the error increases, $\eta=\varepsilon\eta$, with $\varepsilon<<1$, $\alpha=0$ and the current iteration is discarded.

4. Results and Conclusions

The proposed method for extracting edges classifies points directly. The training set can be obtained by using just a few images, since from each image we can extract a large *PES*. Thus, a reduced number of 15 images is enough for representing several different environmental conditions.

The performances of the developed method have been computed both in relation to the classification task and in terms of the visual quality of the extracted edges. The ANN can correctly detect 95.2% of the edge points of a test set of ten images, thus revealing good results of the classification task.

Figure 6 and Figure 7 show some visual results and comparisons with other edge-extraction methods. First, the comparison is made with the ANN regulation of the hysteresis thresholds developed in [Ac95], which performs as the classical Canny edge-extraction method, followed by the hysteresis selection with thresholds set optimally by hand. The direct pixel selection performs a little better, since small unwanted edges (e.g. near the handle of the door and at the bottom of the door in Figure 6) are not extracted and important edges missed by the previous method are here detected (the bottom of the door and the basket in Figure 7). Then, the comparison is made with the Marr-Hildreth edge extraction, regulated by hand with the best threshold for avoiding noisy edges.

In conclusion, the ANN-driven direct point selection solves the problem of the human intervention for achieving an optimal edge extraction, and makes it possible to use edge analysis in autonomous systems. Even though this method has been applied to the Canny algorithm, it is able to set ad hoc parameters for those algorithms that use a filter for the enhancement of edge points, followed by a selection for removing false points. Moreover, the selection criterion exploits local features of an image and outperforms even optimally tuned classical algorithms.

Acknowledgment

This work was partially funded within the ESPRIT BR Project EP7274 "Behavioural Learning: Combining Sensing and Action".

REFERENCES

1. MARR, D., **Vision: A Computational Investigation into the Human Representation and Processing of Visual Information**, W.H. FREEMAN, San Francisco, 1982.

2. MARR, D. and HILDRETH, E., **Theory of Edge Detection**, APPLIED OPTICS, Vol. 16, 1980, pp.145-148.

3. HARALICK, R.M., **Digital Step Edges from Zero Crossing of Second Directional Derivatives**, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , Vol. 6, No. 1, 1984.

4. CANNY, J., **A Computational Approach to Edge Detection**, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol. 8, No. 6, November 1986, pp.679-698.

5. HONG, T.H., SCHNEIDER, M.O., and ROSENFELD, O., **Border Extraction Using Linked Edge Pyramids**, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, Vol. 12, No. 5, 1982.

6. SCHNEIDER, M.O., **Extracting Linear Features from Images Using Pyramids**, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, Vol. 12, No. 4, 1982.

7. VOORHEES, H. and POGGIO, T., **Detecting Textons and Texture Boundaries in Natural Images**, Proceedings of the First International Conference on Computer Vision, 1987, pp.250-258.

8. HANCOCK, E.R. and KITTLER, J., **Adaptive Estimation of Hysteresis Thresholds**, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991, pp.196-201.

9. ACCAME, M. and DE NATALE, F.G.B., **Neural Tuned Edge Extraction in Visual Sensing**, Proceedings of the Third European Workshop on Learning Robots, Santorini, Greece, 1995.

10. MONETA, C. and DE NATALE, F.G.B., **Adaptive Control in Visual Sensing**, Proceedings of the IMACS International Symposium on Signal Processing, Robotics, and Neural Networks, 1994.

11. KAISER, M., KLINGSPOR, V., MILLÁN, J.d.R., ACCAME, M., WALLNER, F., and DILLMANN, R., **Using Machine Learning Techniques in Real-World Mobile Robots**, IEEE EXPERT, Vol. 10, No. 2, April 1995.

12. ACCAME, M. and DE NATALE, F.G.B., **Intelligent Visual Sensing System for Autonomous Applications**, in Tascini, Esposito, Vito, Zingaretti (Eds.) Machine Learning and Perception, WORLD SCIENTIFIC PUBLISHING, Singapore, 1996.

13. RUMELHART, D.E., HINTON, G.E. and WILLIAMS, R.J., **Learning Internal Representations by Error Back Propagation**, in D.E. Rumelhart and J.L. McClelland (Eds.) Parallel Distributed Processing: Exploration in the Microstructure of Cognition, (Chap.8), Cambridge, MA, MIT PRESS, 1986.

14. VOGL, T.P., MANGIS, J.K., RIGLER, A.K., ZINK, W.T. and ALKON, D.L., **Accelerating the Convergence of the Back-Propagation Method**, BIOLOGICAL CYBERNETICS, Vol. 59, 1988, pp.257-263.