Proceedings of the V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 481–486, April 2–5, 2001 - Rio de Janeiro - RJ - Brazil

Fuzzy Inference System Applied to Edge Detection in Digital Images

Cristiano Jacques Miosso[1], Adolfo Bauchspiess[2]
Departamento de Engenharia Elétrica
Faculdade de Tecnologia
Universidade de Brasília – UnB
[1]miosso@engineer.com
[2]bauchspiess@ene.unb.br

Abstract

First-order linear filters constitute the algorithms most widely applied to edge detection in digital images. Nevertheless they don't allow good results to be obtained from images where the contrast varies a lot, due to non-uniform lighting, as it happens during acquisition of most part of natural images. In this paper, we evaluate the performance of a fuzzy inference system in edge detection. The results for images with high contrast variation are compared to those obtained with the linear Sobel operator.

1. Introduction

The goal of the edge detection process in a digital image is to determine the frontiers of all represented objects, based on automatic processing of the color or gray level information in each present pixel. This procedure has many applications in image processing and computer vision, and is an indispensable technique in both biological and robot vision [1]. In [2], for example, the output of an edge detector filter is one of the inputs of a fuzzy system applied to contrast enhancing in images and to their segmentation according to the various present classes. In [3], the same technique is used as a decision element in computing sparse correspondence maps between stereo images.

Usage of specific linear time-invariant (LTI) filters is the most common procedure applied to the edge detection problem, and the one which results in the least computational effort. In the case of first-order filters, an edge is interpreted as an abrupt variation in gray level between two neighbor pixels. The goal in this case is to determine in which points in the image the first derivative of the gray level as a function of position is of high magnitude. Figure 1 depicts the applied process, given an input image I. Two filters with impulse responses h_{DX} and h_{DY} are used to estimate derivatives in both horizontal (x) and vertical (y) directions; points where filtered images I_{DX} and I_{DY} have values above an specified threshold are associated with vertical and horizontal edges, respectively. By computing norm-2 ($\|x, y\| = \sqrt{x^2 + y^2}$) in each pixel of I_{DX} and I_{DY} and applying the threshold to the new output image, edges in arbitrary directions are detected.

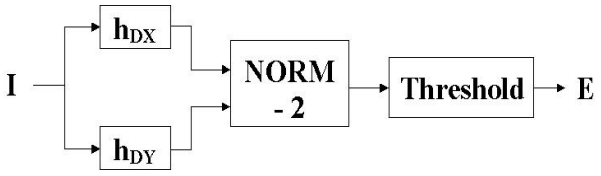


Figure 1: Edge detection system using LTI filters to estimate the image's derivatives in x (h_{DX}) and y (h_{DY}) directions.

The threshold value to be applied may be estimated given the root mean square value (RMS) associated to the input image [4, page 2-51]. Filters h_{DX} and h_{DY} are usually kernels with 5x5 or, most often, 3x3 elements. The most used of them are those by Sobel (see Section 3), Prewitt and Roberts.

Despite the structure shown in Figure 1 being the most frequently used in edge detection, we observe that it does not give rise to good results when applied to images with non-uniform lighting, as it is the case of most part of natural images. In this situation, choosing a threshold value which is appropriate to regions with high contrast does not allow edges to be detected in regions with low contrast. Adoption of a smaller threshold value so as to detected edges in low contrast results, by the other hand, in including in the output binary image points which are not part of edges, in regions where gray level fluctuations are higher even outside the objects' frontiers.

Second order operators are sometimes used to minimize this problem. In this situation, the LTI filters are employed to estimate the second derivative of the gray level in the image, instead of the h_{DX} and h_{DY} filters. The edges detected by this procedure are formed by all points where this second derivative is null, or near zero according to a noise tolerance previously established. The most commonly used filter in this class is the so-called LoG (*Laplacian-of-Gaussian*); besides the greater computational complexity drawback,

this operator also has the particularity of generating an uninterrupted line to represent all edges in the input image, and is not adequate then to represent more general structures.

A more robust and general solution to the problem is presented in [5]. A recurring neural network with three outputs for each pixel of the original image is employed. Two of these outputs, h and v, represent discontinuities in horizontal and vertical directions, respectively; the third output, f, will be 1 or 0 after an iterative process, indicating if the corresponding pixel is or is not contained in an edge. An energy function is also defined and associated to all outputs of the network; this function, initially proposed by Christof Koch, is of greater value the less inadequate the values h, v and f are as representations of the real edges of I. The neural network's parameters are then altered in each iteration in such a way as to reduce the energy function value.

This approach results however in great computational effort. For an image with MxN pixels, a neural network with 3xMxN neurons is generated (for an image with 512x512 pixels, there will be 786432 neurons). Besides, the images' edges are not detected before an iterative process, in which all outputs of all these neurons are altered, 50 to 70 training epochs being typically necessary to attaining convergence [1].

2. General Description of the Proposed System

A non-linear image filtering technique is presented in [2] which is based in fuzzy inference systems (FIS). First, an input image is processed in different non-successive linear filtering stages, which means that the input to each filter is always the original image. The gray level in each pixel of the resulting image is then obtained by applying the FIS system to the corresponding values in the output images of the linear operators, in the same pixel. The adopted fuzzy rules and the fuzzy membership functions are specified according to the kind of filtering to be executed.

In the system described in [2], all inputs to the the FIS system are obtained by applying to the original image a high-pass filter, a first-order edge detector filter (Sobel operator) and a low-pass (mean) filter. The whole structure is then tuned to function as a contrast enhancing filter and, in another problem, to segment images in a specified number of input classes.

In this research, we evaluate the efficiency of a FIS system applied to the edge detection problem. During input image pre-processing, three kinds of linear filters are applied to it: Sobel operators, used to estimate its derivatives in horizontal and vertical directions (h_{DH} and h_{DV} filters), a low-pass (mean) filter and a high-pass filter. The key difference between this approach and that proposed in [2] is that here the gray level associate to pixel (i, j) in the output image E depends not only on the pixel (i, j) in each pre-processed image but also on some neighbor pixels, as it is depicted in Figure 2. Besides, each image DH and DV that results from applying Sobel operators is passed to the FIS system, and not only the image composition $D = \sqrt{DH^2 + DV^2}$. In Section 3, where the adopted fuzzy rules are exposed, the relevance of each of the inputs employed to compute output image will become clear.

The developed fuzzy system's purpose is to determine if pixel (i, j) evaluated is or is not present in one of image's edges, given the information explicit in the input filtered images. In the first case, and only in it, the ouptut of the FIS system must be high, the reference being the fuzzy sets associated to this output. This justifies the usage of a threshold in the binarization stage, when computing $B(i, j)$.

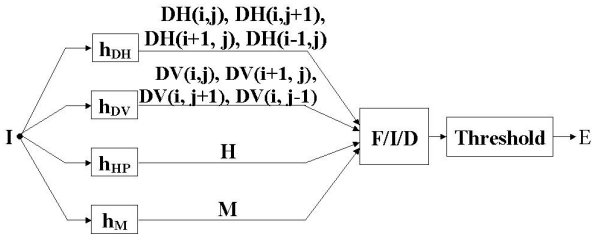


Figure 2: Fuzzy inference system (FIS) applied to edge (E) detection in image I. h_{DH} and h_{DV} are the Sobel operators used to estimate the first derivative of I in horizontal and vertical directions. h_{HP} and h_M are the kernels of a high-pass and a low-pass (mean) filters. F, I and D refer to the *fuzzification*, *inference* and *deffuzification* stages during usage of the FIS system, respectively.

3. Implementation of the FIS System Applied to Edge Detection

3.1. Image Pre-Processing

During input image pre-processing stage, 4 linear filters were employed, as shown inFigure 2. Sobel operators h_{DH} and h_{DV} are kernels with 3x3 elements given by [3, page 53]:

$$h_{DH} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (1)$$

$$h_{DV} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

As a high-pass filter, we adopted also a 3x3 kernel, given by:

$$h_{HP} = \begin{bmatrix} -1/16 & -1/8 & -1/16 \\ -1/8 & 3/4 & -1/8 \\ -1/16 & -1/8 & -1/16 \end{bmatrix} \quad (3)$$

Figure 3-(a) shows the magnitude of the frequency response of the filter described by equation (3). Notice that this magnitude is higher for higher frequencies, which means that the chosen filter presents the desired behavior.

Filter MF, in turn, was chosen in such a way as to guarantee that the gray level in each pixel of the output image is the arithmetic mean of the gray levels in a 5x5 neighborhood of the same pixel in the input image. In that manner:

$$h_{MF} = \frac{1}{25} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4}$$

The magnitude of the frequency response of this filter is depicted in Figure 3-(b). In this case, this magnitude is higher for smaller frequencies, which means that the filter has a low-pass behavior.

Given the kernels associated with each filter, the filtered images may be computed through a bidimensional convolution operation:

$$DH = h_{DH} * I \tag{5}$$

$$DV = h_{DV} * I \tag{6}$$

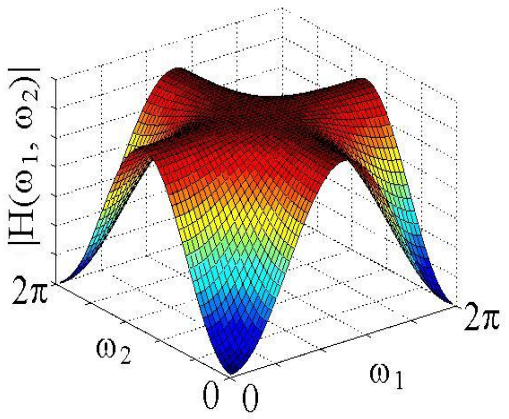
$$HP = h_{HP} * I \tag{7}$$

$$M = h_{MF} * I \tag{8}$$

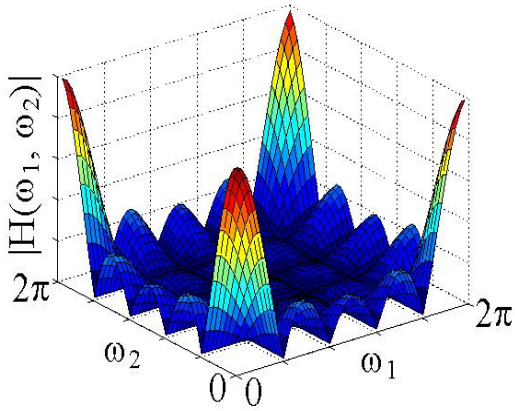
3.2. Fuzzy Sets and Fuzzy Membership Functions Definitions

The system implementation was carried out considering that the input image I and the output image obtained afer *defuzzification* are both 8-bit quantized; this way, their gray levels are always between 0 and 255. These values define the working interval of the output variable and the input variable M (the other input variables are not guaranteed to be less than 255). Besides, three fuzzy sets were created to represent each variable's intensities; these sets were associated to the linguistic variables "low", "medium" and "high".

The adopted membership functions for the fuzzy sets associated to the input M and to the output were Gaussian functions with means 0, 127.5 and 255, as shown in Figure 4-(a). For the sets associated to the other input images, Gaussian functions were also adopted for the linguistic variables "low" and "medium", but for the variable "high" a sigmoid function was chosen (Figure 4-(b)), since in this case we can not guarantee that the input values will be restricted to the interval $[0, 255]$.



(a)



(b)

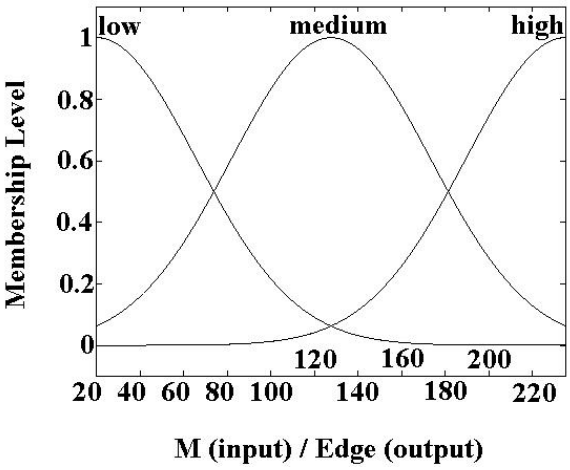
Figure 3: Discrete-time Fourier Transform of the high-pass (a) and the low-pass (mean) filter (b) adopted, in magnitude.

3.3. Fuzzy Logical Operations and Defuzzification Method Definitions

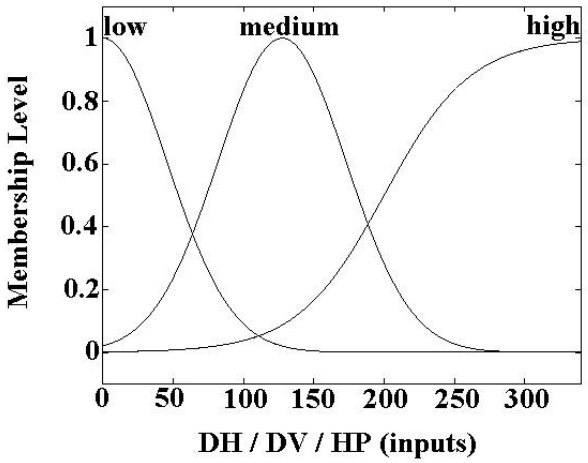
The functions adopted to implement the "and" (norm-T) and "or" (norm-S) operations were the minimum and maximum functions, respectively. The Mamdani method was chosen as the defuzzification procedure, which means that the fuzzy sets obtained by applying each inference rule to the input data were joined through the add function; the output of the system was then computed as the centroid of the resulting membership function [6, pages 2-20 to 2-23].

3.4. Inference Rules Definitions

As it was mentioned in Section 2, the fuzzy inference rules were defined in such a way that the FIS system



(a)



(b)

Figure 4: Membership functions of the fuzzy sets associated to the output and to the input M (a) and to inputs DH, DV, HP (b).

output ("Edges") is high only for those pixels belonging to edges in the input image. A robusteness to contrast and lighting variations were also in mind when these rules were established.

The first 3 rules were defined to represent the general notion that in pixels belonging to an edge there is a high variation of gray level in the vertical or horizontal direction:

1. (DH low) and (DV low) → ("Edges" low).
2. (DH medium) and (DV medium) → ("Edges" high).
3. (DH high) or (DV high) → ("Edges" high).

To guarantee that edges in regions of relatively low contrast can be detected, the two following rules were established to favour medium variations of the gray level in a specific direction in regions of low frequency of the input image (HP "low"):

4. (DH medium) and (HP low) → ("Edges" high).
5. (DV medium) and (HP low) → ("Edges" high).

Rules 6 and 7 were chosen in such a way as to avoid including in the output image pixels belonging to regions of the input were the mean gray level is lower. These regions are proportionally more affected by noise, supposed it is uniformly distributed over the whole image. The goal here is to design a system which makes it easier to include edges in low contrast regions, but which does not favour false edges by effect of noise.

6. (DV medium) and (M low) → ("Edges" low).
7. (DH medium) and (M low) → ("Edges" low).

Rules 8 to 11 were established to avoid forming double edges in the output image (they tend to appear due to shadows in the natural images). Considering that high variations in gray level in horizontal direction correspond to vertical edges, we conclude that high values of $DH(i,j)$ and $DH(i,j \pm 1)$ do not imply edge pixels in (i,j) and $(i,j \pm 1)$, simultaneously. Analagously, high values of $DV(i,j)$ and $DV(i \pm 1,j)$ do not correspond to edge pixels in (i,j) and $(i \pm 1,j)$.

8. (DV high) and ($DV(i+1,j)$ high) → ("Edges" medium).
9. (DH high) and ($DH(i,j+1)$ high) → ("Edges" medium).
10. (DV medium) and ($DV(i+1,j)$ high) → ("Edges" low).
11. (DH medium) and ($DH(i,j+1)$ high) → ("Edges" low).

Finally, rule 12 was defined to avoid including isolated pixels in the output image, favouring only continuous lines. It also avoids including points by effect of noise, since this tends to generate isolated pixels in the image which represents the input's edges.

12. ($DV(i,j+1)$ low) and ($DH(i+1,j)$ low) and ($DV(i,j-1)$ low) and ($DH(i-1,j)$ low) → ("Edges" low).

4. Results

The system described in Section 3 was tested with different images, its performance being compared to that of the Sobel operator. The weights associated with each fuzzy rule were tuned to allow good results to be obtained while extracting edges of the image shown in Figure 5-(a), where contrast varies a lot from one region

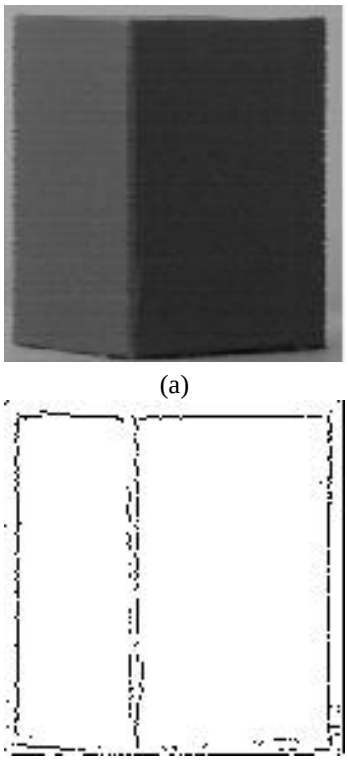


Figure 5: (a) Image used to tune the weights associated to the fuzzy inference rules, in the developed system. (b) Edges detected by the FIS system.

to another. During the performance tests, however, all parameters were kept constant.

Figure 6 depicts the image of a digital cameras calibration pattern [3, pages 20 to 27] in which there is a high contrast variation. We observe, in (a), that the Sobel operator with threshold automatically estimated from image's RMS value does not allow edges to be detected in the regions of low contrast. When this threshold is reduced to 0.012, so that these edges are included in the output image, we observe that pixels not belonging to edges are also included, mainly in the regions of low gray levels (b). The FIS system, in turn, allows edges to be detected even in the low contrast regions without the output image being affected by noise (c). This is due to the different treatment given by the fuzzy rules to the regions with different contrast levels, and to the rule established to avoid including in the output image pixels not belonging to continuous lines.

Figure 7-(a) in turn shows a chess pattern with gray levels gradation, which simulates a lighting gradient over the image. We notice that Sobel operator again is not able to detect edges in regions with lower contrast (b), due to indistinct treatment to regions with different levels of contrast. By applying the FIS system, all present edges are detected (c). This same example is also present in [5], where the employed neural network also allows all edges in Figure 7-(a) to be detected correctly, but resulting in greater computational effort.

In Figure 8, a synthetic image of a cube with its edges detached in black is shown. When Sobel operator is applied to this image, two abrupt variations in gray level are perceived in the frontier between each two faces (one between the face and the detached edge, other between this edge and the neighbor face), which results in two edges being detected (double edges) (b). The adoption of fuzzy rules specifically established to avoid double edges results in obtaining an image with single edges when the FIS system is applied to the same image (c).

5. Conclusion

These results allow us to conclude that despite the much superior computational effort when compared to the Sobel operator, the implemented FIS system presents greater robusteness to contrast and lighting variations, besides avoiding obtaining double edges. Further tuning of the weights associated to the fuzzy inference rules is still necessary to reduce even more inclusion in the output image of pixels not belonging to edges, as observed in Figures 5 and 6.

References

[1] Md. Shoiab Bhuiyan, Yuiji Iwahori, and Akira Iwata. Optimal edge detection under difficult imaging conditions. Technical report, Educational Center for Information Processing and Dept. of Electrical and Computer Engineering, Nagoya Institute of Technology, Showa, Nagoya, 466-8555, JAPAN, URL:-http://www.center.nitech.ac.jp/people/bhuiyan /pub.html.

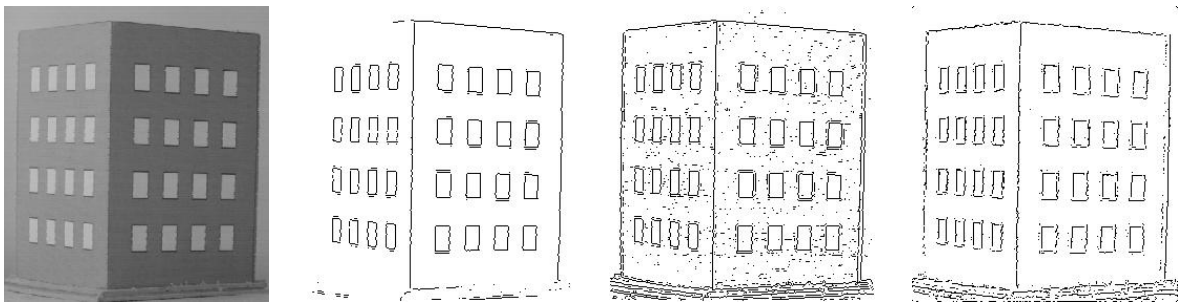
[2] E. O. Salles and L. L. Ling. Uma aplicação de sistemas nebulosos em processamento de imagens. In *Anais do 3º Simpósio Brasileiro de Automação Inteligente*, Vitória - ES, September 1997. Universidade Federal do Espírito Santo e Sociedade Brasileira de Automática.

[3] Cristiano Jacques Miosso and Mirele de Almeida Mencari. *Visão Computacional Estéreo Aplicada a Formas Poliédricas*. March 1999. Projeto de Graduação.

[4] Clay M. Thompson and Loren Shure. *Image Processing Toolbox User's Guide - For Use With MatLab*. The MathWorks, Inc., January 1995.

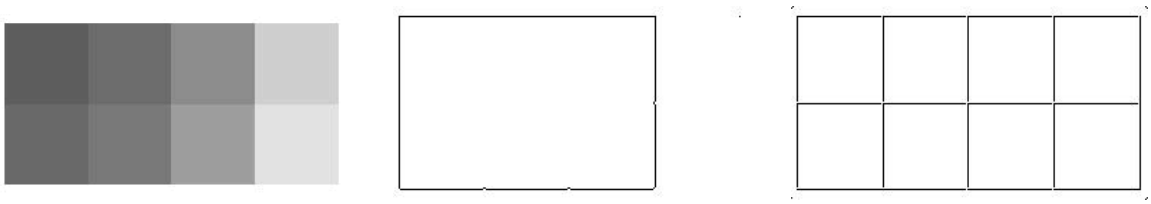
[5] Md. Shoaib Bhuiyan, Hiroshi Matsuo, Akira Iwata, Hideo Fujimoto, and Makoto Sato. An improved neural network based edge detection method. Technical report, Dept. of Electrical and Computer Engineering and Dept. of Mechanical Engineering, Nagoya Institute of Technology, Nagoya, JAPAN 466., bhuiyan@mars.elcom.nitech.ac.jp.

[6] J. S. Roger Jang and Ned Gulley. *Fuzzy Logic Toolbox User's Guide*. The MathWorks, Inc., January 1995.



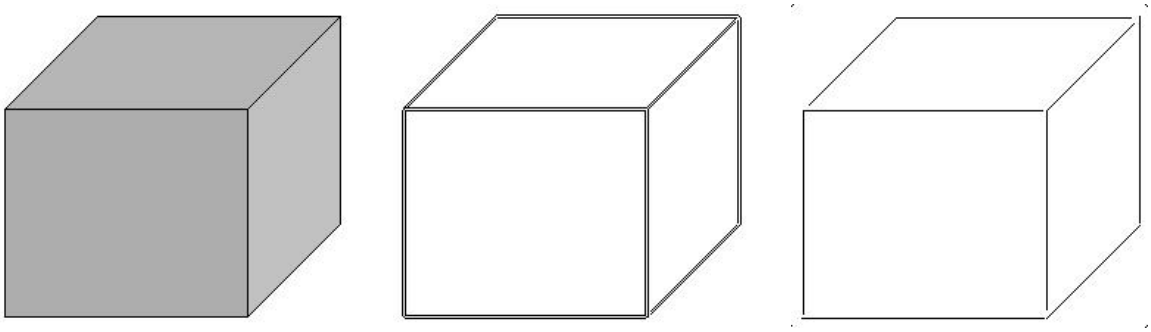
(a) (b) (c) (d)

Figure 6: (a) A digital cameras calibration pattern's image. (b) Edges detected by the Sobel operator (threshold computed based on image's RMS value). (c) Edges detected by the Sobel operator (threshold equaled to 0.012 to include edge pixels not represented in b). (d) Edges detected by the studied FIS system.



(a) (b) (c)

Figure 7: (a) Image of a chess pattern with gradation of gray levels. (b) Edges detected by the Sobel operator (threshold computed based on image's RMS value). (c) Edges detected by the studied FIS system.



(a) (b) (c)

Figure 8: (a) Synthetic image of a cube with all edges detached. (b) Edges detected by the Sobel operator (threshold computed based on image's RMS value). (c) Edges detected by the studied FIS system.