Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería

MÉTODO DE DETECCIÓN DE BORDES POR MEDIO DE LÓGICA DIFUSA TIPO-2 GENERALIZADA

TESIS

Que para obtener el grado de Doctor en Ciencias

Presenta:

M.C. Claudia Ibeth González Berrelleza

Bajo la dirección de:
Dra. Elba Patricia Melin Olmeda

Co-dirigido por:
Dr. Juan Ramón Castro Rodríguez

Tijuana, Baja California

Agosto 2016

Autonomous University of Baja California

Faculty of Chemical Sciences and Engineering

Masters and Doctorate in Science and Engineering

EDGE DETECTION METHOD BASED ON GENERALIZED TYPE-2 FUZZY LOGIC

THESIS

That to obtain the degree of Doctor in Sciences

Presented by:

M.Sc. Claudia Ibeth González Berrelleza

Under the direction of:
Dr. Elba Patricia Melin Olmeda

Co-directed by:
Dr. Juan Ramón Castro Rodríguez

Tijuana, Baja California

August 2016

Acknowledgements

I would like to thanks God to give me the strength, dedication and perseverance that were the engine to reach this stage of my professional improvement. To my parents and brothers for give me the motivation to go ahead. Especially grateful with my husband Edgar for their love, patience and unconditional support in last years to successfully conclude my PhD.

I would like to extend my gratitude to my thesis director, Dr. Patricia Melin who guide me with their experience and dedication to take the best way to achieve my goals and objectives. To my thesis co-director Dr. Juan Ramón Castro, for share their knowledge, experience, show me new challenges and help me to overcome technical difficulties derived by this work. To Dr. Oscar Castillo, thank you for being an excellent guidance, with their vision and experience in investigation area it helped to realize some publications. To Dr. Olivia Mendoza and Dr. Antonio Rodríguez for being part of my thesis commission and for count with their advice.

Finally, I would to thank you to the university UABC for take care of me the last years and feel proud to be graduated of an excellent university. Thanks to CONACYT for the economic support to study for four years.

Abstract

Edge detectors have traditionally been an essential part of many computer vision systems. There are different methods that have been proposed for improving edge detection in real images. This thesis presents a new edge detection method based on generalized type-2 fuzzy systems (GT2 FSs) which allows better modeling of the uncertainty that exists in processing digital images.

In this thesis four new methods are proposed. In the first method the generalized type-2 fuzzy logic is combined with the morphological gradient technique. The second method combines the GT2 FSs and the Sobel operator; in the third approach the methodology based on Sobel operator and GT2 FSs is improved to be applied on color images. In the fourth approach, we proposed a novel edge detection method where, a digital image is converted a generalized type-2 fuzzy image. In the contribution is also included a comparative study of type-1, interval type-2 and generalized type-2 fuzzy systems as tools to enhance edge detection in digital images when used in conjunction with the morphological gradient and the Sobel operator. The proposed generalized type-2 fuzzy edge detection methods were tested with benchmark images and synthetic images, in a grayscale and color format.

Another contribution is that the generalized type-2 fuzzy edge detector method is applied in the preprocessing phase of a face recognition system; where the recognition system is based on a monolithic neural network. The aim of this part of the work is to show the advantage of using a generalized type-2 fuzzy edge detector in pattern recognition applications. Additionally, make a comparative analysis with the recognition rates obtained by the generalized type-2 against the results achieved by type-1 and interval type-2 fuzzy edge detectors.

The main goal of using generalized type-2 fuzzy logic in edge detection applications is to

provide them with the ability to handle uncertainty in processing real world images; otherwise, to demonstrate that a GT2 FLS has a better performance than the edge detection methods based on type-1 and type-2 fuzzy logic systems.

Resumen

Los detectores de bordes han sido tradicionalmente una parte esencial de muchos sistemas de visión por computadora. Existen diferentes métodos que se han propuesto para mejorar la detección de bordes en imágenes reales. Esta tesis presenta un nuevo método de detección de bordes basado en lógica difusa tipo-2 generalizada el cual pueda permitir un mejor modelado de la incertidumbre que pudiera existir en el procesamiento digital de imágenes.

En esta tesis se proponen cuatro métodos nuevos. En el primer método, el sistema difuso tipo-2 generalizado es combinado con la técnica de gradiente morfológico. El segundo método combina un sistema difuso tipo-2 generalizado con el operador de Sobel; en la tercera propuesta la metodología basada en el operador Sobel y sistemas difusos tipo-2 generalizado es mejorada para ser aplicada en imágenes a color. En el cuarto enfoque, proponemos un novedoso método de detección de bordes, donde una imagen digital es convertida a una imagen difusa tipo-2 generalizada.

En la contribución de este trabajo, también está incluido un estudio comparativo de sistemas difusos tipo-1, tipo-2 por intervalos y tipo-2 generalizado, como herramientas para mejorar la detección de bordes en imágenes digitales cuando se utiliza junto con el gradiente morfológico y el operador Sobel.

Como parte de los objetivos de esta tesis, el detector de bordes difuso tipo-2 generalizado es aplicado en la fase de pre-procesamiento de un sistema de reconocimiento de rostros; donde el sistema de reconocimiento es desarrollado con una red neuronal monolítica con el objetivo de mostrar las ventajas de usar un detector difuso en aplicaciones de reconocimiento de patrones. Adicionalmente, hacer un análisis comparativo con la tasa de reconocimiento obtenida con el

sistema difuso tipo-2 generalizado contra los resultados logrados por los detectores de bordes basados en lógica difusa tipo-1 and tipo-2.

El principal objetivo de usar lógica difusa tipo-2 generalizada en aplicaciones de detección de bordes es proporcionar a los sistemas la capacidad de manejar la incertidumbre en el procesamiento de imágenes del mundo real; por otro lado, demostrar que un sistema de lógica difusa tipo-2 generalizada tiene un mejor funcionamiento que los detectores de bordes basados en lógica difusa tipo-1 y tipo-2 por intervalos.

Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 178

Tijuana, B. C., a 05 de agosto de 2016

C. Claudia Ibeth González Berrelleza
Pasante de: Doctor en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la Opción TESIS

Es propuesto, por los CC. Dres. Elba Patricia Melin Olmeda y Juan Ramón Castro Rodríguez

Quienes serán los responsables de la calidad del trabajo que usted presente, referido al tema: MÉTODO DE DETECCIÓN DE BORDES POR MEDIO DE LÓGICA DIFUSA TIPO-2 GENERALIZADA

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

I.- INTRODUCCIÓN
II.- LOGICA DIFUSA TIPO-2 GENERALIZADA
III.- METODOS DE DETECCION DE BORDES Y FILTROS USADOS EN PROCESAMIENTO DIGITAL DE IMAGENES
IV.- METRICAS PARA METODOS DE DETECCION DE BORDES
V.- METODOS DE DETECCION DE BORDES BASADOS EN SISTEMA DE LOGICA DIFUSA TIPO-2 GENERALIZADA
VI.- DETECTOR DE BORDES DIFUSO TIPO -2 GENERALIZADO APLICADO A UN SISTEMA DE RECONOCIMIENTO
VII.- EXPERIMENTACION Y DISCUSION DE RESULTADOS
VIII.- CONCLUSIONES Y TRABAJO FUTURO
IX.- REFERENCIAS

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS QUIMICAS E INGENIERIA

Dra. Elba Patricia Melin Olmeda
Directora de Tesis

Dr. José Luis González Vázquez
Secretario

Dr. Juan Ramón Castro Rodríguez
Co-Director de Tesis

Dr. Luis Enrique Palafox Maestre
Director

Table of contents

Figures index

Tables index

1. Introduction

An edge may be the result of changes in light absorption, color, shade and texture, and these changes can be used to determine the depth, size, orientation and surface properties of a digital image [1]. In digitally analyzing the image, edge detection is a process that involves filtering irrelevant information to select the edge points. The detection of subtle changes may be complicated by noise and this depends on the threshold for pixel change that defines an edge. Detection of these continuous edges is very difficult and time consuming especially when an image is corrupted by noise [2].

Edge detectors have been an essential part of many computer vision systems. The edge detection process is useful to simplify the analysis of images by drastically reducing the amount of dataset to be processed [3]. The main application areas of edge detectors include: geographical systems, military, medicine, robotics, meteorology and pattern recognition systems [4-8].

In the area of image processing there exist some edge detection methods that are based on type-1 fuzzy systems [9-11], neural networks [12], genetic algorithms with PSO (Particle Swarm Optimization) [2], Ant Colony Optimization (ACO) [13-14], interval-valued fuzzy operators [15], interval type-2 fuzzy systems combined with the Sobel operator [16-17], interval type-2 fuzzy systems and the morphological gradient [18] and an improved Canny method based on interval type-2 fuzzy logic [19]. Of course, we can also find the traditional methods for image processing, like the Canny [3], Morphological Gradient [20], Sobel [21], Roberts, Kirsch [22] and Prewitt [23].

The motivation to use fuzzy logic in the image processing area; more specifically in edge detection is because one of the most common problems encountered during the acquisition and

transmission of digital images is the contamination of the image with impulse noise [24-25]. Due to this inevitable fact, the acquired or received digital image usually must be pre-processed by applying an appropriate filtering operator to the image to suppress the contaminating noise [26-27].

In the last years great progress has been made in transitioning from traditional type-1 fuzzy systems (T1 FSs) [28-30] to interval type-2 fuzzy systems (IT2 FSs) [31-35] and, most recently, to generalized type-2 fuzzy systems (GT2 FSs) [36-43]. Of course, the idea of going into higher orders or types of fuzzy logic is to achieve better models of uncertainty. In this sense, it is theoretically expected that using IT2 FSs can provide better performance for a fuzzy system than using T1 FSs. Therefore, the GT2 FSs, have the potential to provide better performance than the IT2 FSs, which is why there has been so much interest in interval type-2 and generalized type-2 fuzzy systems. However, generalized type 2 requires a higher computational overhead and several efforts have been made in order to limit the complexity of generalized type-2 fuzzy logic; for example, Wagner and Hagras [36-37] have introduced the zSlices-based representation and Mendel and Liu [38-40], have put forward a representation based on α-planes, which both enable the representation of, and computation with, generalized type-2 fuzzy sets.

The main contribution of this thesis is to propose a new edge detection method based on generalized type-2 fuzzy systems, which allows better modeling of the uncertainty that exists in processing digital images.

In this thesis four new methods are proposed. In the first method the generalized type-2 fuzzy logic is combined with the morphological gradient technique. The second method combines the GT2 FSs and Sobel operator; in the third approach the methodology based on Sobel operator and GT2 FSs is improved to be applied on color images. In the fourth approach,

we proposed a novel edge detection method where, a digital image is converted on generalized type-2 fuzzy image. In the contribution is also included a comparative study of T1, IT2 and GT2 fuzzy systems as tools to enhance edge detection in digital images when used in conjunction with the morphological gradient and the Sobel operator. The proposed generalized type-2 fuzzy edge detection methods were tested with benchmark images and synthetic images, in a grayscale and color format.

Another contribution is that the GT2 fuzzy edge detector method is applied on the preprocessing phase of a face recognition system; where the recognition system is based on a monolithic neural network. The aim of this part of the work is to show the advantage of using a GT2 fuzzy edge detector in pattern recognition applications. Additionally, make a comparative analysis with the recognition rates obtained by the GT2 against the results achieved by T1 and T2 fuzzy edge detectors.

This thesis is organized as follows. In Section 2, the basic concepts of generalized type-2 fuzzy logic and the theory of α-planes approximation are presented, which are used for the implementation of the proposed method. Section 3 describes some edge detection methods and filters as a part of the digital image processing**.** Section 4 explains the technique to evaluate the quality of the detected edges. The methodology used for developing the proposed GT2 fuzzy edge detection methods is presented in Section 5. The generalized type-2 fuzzy edge detection applied on a face recognition system is explained in Section 6. On the other hand, Section 7 presents simulation results with benchmark image databases to illustrate the advantages of the proposed generalized type-2 fuzzy edge detection methods. Finally, Section 8 offers some conclusions about the results and future work.

1.1. Research contributions

1.1.1. Conference proceedings

- "An edge detection method based on generalized type-2 fuzzy logic". Claudia I. Gonzalez, Juan R. Castro, Patricia Melin, Oscar Castillo. Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), pp. 39-44, 2013.
- "A new approach based on generalized type-2 fuzzy logic for edge detection". Claudia I. Gonzalez, Juan R. Castro, Gabriela E. Martinez, Patricia Melin, Oscar Castillo. IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pp. 424-429, 2013.
- "Cuckoo search algorithm for the optimization of type-2 fuzzy image edge detection systems". Claudia I. Gonzalez, J. R. Castro, Patricia Melin, Oscar Castillo. Congress on Evolutionary Computation (CEC), pp. 449-455, 2015.
- "Edge detection method based on Interval type-2 fuzzy systems for color images". Claudia I. Gonzalez, Juan R. Castro, Olivia Mendoza, Antonio Rodríguez-Díaz, Patricia Melin, Oscar Castillo. Fuzzy Information Processing Society (NAFIPS) held jointly with 5th World Conference on Soft Computing (WConSC), 2015.
- "General Type-2 Fuzzy edge detector applied on face recognition system using neural networks". Claudia I. Gonzalez, Juan R. Castro, Olivia Mendoza. IEEE World Congress on Computational Intelligence (IEEE WCCI), 2016.

1.1.2. Book chapters

- "Color Image Edge Detection Method Based on Interval Type-2 Fuzzy Systems". Claudia I. González, Patricia Melin, Juan R. Castro, Olivia Mendoza, Oscar Castillo. Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization, 2015.

- "Optimization by Cuckoo Search of Interval Type-2 Fuzzy Logic Systems for Edge Detection". Claudia I. Gonzalez, Juan R. Castro, Olivia Mendoza, Patricia Melin, Oscar Castillo. Recent Developments and New Direction in Soft-Computing Foundations and Applications, 2016.
- "General Type-2 Fuzzy edge detection in the preprocessing of a face recognition system". Claudia I. Gonzalez, Patricia Melin, Juan R. Castro, Olivia Mendoza, Oscar Castillo. Nature Inspired Design of Hybrid Intelligent Systems, 2016.

1.1.3. Journal articles

- "Edge-Detection Method for Image Processing Based on Generalized Type-2 Fuzzy Logic". Patricia Melin, Claudia I. Gonzalez, Juan R. Castro, Olivia Mendoza, Oscar Castillo. IEEE Transaction Fuzzy Systems, vol. 22, no. 6, pp. 1515-1525, 2014.
- "An improved sobel edge detection method based on generalized type-2 fuzzy logic". Claudia I. Gonzalez, Patricia Melin, Juan R. Castro, Olivia Mendoza, Oscar Castillo. Soft Computing, vol. 20 no.2, pp. 773-784, 2016.
- "Review of Recent Type-2 Fuzzy Controller Applications". Kevin Tai, Abdul-Rahman El-Sayed, Mohammad Biglarbegian, Claudia I. Gonzalez, Oscar Castillo, Shohel Mahmud. Algorithms, vol. 9, no. 2, pp. 1-19, 2016.
- "Optimization of interval type-2 fuzzy systems for image edge detection". Claudia I. Gonzalez, Patricia Melin, Juan R. Castro, Oscar Castillo, Olivia Mendoza. Applied Soft Computing, vol. 47, pp. 631-643, 2016.
- "A Color Edge Detection Method based on Generalized Type-2 Fuzzy Logic". Claudia I. Gonzalez, Patricia Melin, Juan R. Castro, Olivia Mendoza, Oscar Castillo. Information Sciences, 2016.

2. Generalized type-2 fuzzy logic

The main contribution of this work is based on the use of generalized type-2 fuzzy logic and the α-planes theory for achieving edge detection under uncertainty; therefore, in this section some important concepts and the methodology used to develop the fuzzy inference system are presented.

2.1. Definition of generalized type-2 fuzzy sets

A generalized type-2 fuzzy set (T2 FS), denoted by $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, where $x \epsilon X$, u $\epsilon J_x \subseteq [0,1]$ and $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$, and can be represented by eq. (2.1) [36, 38, 47, 45].

$$\tilde{A} = \left\{\left((x,u), \mu_{\tilde{A}}(x,u)\right) | \forall x \in X, \forall u \in J_x \subseteq [0,1]\right\} \tag{2.1}$$

If $\tilde{A}$ is continuous it can be denoted by eq. (2.2), where $\iint$ denotes the union for x and u;

$$\tilde{A} = \left\{ \int_{x \in X} \int_{u \in J_{x \subseteq [0,1]}^{u}} \mu_{\tilde{A}}(x,u)/(x,u) \right\} \tag{2.2}$$

In eq. (2.2), J_x is called the primary membership of x in $\tilde{A}$. At each value of x say $x = x'$, the two-dimensional (2-D) plane, whose axes are u and $\mu_{\tilde{A}}(x', u)$, is called a vertical slice of $\tilde{A}$ [45]. A secondary membership function is a vertical slice of $\mu_{\tilde{A}}(x, u)$. It is $\mu_{\tilde{A}}(x = x', u)$, for $x' \in$ X and $\forall_u \in J_{x'} \subseteq [0,1]$, and it is described in eq. (2.3), in which $0 \leq f_{x'}(u) \leq 1$.

$$\mu_{\tilde{A}}(x = x', u) = \int_{u \in J_{x'}} f_{x'}(u)/u \quad J_{x'} \subseteq [0,1] \tag{2.3}$$

Uncertainty in the primary membership of a GT2 fuzzy set $\tilde{A}$ is represented by a bounded region; therefore, the two-dimensional support of $\mu_{\tilde{A}}(x, u)$ is called footprint of uncertainty (FOU) of $\tilde{A}$ and is denoted by eq. (2.4).

$$FOU(\tilde{A}) = \{(x,u) \in X \times [0,1]\, \mu_{\tilde{A}}(x,u) > 0\} \tag{2.4}$$

$\mathrm{FOU}(\tilde{A})$ can also be expressed as the union of all primary memberships, i.e.

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x^u \tag{2.5}$$

In Figure 2.1 we can find a representation of a generalized type-2 membership function, and in Figure 2.2, the footprint of uncertainty (FOU) is illustrated, which is associated with the third dimension and allows a better modeling of real world uncertainty.

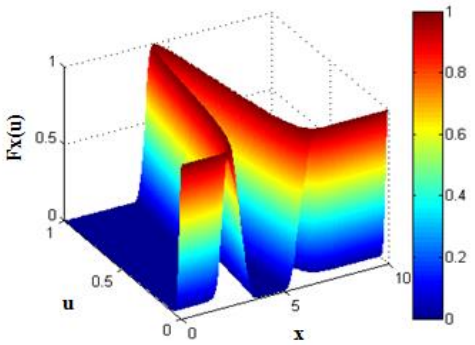


Figure 2.1 Generalized type-2 membership function

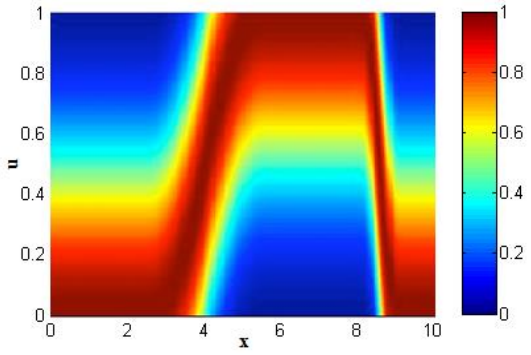


Figure 2.2 FOU of the generalized type-2 membership function

2.2. α-planes representation

In order to limit the complexity of generalized type-2 fuzzy logic several efforts have been proposed; for example, Wagner and Hagras [36-37] have introduced the zSlices representation and Mendel and Liu [38-39], have put forward a representation based on α-planes. These approximation techniques decompose the three-dimensional GT2 membership function by using different kinds of cuts to obtain a collection of IT2 FSs.

An α-plane for the GT2 FS $\tilde{A}$, denoted by $\tilde{A}_\alpha$, is the union of all primary memberships functions of $\tilde{A}$ whose secondary grades are greater than or equal to α $(0 \leq \alpha \leq 1)$ [45]. The equation of the α-plane is represented by eq. (2.6).

$$\begin{aligned} \tilde{A}_\alpha &= \{(x,u), \mu_{\tilde{A}}(x,u) \geq \alpha | \forall x \in X, \forall u \in \ \forall u \subseteq [0,1]\} \\ &= \int_{\forall x \in X} \int_{\forall u \in [0,1]} \{(x,u) | f_x(u) \geq \alpha\} \end{aligned} \tag{2.6}$$

The union of all α-planes is expressed in eq. (2.7); where $R_{\tilde{A}_\alpha}$ is one horizontal slice at level α.

$$\tilde{A} = \bigcup_{\alpha \in [0,1]} R_{\tilde{A}_\alpha} \tag{2.7}$$

2.3. Generalized type-2 fuzzy systems based on α-planes

In Figure 2.3, the block diagram of the generalized type-2 fuzzy inference system is presented. The generalized type-2 Mamdani fuzzy systems contain four main components: fuzzifier, rules, inference engine, type-reducer and defuzzifier. In the following, a general description about these components is presented.

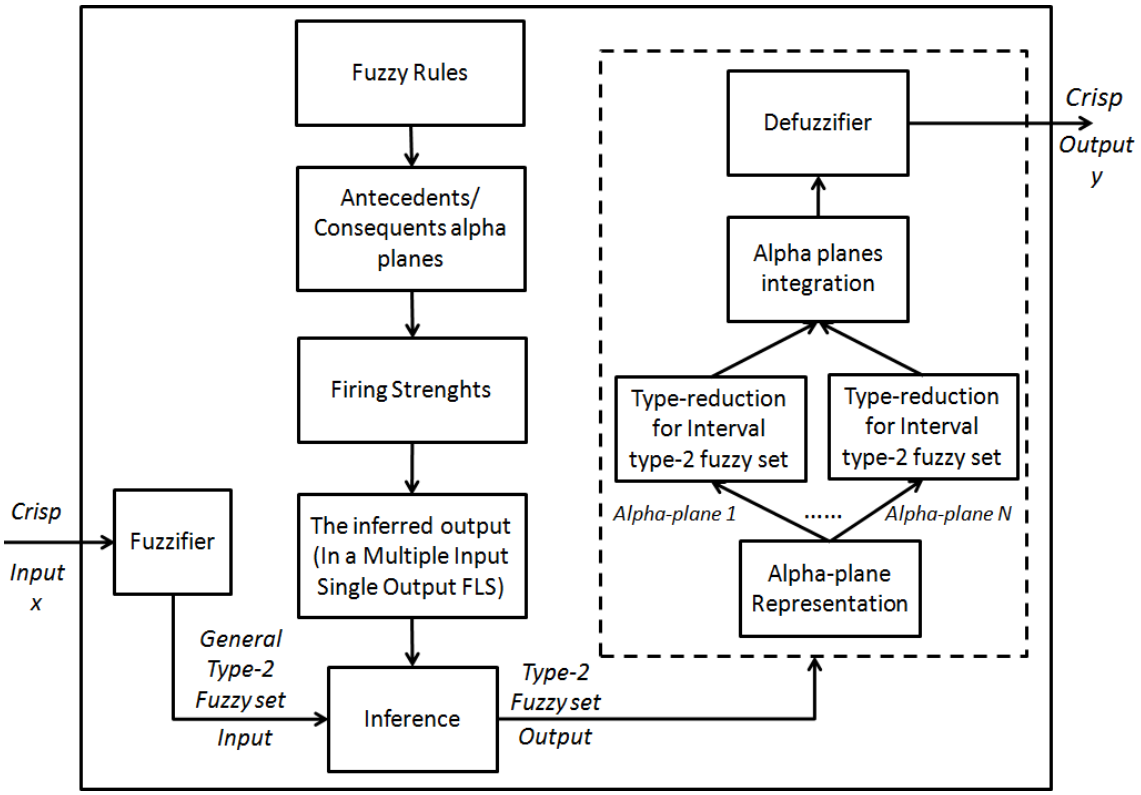


Figure 2.3 Generalized type-2 fuzzy logic system based on the α-planes representation

2.3.1. Fuzzifier process

The fuzzifier maps crisp inputs into generalized type-2 fuzzy sets to process within the FSs. In this thesis, we will focus on the type-2 singleton fuzzifier as it is fast to compute and, thus, suitable for the generalized type-2 FSs real-time operation. Singleton fuzzification maps the crisp input into a fuzzy set, which has a single point of nonzero membership. Hence, the singleton fuzzifier maps the crisp input x_p' into a type-2 fuzzy singleton, whose MF is $\mu_{\tilde{A}_p}(x_p) = 1/1$ for $x_p = x_p'$ and $\mu_{\tilde{A}_p}(x_p) = 0$ for all $x_p \neq x_p'$ for all $p = 1, 2, \ldots, P$, where P is the number of FLS inputs [34,36].

2.3.2. Fuzzy rules

The structure of the rules in the generalized type-2 FLS is the standard Mamdani-type FLS rule structure used in the type-1 FLS and in interval type-2 FLS, but in this thesis, we assume that the antecedents and the consequents sets are represented by generalized type-2 fuzzy sets. So for a type-2 FLS with p inputs $x_1 \in X_1, \ldots, x_p \in X_P$ and one output $y \in Y$, Multiple Input Single Output (MISO), if we assume there are M rules, the kth rule in the generalized type-2 FLS can be written as follows [32].

$$R^k: IF\ x_1\ is\ \tilde{F}_1^k\ and\ \ldots\ and\ x_p\ is\ \tilde{F}_p^k, THEN\ y\ is\ \tilde{G}^k \tag{2.8}$$

2.3.3. Inference engine

For performing the inference in the GT2 FLS, the α-planes representation was used. The equation of the α-plane is represented in eq. (2.6).

The inference engine for a generalized type-2 FLS based on α-planes, can be viewed as a combination of a series of interval type-2 fuzzy sets. During this inference process, a series of operations are computed as follows.

1. Obtain the α-planes for the antecedents and consequents. In this case, the α-planes are obtained in the secondary membership functions of the antecedents $\tilde{F}_i^k$ and consequents $\tilde{G}_i^k$ of the *ith* input, *kth* rule. The α-planes of the $\tilde{F}_i^k$, create an interval type-2 fuzzy set [38, 40, 44], which is defined by eq. (2.9).

$$\left(\tilde{F}_i^k\right)_\propto = \left\{ \int_{x_i' \in X_i} \left[\int_{\mu_{\tilde{F}_i^k}^\propto(x_i') \in \left[\underline{\mu}_{\tilde{F}_i^k}^\propto(x_i'), \overline{\mu}_{\tilde{F}_i^k}^\propto(x_i')\right]} 1 \Big/ \mu_{\tilde{F}_i^k}^\propto(x_i') \right] / x \right\} \tag{2.9}$$

where $\left(\tilde{\mathrm{F}}_{\mathrm{i}}^{\mathrm{k}}\right)_{\propto}$ can be written as

$$\left(\tilde{F}_i^k\right)_{\propto} = \left\{ \int_{x_i' \in X_i} \left[\underline{\mu}_{\tilde{F}_i^k}^{\propto}(x_i'), \quad \overline{\mu}_{\tilde{F}_i^k}^{\propto}(x_i')\right] / x_i' \right\} \tag{2.10}$$

The α-planes of the consequents $\tilde{G}_i^k$, are defined by (2.11).

$$\left(\tilde{G}_i^k\right)_{\propto} = \left\{ \int_{y_i \in Y_i} \left[\int_{\mu_{\tilde{G}_i^k}^{\propto}(y_i) \in \left[\underline{\mu}_{\tilde{G}_i^k}^{\propto}(y_j), \overline{\mu}_{\tilde{G}_i^k}^{\propto}(y_j)\right]} 1 \Big/ \mu_{\tilde{G}_j^k}^{\propto}(y_i) \right] / y_j \right\} \tag{2.11}$$

Another expression for $\left(\tilde{G}_i^k\right)_{\propto}$ is

$$\left(\tilde{G}_i^k\right)_{\propto} = \left\{ \int_{x_i' \in X_i} \left[\underline{\mu}_{\tilde{G}_i^k}^{\propto}(y_j), \quad \overline{\mu}_{\tilde{G}_i^k}^{\propto}(y_j)\right] / y_j \right\} \tag{2.12}$$

2. Calculate the firing strengths. The firing strengths of the rules are calculated, where the firing sets $\mu_{\tilde{\mathrm{F}}_{\mathrm{i}}^{\mathrm{k}}}^{\alpha}(\mathrm{x}_{\mathrm{i}}')$ for each α-plane (α), of the ith input and kth rule of a singleton type-2 FLS are represented by eq. (2.13).

$$\begin{aligned} \underline{\Omega}_{\propto}^k(x') &= \sqcap_{i=1}^n \left\{\underline{\mu}_{\tilde{F}_i^k}^{\alpha}(x_i')\right\} \\ \overline{\Omega}_{\propto}^k(x') &= \sqcap_{i=1}^n \left\{\overline{\mu}_{\tilde{F}_i^k}^{\alpha}(x_i')\right\} \end{aligned} \tag{2.13}$$

3. In a Multiple Input Single Output (MISO) FLS, the inferred output $\underline{\mu}_{\tilde{B}_j}^{\alpha}(y_j)$ and $\overline{\mu}_{\tilde{B}_j}^{\alpha}(y_j)$ of each rule k are represented by (2.14), where $\mu_{\tilde{G}_j^k}^{\alpha}$ is the type-2 fuzzy MF that represents the αth α-plane, kth rule, jth input of the consequents.

$$\underline{\mu}_{\tilde{B}_j}^{\alpha}(y_j) = \underline{\Omega}_{\propto}^{k}(x') \sqcap \underline{\mu}_{\tilde{G}_j^k}^{\propto}(y_j)$$
$$\overline{\mu}_{\tilde{B}_j}^{\alpha}(y_j) = \overline{\Omega}_{\propto}^{k}(x') \sqcap \overline{\mu}_{\tilde{G}_j^k}^{\propto}(y_j) \tag{2.14}$$

4. The outputs of the fired rules (M) are combined using the join operation to produce the overall output set, which can be expressed as follows.

$$\underline{\mu}_{\tilde{B}_j}^{\alpha}(y_j) = \sqcup_{k=1}^{r} \left\{\underline{\mu}_{\tilde{B}_j^k}^{\propto}(y_j)\right\}$$
$$\overline{\mu}_{\tilde{B}_j}^{\alpha}(y_j) = \sqcup_{k=1}^{r} \left\{\overline{\mu}_{\tilde{B}_j^k}^{\propto}(y_j)\right\} \tag{2.15}$$

2.3.4. Type reducer

To perform the defuzzification process, the centroid method is used. The centroid of a generalized type-2 fuzzy set Ã, can be obtained by taking the union of the centroids of all the α-planes of Ã, and then the Karnik-Mendel algorithm can be used for computing the centroid of each α-plane. The centroid of a generalized type-2 fuzzy system, introduced by Karnik and Mendel [40, 46, 47], uses the following definition of the centroid, which is expressed in eq. (2.16).

$$Y_C(\propto) = Centroid\left(\tilde{A}(\propto)\right) = \int_{u_1 \in {}^{\propto}J_{x_1}} \dots \int_{u_N \in {}^{\propto}J_{x_N}} \propto / \frac{\sum_{i=1}^{N} x_i \mu_A(x_i)}{\sum_{i=1}^{N} \mu_A(x_i)} = \propto / [{}^{\propto}y_l, {}^{\propto}y_r] \tag{2.16}$$

where $[^{\propto}y_l, ^{\propto}y_r]$ is the domain of the centroid. Obviously, computing $^{\propto}y_l$ and $^{\propto}y_r$ is the same as computing y_l and y_r for an interval type-2 fuzzy set; therefore, centroid type-reduction is performed by the Karnik-Mendel algorithms to compute $^{\propto}y_l$ and $^{\propto}y_r$ [47-48].

2.3.5. Defuzzification process

In the defuzzification phase a type-1 fuzzy output set is produced during the type-reduction process. In this thesis the GT2 fuzzy inference system is approximated using α-planes; so, for each α-planes the centroid type-reducer is applied with eq. (2.16); after that the results of the α-planes are integrated by eq. (2.17) and eq. (2.18) [32]. Finally the defuzzifier output is obtained by using the average of y^l and y^r, this is expressed in eq. (2.19) [39-40].

$$\hat{y}_j^l(x') = \frac{\sum_{i=1}^{N} \propto_i {}^{\propto_i}y_j^l(x')}{\sum_{i=1}^{N} \propto_i} \tag{2.17}$$

$$\hat{y}_j^r(x') = \frac{\sum_{i=1}^{N} \propto_i {}^{\propto_i}y_j^r(x')}{\sum_{i=1}^{N} \propto_i} \tag{2.18}$$

$$\hat{y}_j(x') = \frac{\hat{y}_j^l(x') + \hat{y}_j^r(x')}{2} \tag{2.19}$$

3. Edge detection methods and filters used on digital image processing

3.1. Edge detection methods

The aim of an edge detector, is to identify points on a digital image in which the brightness of the image changes dramatically or has discontinuities. There are many ways to perform edge detection, and one of them are the gradient methods. The gradient methods detects edges by looking for the maximum and minimum in the first derivative of an image. Some methods based on the gradient are: the Sobel, Prewitt, Roberts, Kirsch and Canny [3, 21, 22, 23]. In these algorithms the way to approximate the image gradient is to convolute an image with a kernel or filter. Besides this, there exists another edge detector method known as the morphological gradient, which uses the first derivative to calculate the image gradients, but without using any kernel.

In the following we define the morphological gradient approach [20] and the Sobel operator [21]; since, these are fundamental concepts for the proposed edge detection methods.

3.1.1. Morphological gradient approach

The morphological gradient of a gray scale image can be defined as the difference between the intensity values of two neighboring pixels that belong to a given structural element. The core of gradient edge detection is, of course, the gradient operator, ∇. In continuous form and applied to a continuous space image the gradient, $f_c(x, y)$, is defined by eq. (3.1) [20].

$$\nabla f_c(x,y) = \frac{\partial f_c(x,y)}{\partial x}\, i_x + \frac{\partial f_c(x,y)}{\partial y}\, i_y \tag{3.1}$$

where i_x and i_y are the unit vectors in the x and y directions, respectively. Notice that the gradient is a vector, having both magnitude and direction. Its magnitude, $\left|\nabla f_c(x_{0,}\, y_0)\right|$, measures the maximum rate of change in the intensity at the location $(x_{0,}\, y_0)$. Its direction is that of the greatest increase in intensity. To produce an edge detector, we consider the effect of finding the local maxima of $\nabla f_c(x,y)$ or the local maxima, and this is calculated by using eq. (3.2).

$$\left|\nabla f_c(x_{,}\, y)\right| = \sqrt{\left(\frac{\partial f_c(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f_c(x,y)}{\partial y}\right)^2} \tag{3.2}$$

The precise meaning of "local" is very important here, if the maxima of eq. (3.2) are found over a 2D neighborhood, the result is a set of isolated points rather than the desired edge contours [49].

In this thesis, we are going to use D_i instead of $\nabla f_c(x,y)$, we apply eq. (3.2) for a matrix of 3x3 as it is shown in Figure 3.1, so we can obtain the coefficients z_i with eq. (3.3), and the possible direction of the edge D_i with eq. (3.4). The edges S can be calculated with eq. (3.5) [18, 50].

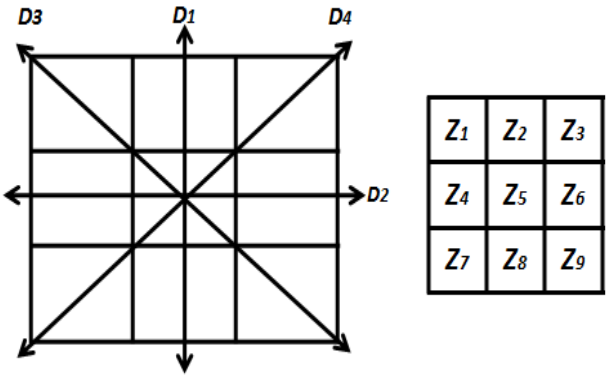


Figure 3.1 Matrix of 3x3 indicating the coefficients $\boldsymbol{Zi}$ and the edge direction $\boldsymbol{Di}$

$$\begin{aligned} \boldsymbol{z_1} &= f(x-1, y-1) \quad \boldsymbol{z_2} = f(x, y-1) \\ \boldsymbol{z_3} &= f(x+1, y-1) \quad \boldsymbol{z_4} = f(x-1, y) \\ \boldsymbol{z_5} &= f(x, y) \qquad\qquad \boldsymbol{z_6} = f(x+1, y) \\ \boldsymbol{z_7} &= f(x-1, y+1) \quad \boldsymbol{z_8} = f(x, y+1) \\ \boldsymbol{z_9} &= f(x+1, y+1) \end{aligned} \tag{3.3}$$

$$\begin{aligned} D1 &= \sqrt{(z_5 - z_2)^2 + (z_5 - z_8)^2} \\ D2 &= \sqrt{(z_5 - z_4)^2 + (z_5 - z_6)^2} \\ D3 &= \sqrt{(z_5 - z_1)^2 + (z_5 - z_9)^2} \\ D4 &= \sqrt{(z_5 - z_3)^2 + (z_5 - z_7)^2} \end{aligned} \tag{3.4}$$

$$S = D1 + D2 + D3 + D4 \tag{3.5}$$

3.1.2. Sobel operator

The classic Sobel operator is a 3 x 3 neighborhood based gradient operator. The convolution masks for the Sobel operator on a gray scale digital image are defined by *Sobelx* in eq. (3.6) and *Sobely* in eq. (3.7) [16, 21].

$$Sobelx = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.6}$$

$$Sobely = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.7}$$

The positions on the two masks are separately applied on the input image to yield two gradient components *gx* in eq. (3.8) and *gy* in eq. (3.9) in the horizontal and vertical orientations respectively; the positions of the input image (f) are shown in Figure 3.2, where the x-axis represents the horizontal positions and y-axis vertical positions.

(x-1, y-1)	(x-1, y)	(x-1, y+1)
(x, y-1)	(x, y)	(x, y+1)
(x+1, y-1)	(x+1, y)	(x+1, y+1)

Figure 3.2 Positions of the input image (f)

$$gx = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobelx_{i,j} \cdot f_{x+i-2,y+j-2} \tag{3.8}$$

$$gy = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobely_{i,j} \cdot f_{x+i-2,y+j-2} \tag{3.9}$$

where f is the source image.

The gradient magnitude or output edge is obtained by using eq. (3.10).

$$G[f(x,y)] = \sqrt{gx^2 + gy^2} \tag{3.10}$$

where gx in eq. (3.8) is the result of the convolution of the input image (*f*) with the $Sobelx$ filter in eq. (3.6), and gy in (3.9) is the result of the convolution of the input image (*f*) with the $Sobely$ filter in eq. (3.7) [51-52].

3.1.3. Sobel operator applied on color images

There are various models to represent color images, such as the RGB color model, YUV model, CMY color model, CMYK color model and the HIS color model [53-54]. In the methodology of this thesis the Sobel operator is applied on color images; these images are under the RGB model representation.

In the RGB model, an image consists of three independent image planes or dimensions, one in each of the primary colors: red, green and blue.

To apply the Sobel Operator on a RGB color image we need to calculate the gradient magnitude of eq. (3.10) for each dimension, which can be defined by the following pseudocode:

Pseudocode to calculate the gradient magnitude (12) for each dimension of a color image (I).
Input. *The kernels of Sobel* $Sobelx$ (8), $Sobely$ (9) *and the RGB color image* $(I_{r,c,d})$ *where* (r) *is the size rows of the image,* (c) *the columns size and* (d) *the dimension number; d= 1, 2, 3.*
Output. *The horizontal gradients* (gx_d)*, vertical gradients* (gy_d)*, gradient magnitude* (G_d) *for each dimension of the image (I) and the output edge (E).*

1. *Calculate the* gx_d *and* gy_d *gradients for each* (d) *dimension to obtain* (G_d)
2. ***for*** *x=2 to r-1*
3. ***for*** *y=2 to c-1*
4. ***for*** *d=1 to 3*
5. $$gx_d = \sum_{i=1}^{i=3}\sum_{j=1}^{j=3} Sobelx_{i,j} \cdot I_{x+i-2,\ y+j-2,\ d}$$
6. $$gy_d = \sum_{i=1}^{i=3}\sum_{j=1}^{j=3} Sobely_{i,j} \cdot I_{x+i-2,\ y+j-2,\ d}$$
7. $$G_d = \sqrt{gx_d{}^2 + gy_d{}^2}$$
8. ***end***
9. ***end***
10. ***end***
11. *Calculate the output edge (E)*

12. $E = \sum_{d=1}^{d=3} G_d$

The output (E) in the previous pseudocode represents the edge detection obtained by the Sobel operator when this is applied on a color image; in Figure 3.3, an illustration of this output is presented.

(a) Input image (b) Edge detection output

Figure 3.3 Color image edge detector using the Sobel operator

3.2. Filters

3.2.1. Low-pass filters

A low-pass filters attenuate high frequencies and the low frequencies are maintained unchanged [55-56]. The result in the spatial domain is equivalent to a smoothing filter. The high frequencies are filtered where there exist strong intensity changes. The low-pass $(lowPF)$ filter is implemented by using eq. (3.11), where $lowM$ in eq. (3.12) we have the mask used to filter the image (I).

$$lowPF = lowM * I \tag{3.11}$$

$$lowM = \frac{1}{25} * \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

3.2.2. High-pass filters

High-pass filters are used to clean up low-frequency noise, as they attenuate the low frequencies while the high frequencies are unchanged. Support the contrasts found in the image. The details, edges and high frequency changes are intensified; otherwise, the uniform shade areas are attenuated [55-56]. The high-pass $(highPF)$ filter is implemented by using eq. (3.13).

$$highPF = highM * I \tag{3.13}$$

The mask $(highM)$ used to obtained the $highPF$ is expressed in eq. (3.14).

$$highM = \begin{bmatrix} -1/16 & -1/8 & -1/16 \\ -1/8 & 3/4 & -1/8 \\ -1/16 & -1/8 & -1/16 \end{bmatrix} \tag{3.14}$$

4. Metrics for edge detection methods

There are different types of methods to measure the performance of an edge detection method, which usually consider different parameters for assessing the abrupt change of color in the pixels. This section describes some available metrics to measure the quality of the edge detection method.

4.1. Figure of merit of Pratt (FOM)

One of the most frequently used techniques is the figure of merit of Pratt (Figure of Merit; FOM). This measure represents the deviation of an actual (calculated) edge point from the ideal edge and it is defined in eq. (4.1).

$$FOM = \frac{1}{\max(K,J)} \sum_{i=1}^{J} \frac{1}{1+ \propto d_i^2} \tag{4.1}$$

In (4.1) J is the actual number of detected edge points, K is the number of edge points on the ideal edge, *d(i)* is the distance between the edge of the current pixel and its correct position in the reference image and α is scaling constant (usually 1/9). To implement this metric a test image is needed, such as the one in Figure 4.1, and the reference image that represents *K*, in this case the reference image (with the ideal edges) of Figure 4.1 is the one in Figure 4.2. Then we apply any edge detection method to obtain the value of J (Figure 4.3) that represents the number of detected edge points. Now if the result of eq. (4.1) is 1 or very close to 1, this means that the detected edge J is the same or very similar to the ideal edge K. Otherwise, the closer the value is to 0, this means that there is a high difference between the detected edge and the ideal edge [57-59]**.**

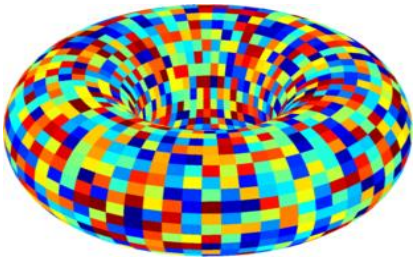


Figure 4.1 Doughnut synthetic image

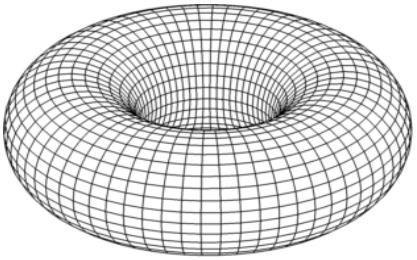


Figure 4.2 Reference image (K) or ideal edge of Figure 4.1

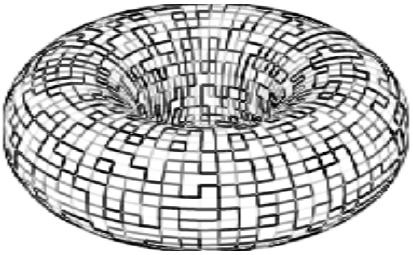


Figure 4.3 Output of the edge detection method (J) for Figure 4.1

4.2. Quality measurement using the MSE, PSNR and SSIM indices

The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image compression quality [60]. In this thesis we use the PSNR and MSE to compare the quality of the proposed edge detector with a reference image or an ideal edge. An example for an edge detector and an ideal edge is shown in Figures 4.3 and 4.2 respectively.

The MSE represents the cumulative squared errors between the proposed edge detector (Figure 4.3) and the ideal edge (Figure 4.2). The PSNR is used to compute the peak signal-to-

noise ratio, in decibels, between two images. The higher the value of the PSNR, then the better the quality of the edge detector found. The PSNR is calculated using eq. (4.2) and (4.3).

$$PSNR = 10log_{10}\left(\frac{\left(MAX_J\right)^2}{MSE}\right) \tag{4.2}$$

In eq. (4.2) MAX_J is the maximum value in the input image (J).

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[J(i,j) - K(i,j)]^2 \tag{4.3}$$

where m and n are the rows and columns number respectively in the images (J, K).

Another metric used to quality measurement is the SSIM index. The structural similarity (SSIM) index is a method for measuring the similarity between two images [61]. The SSIM index can be used to measure the quality achieved by any edge detector method; where the edge (Figure 4.3), can be compared with other image (Figure 4.2) which is regarded as of perfect quality or ideal edge. The resultant SSIM index is a decimal value between -1 and 1, and the value 1 is only reachable in the case of two identical sets of data.

The Structural Similarity (SSIM) Index is a quality assessment index based on the computation of three terms, namely the luminance term, the contrast term and the structural term. The index is expressed in eq. (4.4) [62].

$$SSIM(J,K) = \frac{\left(2\mu_J\mu_K + C_1\right)\left(2\sigma_{JK} + C_2\right)}{\left(\mu_J^2 + \mu_K^2 + C_1\right)\left(\sigma_J^2 + \sigma_K^2 + C_2\right)} \tag{4.4}$$

where μ_J, μ_K, σ_J, σ_K and σ_{JK} are the local means, standard deviations, and cross-covariance for images J, K.

5. Edge detection methods based on generalized type-2 fuzzy logic systems

5.1. Edge detection method based on GT2 FSs and the Morphological gradient

In this section the proposed model for edge detection based on a generalized type-2 fuzzy inference system combined with the morphological gradient technique is presented, which is described in the following steps.

1) Read the input image. The first step in the whole process is reading an input image for applying the edge detection method. In this case, we are only considering images with a gray scale.

2) Obtain the image gradients. In this step the morphological gradient technique, described in Section 3.1.1 is applied, to obtain the gradients in the four directions using the eq. (3.3) and eq. (3.4), and then use them as the inputs for the proposed generalized type-2 fuzzy inference system. For this case study the inputs are represented by the gradients D_i of the original image, and each of them will be an input to the fuzzy system.

3) Define the input linguistic variables. For the GT2 FS four inputs are defined, in which each one has three Gaussian membership functions with uncertain mean. The linguistic values used for the four inputs are: low, medium, high. In order to adapt the membership functions to the range of gray tones depending on the image, we obtain the maximum, minimum and middle values of D_i with the eqs. (5.1)-(5.3) and we use these values for calculating the mean of the membership functions, but adding different sizes of the FOU.

4) Calculate the FOU for the inputs. For this task, we made tests using different sizes of the FOU for the D_i input variables. The Gaussian membership functions for each D input are

obtained with the eqs. (5.12)–(5.16), and the means of each function are obtained with the eqs. (5.5)–(5.6). For example, for the high membership functions, the first mean was obtained with eq. (5.5), the second mean was calculated with eq. (5.6) and the σ value was obtained with eq. (5.4).

5) Define the output. The inference system has one output *S* (the edge), the linguistic values used for the output are: *edge* and *no_edge*, and we selected the range [0, 1], since the input image was normalized in this range, where the minimum value for the output is represented by eq. (5.7) and maximum by eq. (5.8). The Gaussian membership functions for the output are obtained with eqs. (5.12)–(5.16), the means of each function are obtained with eq. (5.10) and (5.11) and the σ value with eq. (5.9).

6) Obtain the FOU for the output. The FOU for the output variable *S*, was calculated in a similar way to the inputs variables. This is the method that we propose to adapt the parameters of the membership functions depending on the contrast level of each image.

7) In Figure 5.1 and Figure 5.2, we show the linguistic values for the variables with generalized type-2 membership functions, where the FOU value is 0.2 on the inputs (D_1 and D_2) and 0.25 for the output (S). We have to note that we are using a number between 0 and 1 to represent the size of the FOU, which of course it is only a crisp value that represents the average size of the footprint to model the particular problem.

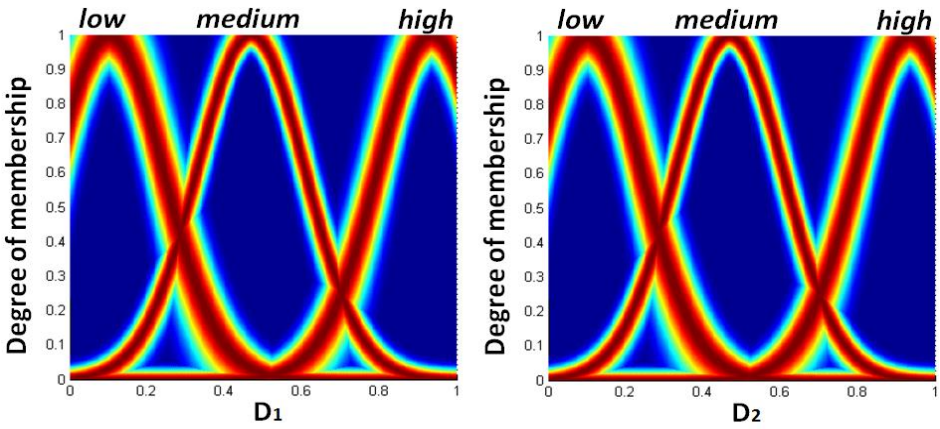


Figure 5.1 GT2 input membership functions for the variables D_1 and D_2

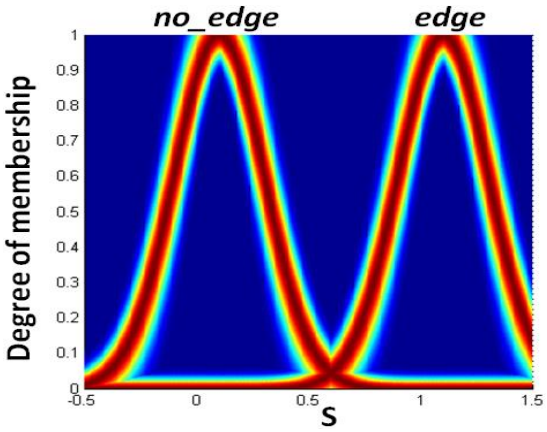


Figure 5.2 GT2 output membership function for the variable S

$$low_i = min(D_i) \quad (5.1)$$

$$high_i = max(D_i) \quad (5.2)$$

$$medium_i = low_i + (high_i - low_i)/2 \quad (5.3)$$

$$\sigma_i = high_i/5 \quad (5.4)$$

$$m_1 = high_i \quad (5.5)$$

$$m_2 = m_1 + (m_1 * FOU), \text{ where } FOU \text{ is in } (0, 1) \quad (5.6)$$

$$no\_edge_i = 0 \quad (5.7)$$

$$edge_i = 1 \quad (5.8)$$

$$\sigma_i = edge/4 \quad (5.9)$$

$$m_1 = edge_i \quad (5.10)$$

$$m_2 = m_1 + (m_1 * FOU), \text{ where } FOU \text{ is in } (0, 1) \quad (5.11)$$

$$\tilde{\mu}(x,u) = \text{gausmgausstype2}(x,u,[\sigma_x, m_1, m_2]) \tag{5.12}$$

where "gausmgausstype2" stands for Gaussian generalized type-2 membership function with uncertain mean

$$\tilde{\mu}(x) = \left[\underline{\mu}(x), \bar{\mu}(x)\right] = \text{igausmtype2}(x,[\sigma_x, m_1, m_2]) \tag{5.13}$$

where "igausmtype2" stands for Gaussian interval type-2 membership function with uncertain mean

$$m_x = \frac{m_1 + m_2}{2}$$

$$m_x = \frac{m_1 + m_2}{2} \tag{5.14}$$

$$\sigma_u = \frac{\delta}{2\sqrt{6}} + \varepsilon$$

$$p_x = \text{gaussmf}(x,[\sigma_x, m_x]) = \exp\left[-\frac{1}{2}\left(\frac{x - m_x}{\sigma_x}\right)^2\right] \tag{5.15}$$

$$\tilde{\mu}(x,u) = \text{gaussmf}(u,[\sigma_u, p_x]) = \exp\left[-\frac{1}{2}\left(\frac{x - p_x}{\sigma_u}\right)^2\right] \tag{5.16}$$

8) Fuzzy inference rules. For modeling the process with the fuzzy system, we consider three rules (Table 5.1) that help to describe the existing relationship between the image gradients. The fuzzy rules are the following.

Table 5.1 Fuzzy rules for the MG + GT2 FSs edge detection

Inputs				**Output**	**Operator**
D1	**D2**	**D3**	**D4**	**S**	
high	*high*	*high*	*high*	*edge*	*or*
medium	*medium*	*medium*	*medium*	*edge*	*or*
low	*low*	*low*	*low*	*no_edge*	*and*

5.2. Edge detection method based on GT2 FSs and the Sobel operator

In this section the edge detector based on generalized type-2 fuzzy logic and the Sobel operator is described. The fuzzy approach for edge detection based on the Sobel operator consists in using the eq. (3.8) and eq. (3.9) to obtain the gradients in the two directions, and then

use them as inputs for an appropriately define the fuzzy inference system. The methodology to obtain this edge detection method is defined as follows.

1) Define the inputs. For the generalized type-2 fuzzy inference system, two inputs are required, which are the gradients with respect to the *x-axis* and *y-axis*, calculated with the eq. (3.8) and (3.9), for this case study we call them DH and DV respectively. For all the inputs and output fuzzy variables, the fuzzy sets are defined by Gaussian membership functions with uncertain mean. In the input membership functions we used three linguistic values: *low*, *middle* and *high*; the parameters of the membership functions depend on the gradients of the image to use for the experiment.

2) Define the output. The output (*EDGES*) is represented by the linguistic values: *background* and *edge*. The *EDGES* output variable is used to find and normalize the edges to any range of required values; which also has been adjusted to have a range between -4.5 and 5, since it is the better range of values to normalize the edges matrix. In Figure 5.3, the general structure of the proposed edge detection system is shown.

3) Fuzzy rules. For modeling the process with the fuzzy system, we consider three fuzzy rules that allow to evaluate the input variables; such that, the output image displays the edges of the image in color near white (high tone or EDGE), whereas the background was in tones near black (low tone or background), these rules were obtained based on expert knowledge and after performing several experiments. The fuzzy rules are presented in Table 5.2.

Table 5.2 Knowledge base of fuzzy rules for the Sobel + GT2 FSs edge detection

Inputs		**Output**	**Operator**
DH	**DV**	**EDGES**	
high	*high*	*edge*	*or*
medium	*medium*	*edge*	*or*
low	*low*	*background*	*and*

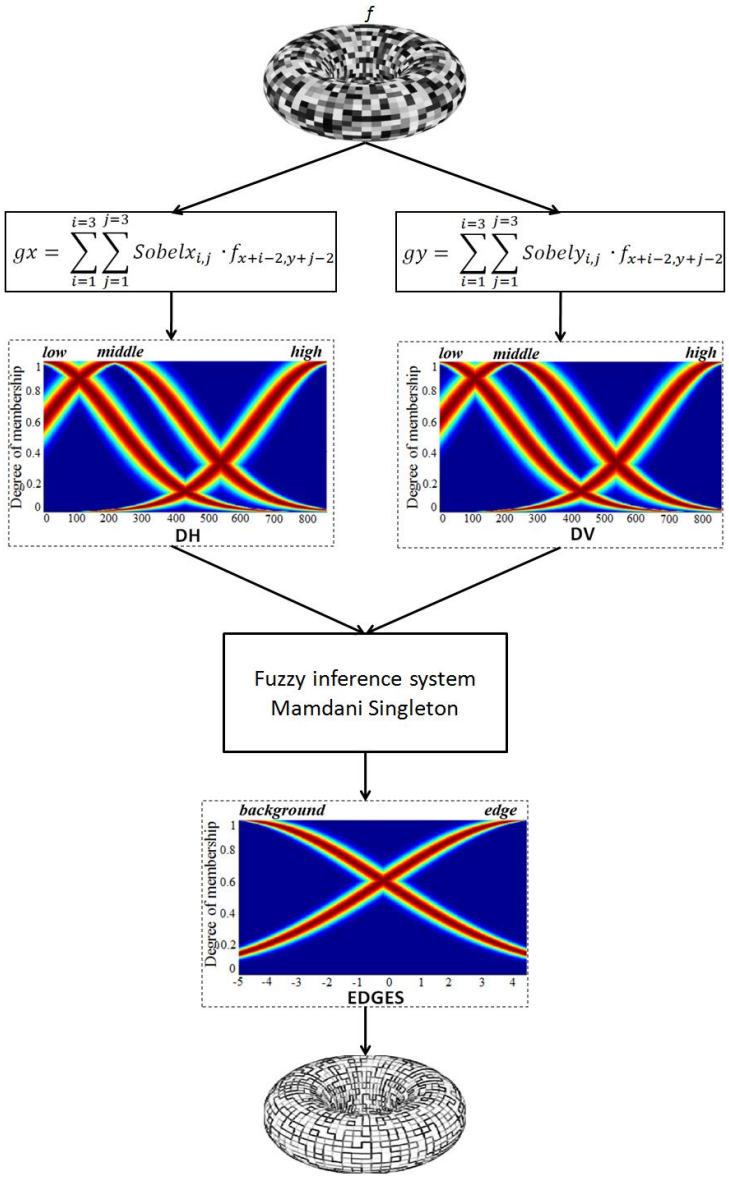


Figure 5.3 Generalized type-2 fuzzy inference system for edge detection

5.3. Generalized type-2 fuzzy edge detection method applied on color images

In this section, an application on digital image processing using fuzzy systems applied on color images is presented. The idea is to develop a color image edge detector based on the Sobel operator combined with the generalized type-2 fuzzy systems. The proposed fuzzy color edge detection method consists of the following steps.

1) The input color image (I) is separated into $(d, for\ d = 1,\ldots,3)$ dimensions which represent the red, green and blue channels.

2) The Sobel operator is applied in each single channel to generate the horizontal gradients (gx_d) and vertical gradients (gy_d); these gradients are calculated by the pseudocode defined in Section 3.1.3, and specifically with the steps 2 to 10.

3) The six gradients previously obtained are indentified as $(gx_1,\ gy_1,\ gx_2,\ gy_2,\ gx_3,\ gy_3)$ and these are used as the inputs in the generalized type-2 FSs.

4) Define the inputs for the GT2 FSs. The general structure of the fuzzy system consists in six inputs; the inputs are fuzzified by Gaussian membership functions which are granulated in three linguistic values labeled as *low*, *middle* and *high*. A sample of the input GT2 MFs are shown in Figure 5.4.

5) Define the outputs. The GT2 FSs consists in three outputs, which represent the gradient magnitude for each color image dimension (G_d). The outputs also are fuzzified using Gaussian membership functions, but these are granulated in two linguistic values (*background* and *edge*). The combination of the three outputs generates the color edge detector. The output GT2 MFs is shown in Figure 5.5.

6) Obtain the fuzzy rules. After that the inputs and outputs membership functions were defined, and the next step was to design the fuzzy rules; so, these were designed considering different combinations of the gradients $(gx_1,\ gy_1,\ gx_2,\ gy_2,\ gx_3,\ gy_3)$ and to produce the gradient magnitude for each image dimension (G_1, G_2, G_3). In our proposal, the fuzzy rules were separately combined according with the color image dimension. In order to improve the fuzzy edge detection, diverse combinations of rules were made. After several tests, the best performance was obtained by the set of rules presented in Table 5.3; these fuzzy rules represent

the knowledge about the edge detection process. The general idea is to achieve good results with a minimum number of rules.

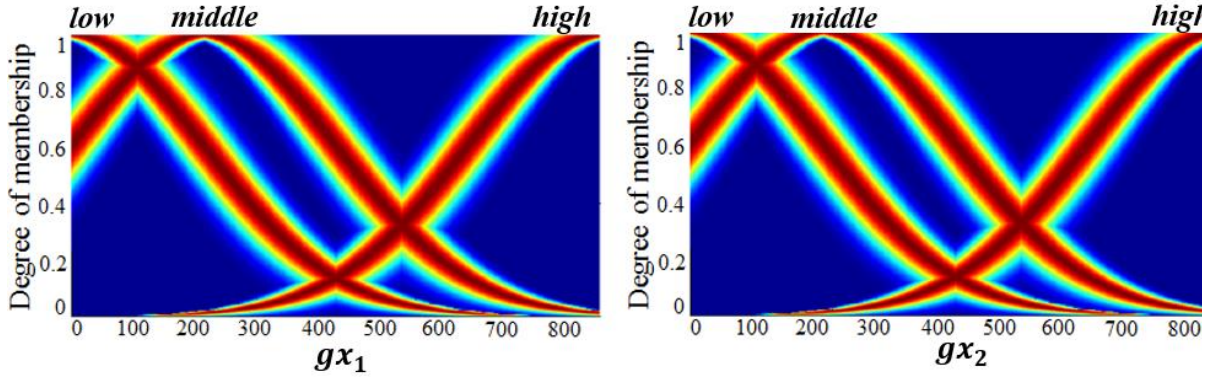


Figure 5.4 GT2 input membership functions for the variables gx_1 and gx_2

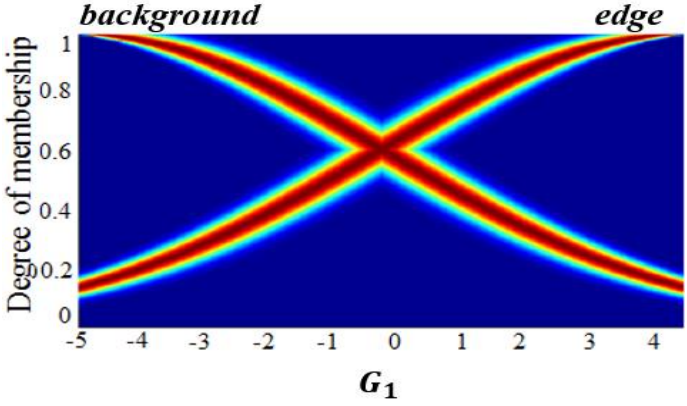


Figure 5.5 GT2 output membership functions for the variable G_1

Table 5.3 Fuzzy rules of the Sobel + GT2 FSs edge detection for color images

Inputs						**Outputs**			**Operator**
gx_1	**gy_1**	**gx_2**	**gy_2**	**gx_3**	**gy_3**	**G_1**	**G_2**	**G_3**	
low	*Low*	--	--	--	--	*background*	--	--	*and*
middle	*middle*	--	--	--	--	*edge*	--	--	*or*
high	*high*	--	--	--	--	*edge*	--	--	*or*
--	--	*Low*	*low*	--	--	--	*background*	--	*and*
--	--	*middle*	*middle*	--	--	--	*edge*	--	*or*
--	--	*High*	*high*	--	--	--	*edge*	--	*or*
--	--	--	--	*low*	*low*	--	--	*background*	*and*
--	--	--	--	*middle*	*middle*	--	--	*edge*	*or*
--	--	--	--	*high*	*high*	--	--	*edge*	*or*

In more detail the generalized type-2 fuzzy inference system is a singleton Mamdani-type, this was approximated by using α-planes [38, 39] and the type-reduction process was performed with the Centroid method [34]. Each α-plane Centroid was computed by the Karnik–Mendel algorithm [40].

5.4. Edge detection method using GT2 fuzzy images

5.4.1. Fuzzy synthetic images

An image can be represented as a two-dimensional function, $f(x, y)$, where $f(x, y)$ is known as the intensity or gray value at a point (x, y). In order to process images with a digital computer, they need to be digitized first. The process of digitization involves sampling and quantization. The image needs to be sampled at $m\ x\ n$ discrete array points. Usually, images are quantized in 256 discrete levels. With 256 discretization levels, a 0 represents black, a 255 represents white, and in-between values represents different gray tones [1].

The representation for a digitized image function is expressed in Figure 5.6. Note that a digital image is composed of a finite number of elements; these elements are referred to as pixels. In Figure 5.7 the numerical pixel value for a grayscale image is shown [59].

$$f(x,y) = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(1,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,1) & \cdots & f(M,N) \end{bmatrix}$$

$$f(x,y) = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{1,n} \\ \vdots & \vdots & & \vdots \\ f_{m,1} & f_{m,1} & \cdots & f_{m,n} \end{bmatrix}$$

Figure 5.6 Digital image representation

(a)

224	181	129	111	101
118	73	67	66	61
86	70	66	63	59
171	74	67	63	57
251	194	74	62	46

(b)

Figure 5.7 (a) Synthetic digital image, (b) Numerical pixel value representation of Figure 5.7(a)

In our approach the idea is to apply the edge detector on fuzzy images. The image fuzzification process is performed using T1, IT2 and GT2 membership functions which is described below.

5.4.1.1. Fuzzy images using type-1 membership functions

For the T1 fuzzy images, all pixels are classified with different membership values. These fuzzy values or fuzzy pixels are obtained using a triangular membership function (Figure 5.8) which is expressed in eq. (5.17). According to the membership degrees, a sample of the fuzzy pixels assigned by eq. (5.17) are shown in Figure 5.9.

$$\mu_A(x) = \frac{x}{max(A)} \tag{5.17}$$

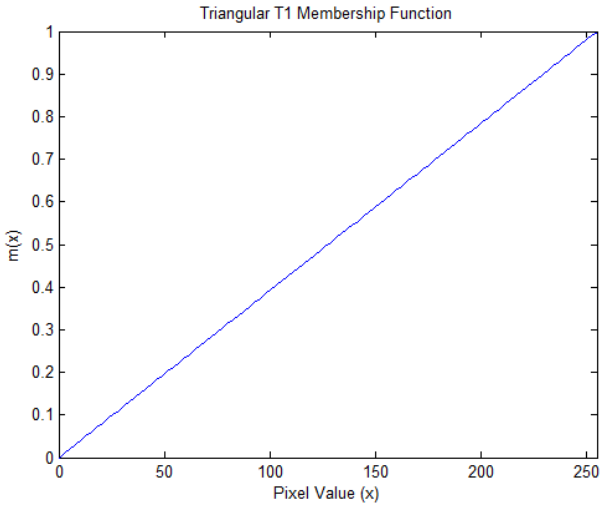


Figure 5.8 Triangular Type-1 Membership Function

0.88	0.71	0.51	0.44	0.40
0.46	0.29	0.26	0.26	0.24
0.34	0.27	0.26	0.25	0.23
0.67	0.29	0.26	0.25	0.22
0.98	0.76	0.29	0.24	0.18

Figure 5.9 Type-1 fuzzy pixel value representation of Figure 5.7(a)

5.4.1.2. Fuzzy images using interval type-2 membership functions

For the IT2 fuzzy images, the fuzzy pixels are obtained by using an IT2 triangular MF (Figure 5.10) which is expressed in eq. (5.18). An example of the IT2 fuzzy pixels are shown in Figure 5.11 and we can note that each pixel is represented by an interval value.

$$\mu_{\tilde{A}}(x) = \left[\frac{x}{max(\overline{A})}, \quad \frac{x}{max(\underline{A})}\right] \tag{5.18}$$

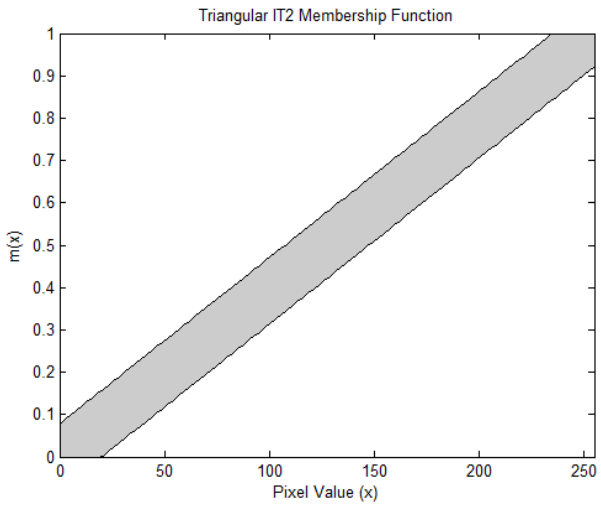


Figure 5.10 Triangular interval type-2 membership function

[0.18, 0.26]	[0.54, 0.63]	[0.93, 1]	[0.79, 0.87]	[0.71, 0.79]
[0.88, 0.92]	[0.49, 0.57]	[0.45, 0.52]	[0.45, 0.52]	[0.4, 0.47]
[0.60, 0.67]	[0.47, 0.55]	[0.45, 0.52]	[0.42, 0.49]	[0.38, 0.46]
[0.63, 0.71]	[0.50, 0.58]	[0.45, 0.52]	[0.42, 0.49]	[0.35, 0.43]
[0, 0.04]	[0.43, 0.52]	[0.50, 0.58]	[0.4, 0.47]	[0.28, 0.36]

Figure 5.11 IT2 fuzzy pixel value representation of Figure 5.7(a)

5.4.1.3. Fuzzy images using generalized type-2 membership functions

To obtain the GT2 fuzzy images, a GT2 MF with triangular on the primary MF and Gaussian in the secondary MF (Figure 5.12) is used. The fuzzifier process is denoted in eq. (5.19). The fuzzy pixels assigned by eq. (5.19) are illustrated in Figure 5.13. The interesting part in this definition is that each pixel is represented by a Gaussian MF; so, it permits better modeling of the uncertainty that could exist in a pixel, when this is corrupted by noise.

$$\mu_{\tilde{\tilde{A}}}(u) = \frac{\mu(x,u)}{Sup_u\big(\mu(x,u)\big)} \tag{5.19}$$

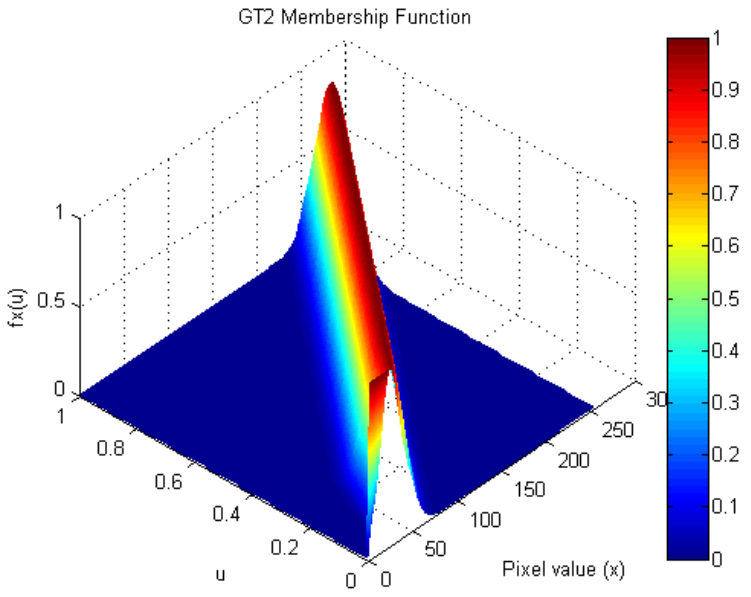


Figure 5.12 Generalized type-2 membership function

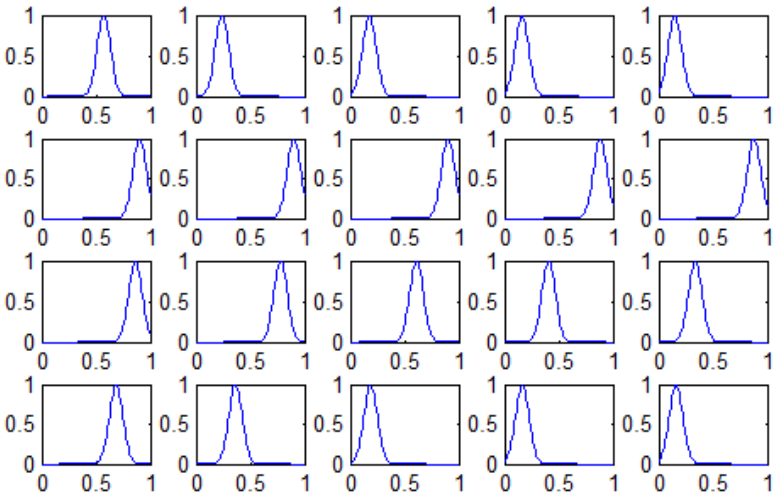


Figure 5.13 Generalized T2 fuzzy pixel value representation of Figure 5.7(a)

5.4.2. The fuzzy Euclidean distance

The Euclidean distance between two crisp points, is commonly used to find the similarity between two images. If one of the images is assigned as reference of correctly detected edges, the Euclidean distance can be a quality measure of the detected edges of another image and is commonly used in the edge detection methods which are based in the gradient technique to find the similarity between two points.

As the proposal presented in this section consists in develop a new edge detection method based in the gradient technique but applied on fuzzy images, we need to use the Euclidean distance for type-1 and interval type-2 fuzzy sets, which are expressed as follows.

Let the fuzzy sets A, B in $X = \{x_1, x_2, \cdots, x_n\}$ the Euclidean distance $E(A, B)$ for type-1 fuzzy sets is defined by eq. (5.20) and for the interval type-2 is expressed in eq. (5.21) [63-65].

$$E(A,B) = \sqrt{\sum_{i=1}^{n} \left(\mu_A(x_i) - \mu_B(x_i)\right)^2} \tag{5.20}$$

$$E(A,B) = \sqrt{\left(\underline{\mu}_A - \underline{\mu}_B\right)^2 + \left(\overline{\mu}_A - \overline{\mu}_B\right)^2} \tag{5.21}$$

5.4.3. Edge detection method applied on fuzzy images

In this Section a new edge detection method applied on fuzzy images is presented. The proposed approach is applied on digital images; these images are fuzzifier using first type-1 fuzzy sets, secondly interval type-2 fuzzy sets and in thirdly generalized type-2 fuzzy sets. The general idea is that each numerical pixel in an image can be manipulate such as fuzzy number. Due is because when an image is captured by an image acquisition hardware; there exist several factors that can add uncertainty to the image more specifically in each pixel by variation of

brightness or color information. Because we are not sure if the pixel value is precise we can add uncertainty and manipulate this pixel such a fuzzy pixel.

In this edge detection method the gradients are obtained using the morphological gradient approach presented in Section 3.1.1; but, due the fact that in our proposal the idea is work with fuzzy images, in the methodology we implement the concepts of fuzzy images and fuzzy Euclidean distance presented at the beginning of this Section. For more detail, to implement this novel approach we present a pseudocode below; in which the pixel is fuzzifier using a generalized type-2 membership function (trigausstype2), but for a type-1 or an interval type-2 the membership function can be change by a "trimf" or "itritype2" according to the fuzzy type set used.

Pseudocode to calculate the gradient magnitude (12) for a GT2 fuzzy image (I)

Input. *The gray scale digital image* (I)

Output. *The four gradients* ED_i *where i=1...4, and the output fuzzy edge (FE)*

1. *Obtain the r (rows) and c (columns) of the input image* (I)
2. *[r, c]= size (I)*
3. *Calculate the* ED_i *gradients for i=1...4*
4. ***for*** *x=2 to r-1*
5. ***for*** *y=2 to c-1*
6. *Fuzzifier the pixels using GT2 membership functions to obtain the coefficients* z_k*, (k=1...9)*

$$z_k = trigausstype2\left(\left(I_{x+i-2,\ y+j-2}\right), u, [a1, b1, c1, a2, b2, c2, rho]\right),$$

$$\forall\, i = 1 \cdots 3, j = 1 \cdots 3, k = i \cdot j$$

7. *where "trigausstype2" stands for a generalized type-2 membership function (GT2 MF) with triangular in the primary and Gaussian in the secondary membership function. The variables* $a1, b1, c1, a2, b2, c2$ *and* rho*, are the parameters to calculate this GT2 MF.*

8. *Obtain the gradients using the fuzzy Euclidean distance eq. (5.20)*

9. $ED_i = \sqrt{(\sum(z_5 - z_i)^2) + (\sum(z_5 - z_{9-i+1})^2)}, \forall\, i = 1 \cdots 4$

10. ***end***

11. ***end***

12. *Calculate the output edge (FE)*

13. $FE = \sum_{d=1}^{d=3} ED_d$

6. Generalized type-2 fuzzy edge detection applied on a face recognition system

This section consists in two parts; first we are presenting the methodology to develop a GT2 fuzzy edge detection method, and in this proposed method we are including the low-pass and high-pass filters. Secondly, the fuzzy edge detector is applied on a face recognition system.

6.1. Generalized type-2 fuzzy edge detection method using the Sobel operator and filters

The general structure used to obtain the fuzzy edge detector is illustrated in Figure 6.1. The steps to calculate an edge is defined as follows; first we select the input image (I), secondly, we calculate the horizontal gx in eq. (3.8) and vertical gy in eq. (3.9) gradients of the image using the Sobel operator which as explained in Section 3.1.2, in the third step the low-pass and high-pass filters are calculated by using eq. (3.11) and (3.13). Finally, the gx, gy, $lowPF$ and $highPF$ values are used as the inputs for the GT2 FS to obtain the output edge.

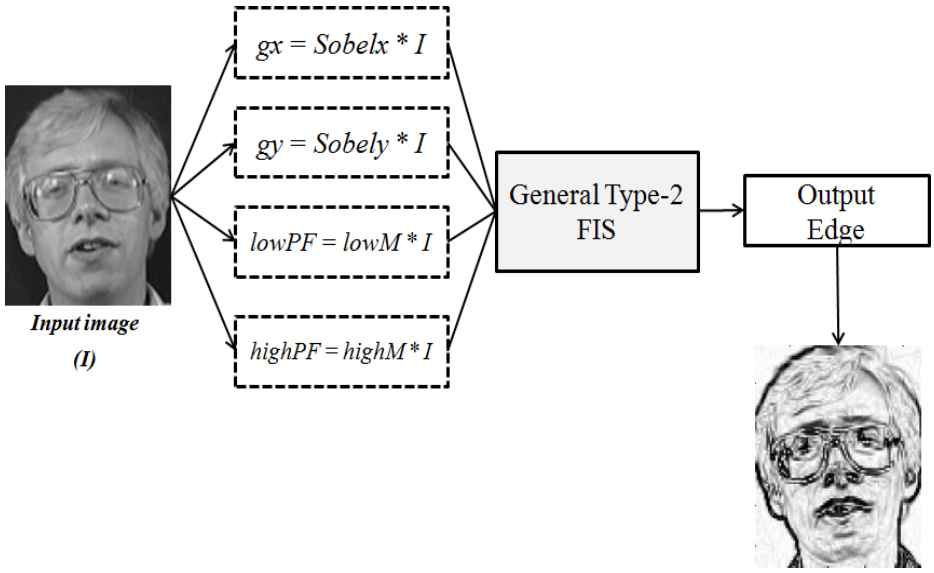


Figure 6.1 Edge detection method improved with a GT2 FSs, Sobel operator and low and high pass filters

The GT2 FS was designed with four inputs and one output. The inputs are the values obtained by gx in eq. (3.8), gy in eq. (3.9), $lowPF$ in eq. (3.11) and $highPF$ in eq. (3.13) respectively; each input was granulated in three membership functions with the linguistic values (*low*, *middle and high*); moreover, the GT2 FS has one output which is labeled as *OuputEdge*; the output represents the fuzzy gradient magnitude (*edge*).

The input and output membership functions used in the GT2 FSs are shown in Figure 6.2 and 6.3 respectively.

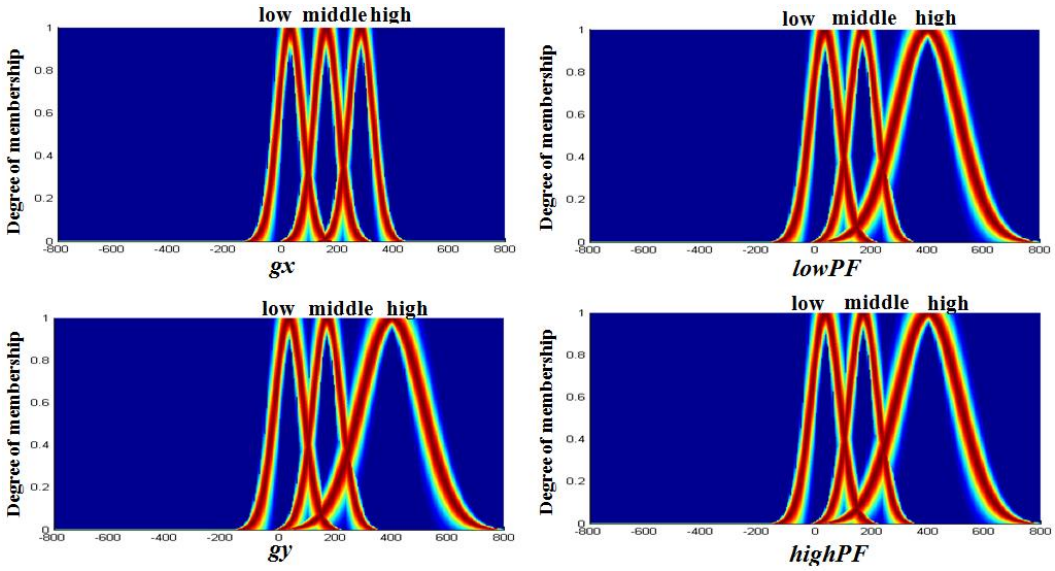


Figure 6.2 Input membership functions using in the GT2 FS edge detection

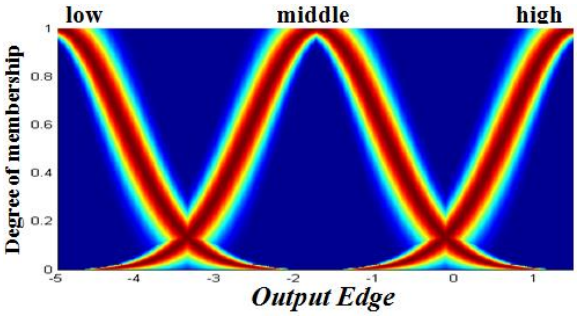


Figure 6.3 Output membership function using in the GT2 FS edge detection

The knowledge base of the fuzzy rules used in the fuzzy inference system were obtained based on expert knowledge and after several experiments and the values are presented in Table 6.1.

Table 6.1 Fuzzy rules for GT2 fuzzy edge detection method applied on the face recognition system

Inputs				**Output**	**Operator**
gx	**gy**	**lowPF**	**highPF**	**OutputEdge**	
low	*low*	--	--	*high*	*and*
middle	*middle*	--	--	*low*	*and*
high	*high*	--	--	*low*	*and*
middle	--	*low*		*high*	*and*
middle			*low*	*low*	*and*
--	*middle*	*low*		*high*	*and*
--	*middle*	--	*low*	*low*	*and*

The previous fuzzy rules are used to infer the gray tone of each pixel for the image with the following reasoning.

In the first rule when the horizontal gradient gx and vertical gradient gy are *low* means that there is no difference between the gray tones in its neighbors pixels, then the output pixel must belong to a homogeneous or not edges region, and then the output pixel is *high*.

In the second and third rule when gx and gy are *high* means that a difference there exists between the gray tones in its neighborhood, then the output pixel is *low* or in this case represents an *EDGE*.

6.2. Face recognition system based on a monolithic neural network

As already mentioned above, one of the main goals of this thesis is to apply the fuzzy edge detector presented in this Section in a pattern recognition system using Monolithic Neural Networks.

The general structure used for the pattern recognition system is shown in Figure 6.4 and is explained in more detail the following steps.

A. Define the input image databases.

In the experiment three different face databases were used for testing, the ORL [66], the Cropped Yale [67-68] and FERET [69].

B. Detect the edges of each image.

The edge detection process was performed using the GT2 fuzzy edge detector described above in this Section.

C. Training the monolithic neural network.

After applying the edge detection, the images are used as inputs of the neural network. In order to evaluate more objectively the recognition rate a k-fold cross validation method was used. The training process is defined as follows.

1. Define the parameter for the monolithic neural network [55].
    - Hidden layers: two
    - Neuron number in each layer: 200
    - Learning algorithm: Descent gradient with momentum and adaptive learning.
    - Goal error: 1e-4.
2. The indices for training and testing the k-folds were calculated as follows.
    - Define the people number (p).
    - Define the sample number for each person (s).

- Define the k-folds ($k = 5$).
- Calculate the number of samples (m) in each fold by using eq. (6.1)

$$m = (s/k) \cdot p \qquad (6.1)$$

- The train data set size (i) is calculated with eq. (6.2)

$$i = m(k - 1) \qquad (6.2)$$

- Finally, the test data set size is calculated with eq. (6.3), are the samples number in only one fold.

$$t = m \qquad (6.3)$$

- The train and test set obtained for the three face databases used in this work are shown in Table 6.2.

3. The neural network was trained k-1 times, one for each training fold previously calculated.
4. The neural network was tested k times, one for each fold test set previously calculated.

The mean of the rates of all the k-folds are calculated to obtain the recognition rate.

Table 6.2 Information for the tested face databases

Database	**People number (p)**	**Samples number (s)**	**Fold size (m)**	**Training set size (i)**	**Test set size (t)**
ORL	40	10	80	320	80
Cropped Yale	38	10	76	304	76
FERET	20	10	40	160	40

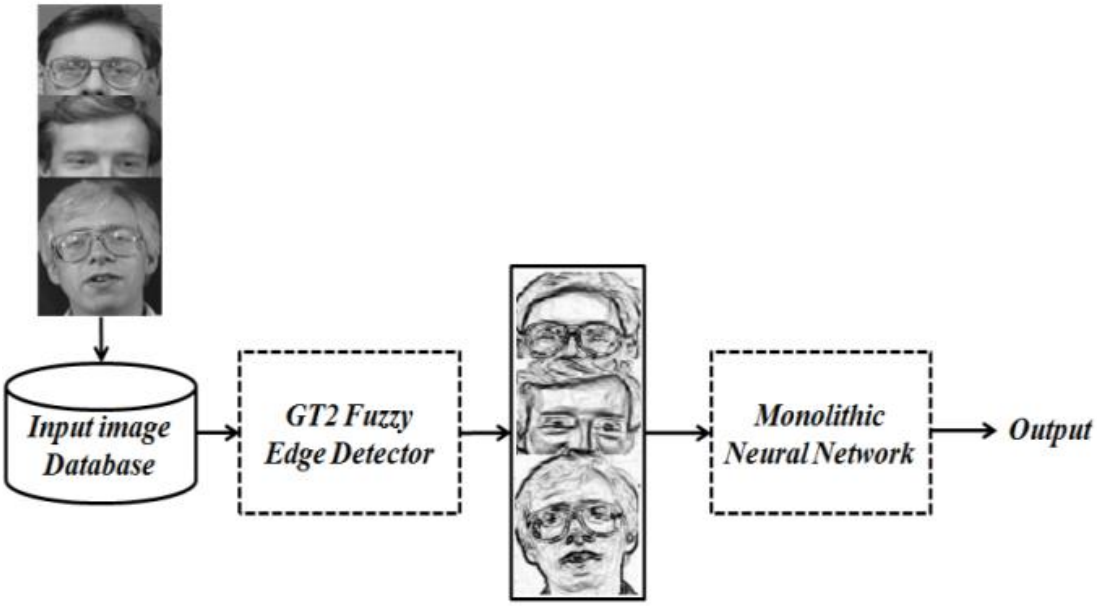


Figure 6.4 General structure for the face recognition system

7. Experimentation and results discussion

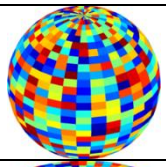
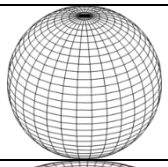
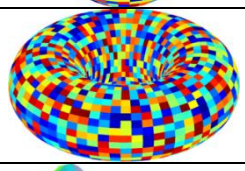
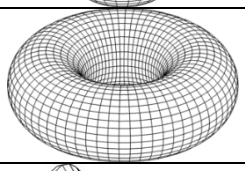
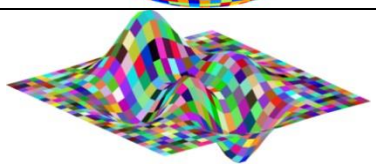
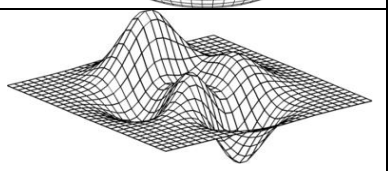
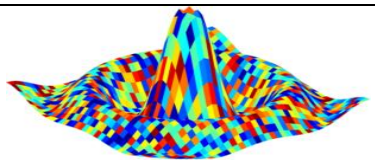
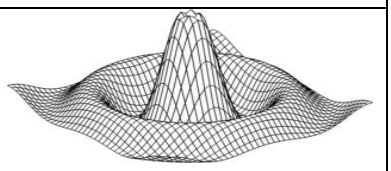
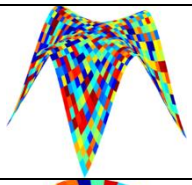
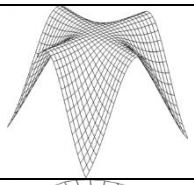
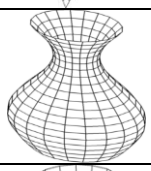
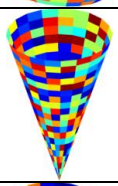
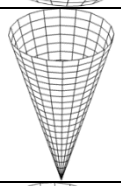
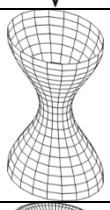
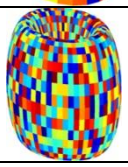
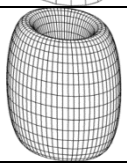
For an objective evaluation of the proposed fuzzy edge detection methods it is necessary to choose a ground truth image with known ideal edges because the method takes them as reference to compare with the detected edges. Some authors use synthetic images [17, 52] for the results validation, and this is a good strategy because with the computer we can generate the image edges with some mathematical functions and we can use it as the reference for the ideal edges (Figure 4.2); then using the same function, the computer can generate a filled image and we can use it as a simulation of a realistic image (Figure 4.1).

In order to evaluate the performance achieved by the proposed edge detection methods, for the simulation we perform the experiments using synthetic images which were built by plotting eleven mathematical functions as original images and as ground truth reference for the ideal edges. In Table 7.1 the synthetic and reference images are presented. In the experimental results we also use a benchmark database of real images, which are shown in Table 7.2; these were obtained from the USC-SIPI database [66].

This section provides a comparison analysis of the edge detection accuracy and noise robustness of the proposed edge detection methods based on GT2 FSs, which were presented in Section 5; besides this, the values obtained by the GT2 fuzzy edge detection methods are compared against the results achieved by the T1 and IT2 fuzzy edge detection methods.

Additionally, in this section we also present a comparative analysis with the recognition rates achieved by the GT2 against the results obtained by T1 and T2 fuzzy edge detectors; when these are applied on a face recognition system.

Table 7.1 Synthetic images for the simulation results

Image Number	Synthetic images	Reference images
1		
2		
3		
4		
5		
6		
7		
8		
9		

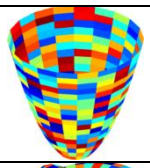
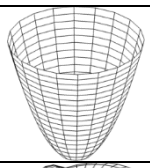
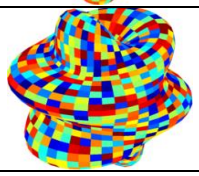
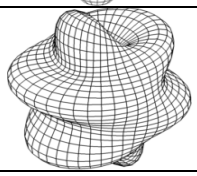
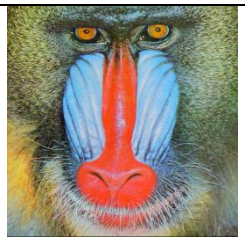
10		
11		

Table 7.2 Real color images database

Real images	
	
	
	

7.1. Generalized type-2 fuzzy systems combined with the Morphological gradient

In this section the experimental results about the edge detection method based on GT2 FSs and MG technique are presented. The methodology about this fuzzy edge detection method was explained in Section 5.1 and the images used for these tests can be find in Tables 7.1 and 7.2.

For all the images the morphological gradient was obtained with eq. (3.3) and eq. (3.4). In the implementation of the generalized type-2 system (GT2 FSs) the gradients obtained with eq. (3.4) are used as inputs and the fuzzy system was build using the membership functions presented in Figures 5.1 and 5.2. For measuring the quality of detected edges, the figure of merit of Pratt (FOM) was used (described in Section 4.1).

In this thesis to approximate the GT2 FSs the α-planes theory was used (described in Section 2.2); for this reason, the first experiment was performed varying the number of α-planes necessary to approximate the output. The number of α-planes used in this test were considered to be with values of 5, 10, 50, 100, 150, 200 and 1000. Is important to mention that this test was applied only for the first image of Table 7.1 (Sphere).

The results of this comparison are shown in Table 7.3. In Table 7.3 can be noted that in images without noise, the obtained FOM value was of 0.9554 for all number of α-planes, but in images with noise, a variation in the metric values is observed, for images with noise level of 0.001 the metric value was of 0.9382 with 50 α-planes, and for images with noise level of 0.002 the best value obtained was 0.9319 applying for this case 100 α-planes.

Table 7.3 Simulation results obtained by MG + GT2 FSs varying the number of α-planes

α-planes Number	**FOM**		
	Noise Level		
	0	0.001	0.002
5	0.9554	0.9267	0.9295
10	0.9554	0.9211	0.9306
50	0.9554	**0.9382**	0.9283
100	0.9554	0.9292	**0.9319**
150	0.9554	0.9291	0.9151
200	0.9554	0.9291	0.9093
1000	0.9554	0.9223	0.9004

Due to the fact that for different number of α-planes similar FOM values were obtained, the rest of the experiments were performed using only five α-planes, this in order to minimize the computer time.

In the second test the MG + GT2 FSs method was applied on the first ten synthetic images presented in Table 7.1. In order to make a comparison anlysis with another type of fuzzy systems, the experiments were also performed using the MG + IT2 FSs [16-18], MG + T1 FSs [18] and of course the traditional MG technique. The T1 and IT2 FSs were improved using a similar methodology presented in Section 5.1, same number of inputs and outputs, same fuzzy rules and type reducer method. For more detail on the comparison results about the four edge detection methods the results are presented in Table 7.4.

Table 7.4 FOM values for edge detection method using MG + GT2 FSs, MG + IT2 FSs, MG + T1 FSs and MG applied on synthetic images

Image number	**FOM VALUES**			
	MG + GT2 FSs	**MG + IT2 FSs**	**MG + T1 FSs**	**MG**
1	0.9536	0.9494	0.9493	0.8165
2	0.9446	0.9440	0.9437	0.8531
3	0.9448	0.9368	0.9336	0.8605
4	0.9495	0.9481	0.9394	0.9197
5	0.9402	0.9399	0.9399	0.8982
6	0.9525	0.9496	0.9523	0.8636
7	0.9435	0.9432	0.9432	0.8676
8	0.9431	0.9422	0.9408	0.8837
9	0.9557	0.9520	0.9515	0.7921
10	0.9475	0.9447	0.9438	0.8528
Mean	0.9475	0.9450	0.9438	0.8608
Stdev	0.0051	0.0048	0.0059	0.0369

According with the results presented in Table 7.4, we can notice that the best results were

achieved when the MG + GT2 FSs was applied. Beside this, the MG + IT2 FSs improved on the results obtained by the MG + T1 FSs. In general we can observe that the fuzzy edge detection methods were better that the traditional MG technique. A plot about these results is shown in Figure 7.1; otherwise, in Table 7.5 a sample of the output edges obtained by the four methodologies are illustrated; in which for better appreciation, the background was plotted in white color and black in the detected edge.

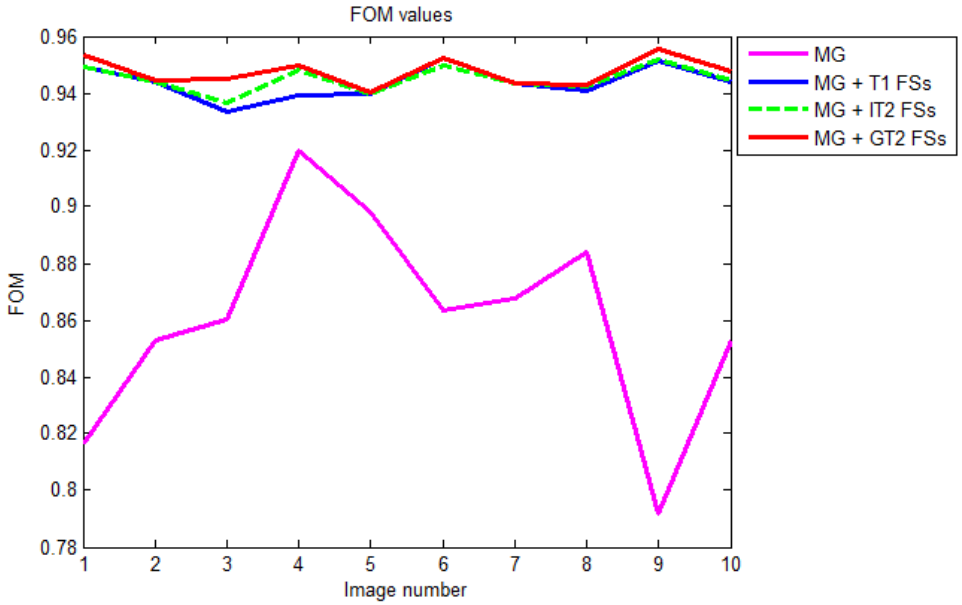
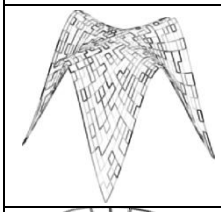
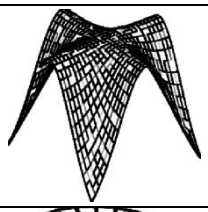
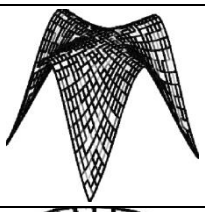
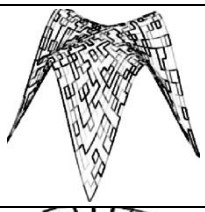
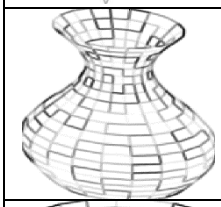
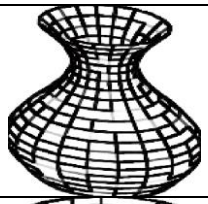
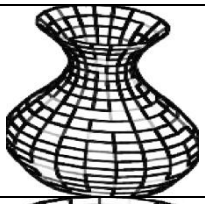
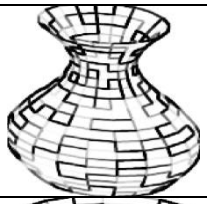
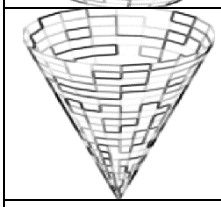
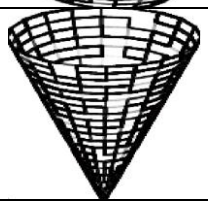
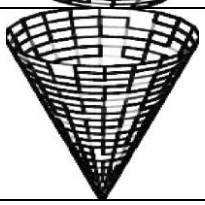
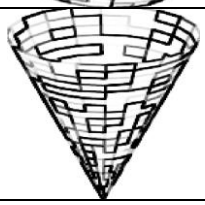
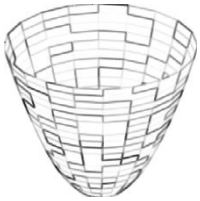
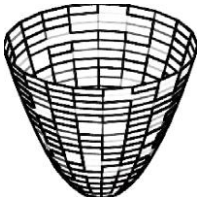


Figure 7.1 Simulations results of the edge detection methods using MG + GT2 FSs, MG + IT2 FSs, MG + T1 FSs and MG applied on synthetic images

The experiments were also performed applying Gaussian noise with levels between 20 and 50 dbi's. In the simulation results for these experiments we included the MG + T1 FSs, MG + IT2 FSs and MG methods. The results presented in Table 7.6 represent the mean of all the FOM values achieved by each synthetic image of Table 7.1, when the different noise levels were applied.

Table 7.5 Edge detection outputs using MG, MG + T1 FSs, MG + IT2 FSs and MG + GT2 FSs

MG	MG + T1 FSs	MG + IT2 FSs	MG + GT2 FSs
			
			
			
			

Based on the results of Table 7.6, we can analyze that the FOM values achieved by the MG + IT2 FSs edge detection were better than the MG + T1 FSs with mean values of (0.7425, 0.9374, 0.9434, 0.9460); as well as the values obtained by the MG + GT2 FSs improved on the results reached by the MG + IT2 FSs for all the dbi's levels, where the mean values are (0.7523, 0.9408, 0.9455, 0.9469). In the other hand, the lowest values were obtained by the traditional MG, this is because the traditional technique does not consider any parameters to prevent noise or regions with very low or high contrast.

Table 7.6 Edge detection method using MG + GT2 FSs, MG + IT2 FSs, MG + T1 FSs applied on synthetic images with Gaussian noise

Edge Detector	FOM VALUES							
	DBI's							
	20		30		40		50	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
MG	0.6243	0.1130	0.7663	0.0316	0.8278	0.0319	0.8545	0.0324
MG + T1 FSs	0.7382	0.1235	0.9374	0.0186	0.9435	0.0090	0.9442	0.0074
MG + IT2 FSs	0.7425	0.1245	0.9374	0.0184	0.9434	0.0086	0.9460	0.0047
MG + GT2 FSs	**0.7523**	0.1172	**0.9408**	0.0152	**0.9455**	0.0056	**0.9469**	0.0047

In another test, three fuzzy edges detection methods based on MG + T1 FSs, MG + IT2 FSs and MG + GT2 FSs are now applied to the real images database found in Table 7.2. We can observe in Table 7.2 that for these images, we do not have the ideal edges or reference images, which are important to calculate the FOM metric; for this reason, in these experiments first the ideal edges were obtained using the traditional Morphological gradient technique; after that we applied Gaussian noise on all images and finally the output edges are calculated with the fuzzy edge detection methods. The FOM values achieved in this test are presented in Table 7.7 were these results represent the mean of the six images of Table 7.2.

In Table 7.7, again we can analyze that the MG + IT2 FSs improved on the results obtained by the MG + T1 FSs with mean values of (0.8038, 0.9690, 0.9748, 0.9733). Otherwise, the MG + GT2 FSs mean values (0.8380, 0.9799, 0.9838, 0.9839) were better that the MG + IT2 FSs; the main reason is that the uncertainty in edge detection is modeled more closely with GT2 FSs; in other words, the GT2 FSs allows for better modeling of uncertainty, because it gives more degrees of freedom in comparison to T1 and IT2 FSs. The behavior about these results are shown graphically in Figure 7.2. In Table 7.8, examples of output edges are shown after performing the simulation with the already mentioned fuzzy edge detection methods; these samples were obtained on images with a noise level of 50 dbi's.

Table 7.7 Edge detection results using MG + GT2 FSs, MG + IT2 FSs, MG + T1 FSs applied on real images with Gaussian noise

Edge detection method	FOM VALUES							
	DBI's							
	20		30		40		50	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
MG + T1 FSs	0.7905	0.0825	0.9620	0.0331	0.9705	0.0202	0.9685	0.0249
MG + IT2 FSs	0.8038	0.0723	0.9690	0.0208	0.9748	0.0143	0.9733	0.0155
MG + GT2 FSs	**0.8380**	0.0672	**0.9799**	0.0075	**0.9838**	0.0051	**0.9839**	0.0048

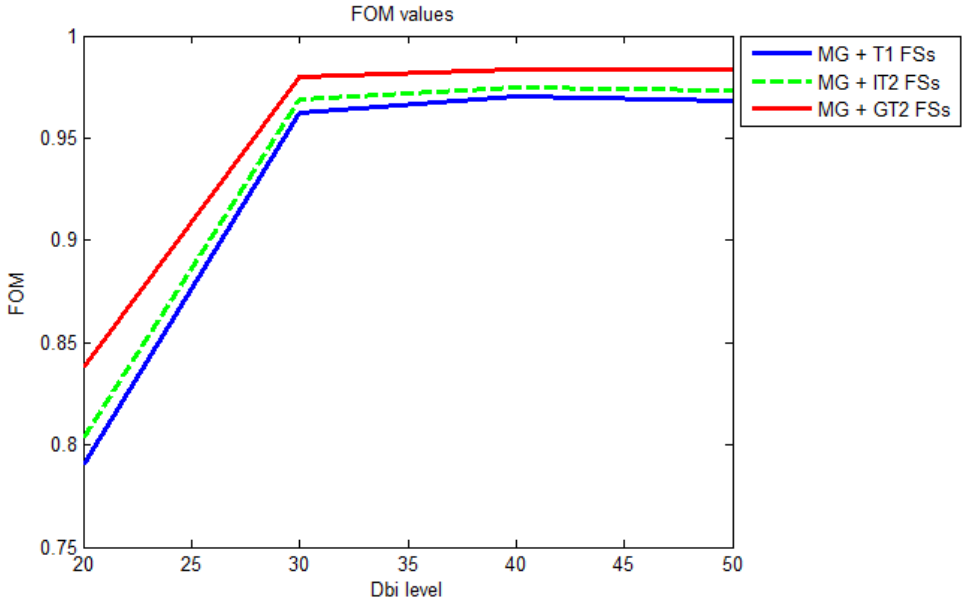
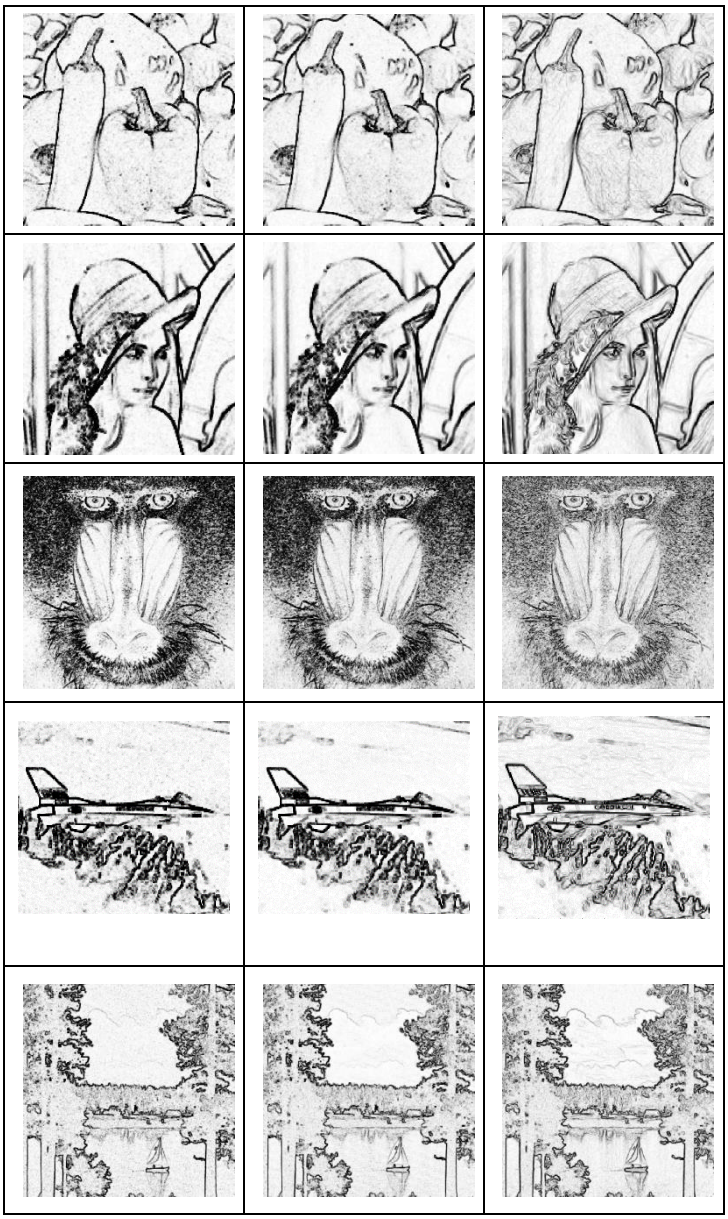


Figure 7.2 Simulation results about edge detection methods using MG + GT2 FSs, MG + IT2 FSs, MG + T1 FSs applied on real images with Gaussian noise

Table 7.8 Output edge detection using MG + T1 FSs, MG + IT2 FSs and MG + GT2 FSs for real images with noise level

MG + T1 FSs	MG + IT2 FSs	MG + GT2 FSs
		



7.2. Generalized type-2 fuzzy systems combined with the Sobel operator

In this section the simulation results about the edge detection method based on GT2 FSs and the Sobel technique are presented. The steps to develop this fuzzy edge detection method are described in Section 5.2. All the experiments implemented in this section were applied on the image databases presented in Tables 7.1 and 7.2.

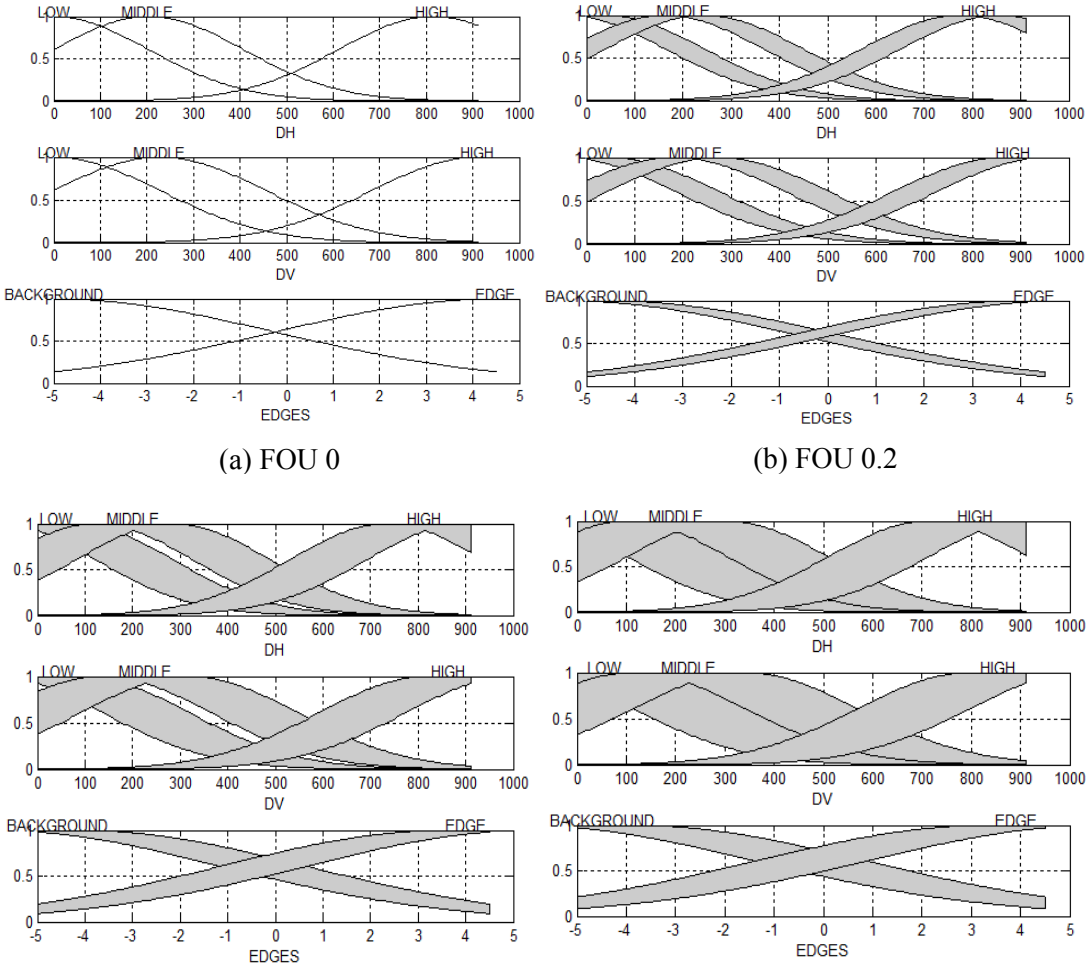
In the first test, the edge detection process was performed using the Sobel + IT2 FSs method and it was applied on the ten synthetic images (Table 7.1). This method (Sobel + GT2 FSs) is designed with a similar structure to the GT2 FSs (which was described in Section 5.2), based on the same inputs, outputs and fuzzy rules. For this test we selected arbitrary values for the Footprint of Uncertainty (FOU); the FOU is varied in the range between 0 and 1. For a better analysis of the detected edges, the metric of Pratt (FOM), described in Section 4.1 is applied. The mean of the metric values obtained after the simulation of the ten synthetic images is presented in Table 7.9. It can be observed that the measurements obtained with the FOM given by eq. (4.1) are better when a FOU value of 0.8 is used, where a FOM of 0.9606 is obtained. As mentioned previously, a FOM close to one means that the detected edge has good quality.

The trend of the FOM values in Table 7.9 is such that as the FOU factor is increased, the values of the FOM also increase. These changes can be explained as follows: different FOU's represent different levels of uncertainty and there should be an optimal level of uncertainty for modeling the image; however, after the execution of the ten experiments, with a FOU factor of one, the FOM value was lower, and this means that the problem reaches its point of generalization and does not need more uncertainty level or has reached its optimum level.

In the simulation results when using a FOU value of 0, the T1 FSs is simulated, where a FOM of 0.9321 is obtained. Based on these results it can be noted that the edge detector obtained using IT2 FSs is better that the edge detector obtained with T1 FSs. In the example of Figure 7.3, the changes on the IT2 membership functions with different values of the FOU are shown.

Table 7.9 Simulation results with variation in the FOU for the Sobel + IT2 FSs

FOU factor	Pratt's (FOM)
	Mean
0	0.9321
0.2	0.9352
0.4	0.9413
0.6	0.9504
0.8	0.9606
1.0	0.9598



(a) FOU 0

(b) FOU 0.2

(c) FOU 0.8

(d) FOU 1

Figure 7.3 IT2 FSs membership functions with different values of the FOU

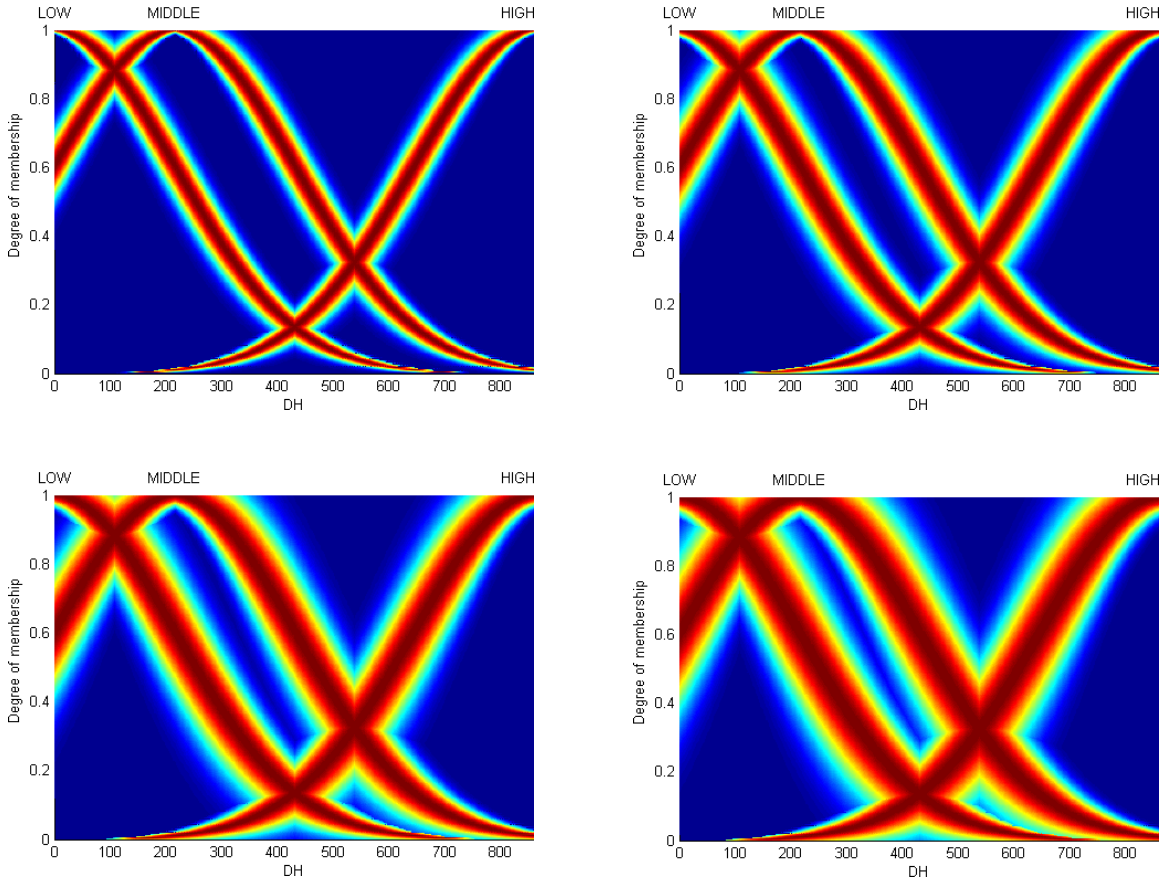


Figure 7.4 DH input for different generalized type-2 fuzzy membership functions

The results of this simulation are shown in Table 7.10, where the best result is obtained with a FOU factor of 0.4 with a FOM value of 0.9618. The FOM values are very similar; however we noticed that when the FOU is increased, the quality of the edge detector (FOM) is decreased, this means that the generalized type-2 memberships functions having more degrees of freedom in the footprint of uncertainty, for this case study, they reach the optimum level when they have a support with a FOU factor of 0.4.

Table 7.10 Simulation results with variation in the FOU value for the Sobel + GT2 FSs

FOU factor	Pratt's (FOM)
	Mean
0.2	0.9617
0.4	0.9618
0.6	0.9617
0.8	0.9616
1.0	0.9615

The tests were performed using images with white Gaussian noise; the noise level added was between 20 and 50 dbi's. The test results of the 10 images with Gaussian noise for the Sobel + IT2 FSs and Sobel + GT2 FSs are shown in Table 7.11, and for all experiments the Sobel + GT2 FSs was better than Sobel + IT2 FSs.

Table 7.11 Comparative analysis of Sobel + IT2 FSs and Sobel + GT2 FSs for images with white Gaussian noise

Edge detection method	DBI's					
	20		30		50	
	Mean	Stdev	Mean	Stdev	Mean	Stdev
Sobel + IT2 FSs	0.8515	0.0264	0.9114	0.0431	0.9371	0.0183
Sobel + GT2 FSs	**0.8730**	**0.0472**	**0.9340**	**0.0138**	**0.9411**	**0.0185**

In the simulation results the traditional Sobel edge detector was also applied, and in the same manner the metric of Pratt (FOM) is implemented to evaluate the quality of edge detection, the results of this simulation are presented in Table 7.12. In Table 7.12, four edge detectors are shown: the traditional edge detector Sobel, the Sobel + T1 FSs (obtained implicitly in simulating the edge detector IT2 FSs with FOU factor of 0), the Sobel + IT2 FSs and the proposed method Sobel + GT2 FSs; for the methods based on IT2 and GT2 FSs the means are obtained from Tables 7.9 and 7.10 respectively.

Tables 7.12 and 7.13 show that the proposed method based on GT2 FSs is better than IT2 FSs; obtaining a FOM value of 0.9617, while IT2 FSs was better than T1 FSs with a FOM value of 0.9495 and the traditional Sobel method is the worst with a FOM value of 0.7875. In Table

7.14, examples of detected edges are shown after performing the simulation with the already mentioned methodologies (Sobel, T1 FSs, IT2 FSs and GT2 FSs). Graphically we cannot notice the difference in the detected edges and for this reason the FOM metric in eq. (4.1) is applied.

Table 7.12 Simulation results using Sobel, Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs

Edge detection method	**Pratt's (FOM)**
	Mean
Sobel	0.7875
Sobel + T1 FSs	0.9321
Sobel + IT2 FSs	0.9495
Sobel + GT2 FSs	0.9617

Table 7.13 Results for the t student test for the edge detection method based on Sobel+IT2 and Sobel+GT2 FSs

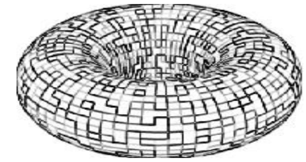
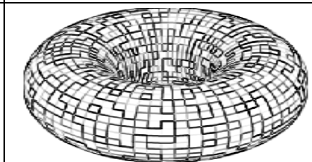
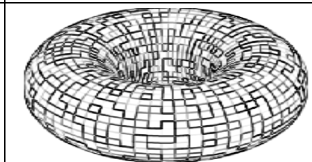
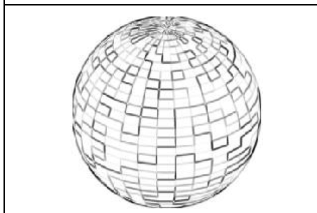
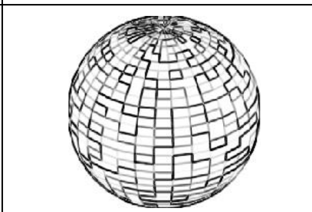
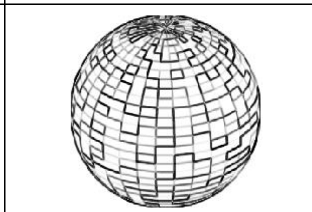
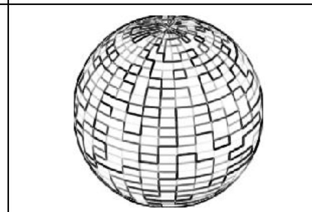
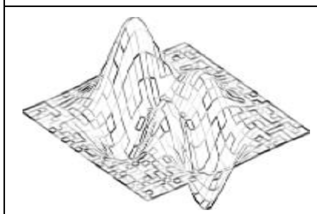
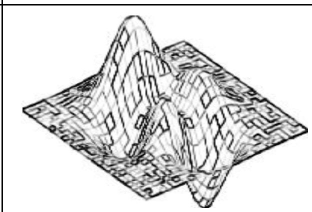
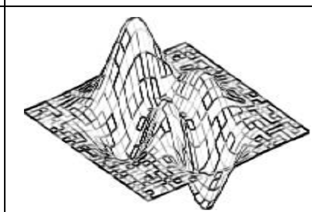
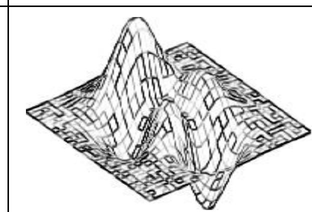
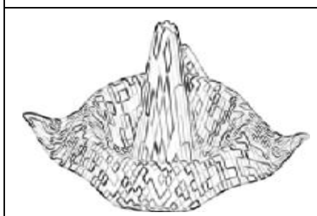
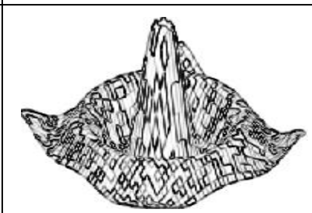
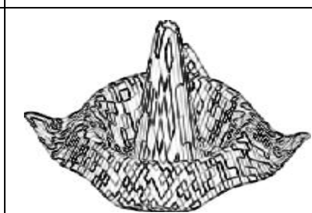
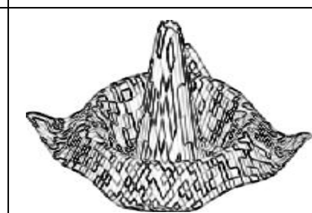
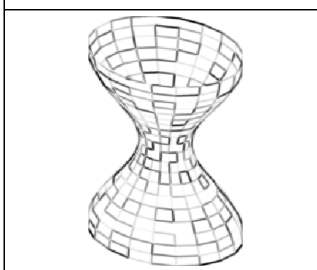
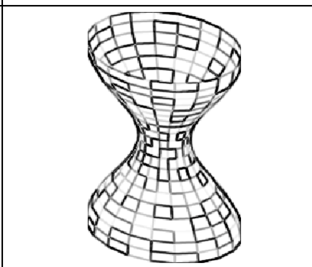
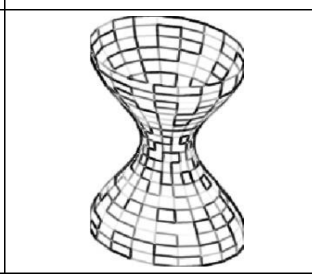
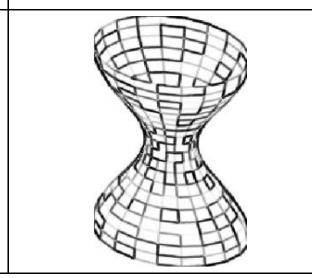
Edge detection method	**Mean**	**StDev**	**Std error mean**
Sobel + IT2FLS	0.9495	0.0112	0.0050
Sobel + GT2FLS	0.9617	0.000114	0.000051

Results T-value = -2.44, P-value=0.036

Based on the statistical values shown in Table 7.13, we can apply a *t* student test and the *t* value is of -2.44 and this has associated a *p* value of 0.036, which means that there is sufficient statistical evidence to state that there is a significant advantage in using the Sobel + GT2 FSs method. In other words, the difference in results shown in Table 7.13 is sufficient to say that generalized type-2 FSs are better than interval type-2 FSs for this process of edge detection.

Finally, the edge detection methods based on Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs were applied on the benchmark images shown in Table 7.2. As was previously mentioned to apply the FOM metric a reference image or ideal edge is required, for this reason to perform these experiments, first the ideal edge was calculated with the traditional Sobel operator, secondly the images were corrupted by Gaussian noise varying the levels between 20 to 50 dbi's, after that the edges were calculated using the proposed GT2 FSs edge detection method. In this analysis of results the T1 and IT2 methodologies were also included.

Table 7.14 Edge detection output using Sobel, Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs

Sobel	Sobel + T1 FSs	Sobel + IT2 FSs	Sobel + GT2 FSs
			
			
			
			
			

The mean of the all the FOM values achieved by each image of Table 7.2 are shown in Table 7.15; in this table we can analyze that the Sobel + IT2 FSs improved on the results achieved by the Sobel + T1 FSs with mean values of (0.9323, 0.9632, 0.9647, 0.9646); besides this, the Sobel + GT2 FSs was better than Sobel + IT2 FSs with mean values of (0.9437, 0.9727, 0.9709, 0.9702). A plot of these results is shown in Figure 7.5 in which we noticed the advantage that exists using GT2 FSs over the IT2 FSs and T1 FSs.

Table 7.15 Edge detection method using Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs applied on real images with Gaussian noise

Edge Detector	FOM VALUES							
	DBI's							
	20		30		40		50	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
Sobel + T1 FSs	0.9198	0.0376	0.9596	0.0139	0.9599	0.0138	0.9607	0.0140
Sobel + IT2 FSs	0.9323	0.0169	0.9632	0.0092	0.9647	0.0096	0.9646	0.0102
Sobel + GT2 FSs	**0.9437**	0.0143	**0.9727**	0.0061	**0.9709**	0.0077	**0.9702**	0.0083

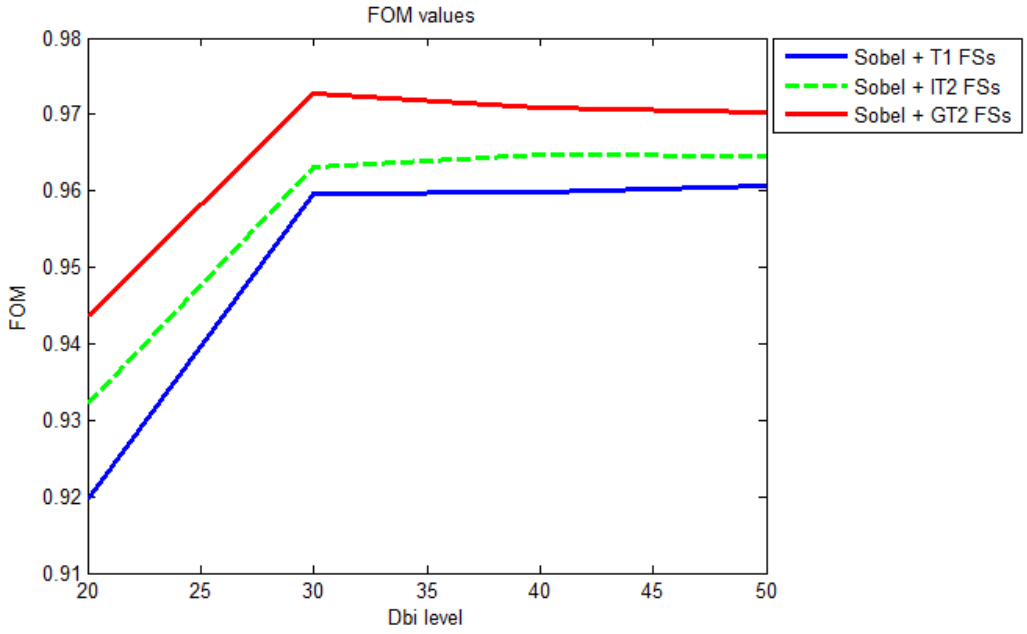


Figure 7.5 Simulation results about edge detection method using Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs applied on real images with Gaussian noise

7.3. Generalized type-2 fuzzy edge detection method using color images

This section provides a comparison of the edge detection accuracy and noise robustness of the proposed image color edge detection method based on the Sobel operator and fuzzy systems (T1 FSs, IT2 FSs and GT2 FSs). The accuracy of these detectors is evaluated on a color image database of synthetic and real images; these images are presented in Tables 7.1 and 7.2, respectively.

The three fuzzy edges detection methods based on type-1, interval type-2 and generalized type-2 FSs were designed under the same conditions; the same inputs, outputs and number of fuzzy rules for achieving a fair comparison. The inputs, outputs and rules were obtained with the methodology described in Section 5.3.

The performance is measured by means of calculating the PSNR in eq. (4.2), MSE in eq. (4.3) and SSIM index in eq. (4.4) respectively. Several simulations are performed, which results are described in detail as follows.

7.3.1 Simulation results using synthetic color images

In the first experiment, we apply the Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs edge detectors to five images of the synthetic images database presented in Table 7.1. In order to evaluate the edge detection quality obtained for these experiments; the PSNR, MSE metric and the SSIM index are used.

In Table 7.16, the MSE, PSNR and SSIM values achieved by the Sobel + T1FSs edge detector are presented.

Table 7.16 Simulation results using Sobel + T1 FSs color edge detection method on synthetic images

Sobel + T1 FSs			
Image	**MSE**	**PSNR**	**SSIM**
Sphere	0.0839	10.7696	0.5457
Doughnut	0.0973	10.1179	0.5743
Peaks	0.0427	13.6952	0.7872
Harmonic	0.0986	10.0614	0.5718
Hat	0.0721	11.4181	0.7200

The simulation results for the Sobel + IT2 FSs fuzzy edge detection method are shown in Table 7.17. Table 7.18 presents the MSE, PSNR and SSIM values obtained by the proposed Sobel + GT2 FSs edge detection method.

Table 7.17 Simulation results using Sobel + IT2 FSs color edge detection on synthetic images

Sobel + IT2 FSs			
Image	**MSE**	**PSNR**	**SSIM**
Sphere	0.0815	10.8961	0.5472
Doughnut	0.0943	10.2542	0.5758
Peaks	0.0415	13.8230	0.7859
Harmonic	0.0956	10.1971	0.5738
Hat	0.0699	11.5560	0.7205

Table 7.18 Simulation results using Sobel + GT2 FSs color edge detection on synthetic images

Sobel + GT2 FSs			
Image	**MSE**	**PSNR**	**SSIM**
Sphere	0.0799	10.9809	0.5562
Doughnut	0.0923	10.3501	0.5926
Peaks	0.0396	14.0195	0.8054
Harmonic	0.0940	10.2720	0.5844
Hat	0.0686	11.6382	0.7324

Based on the tests presented in this section, a comparative analysis was also performed. In Table 7.19, the averages of the measures obtained in the results of Tables 7.16, 7.17 and 7.18 are presented. In Table 7.19, were included also the results achieved by the Sobel, Prewitt and Canny edge detectors.

In the results of Table 7.19, we can note that the MSE (0.0766), PSNR (11.3453), SSIM (0.6406) values obtained with the edge detector based on Sobel + IT2 FSs are improved with respect to the Sobel + T1 FSs. However, the edge detector based on Sobel + GT2 FSs improves on the results achieved by the Sobel + IT2 FSs and the other edge detectors with a MSE value of 0.0749, a PSNR value of 11.4521 and SSIM of 0.6542 respectively. A plot of the behavior of the SSIM values of Table 7.19 is presented in Figure 7.6.

Table 7.19 Comparison results achieved by Sobel, Prewitt, Canny, Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs color edge detectors

Image	MSE	PSNR	SSIM
Sobel	0.0776	11.2087	0.4058
Prewitt	0.0775	11.2108	0.4032
Canny	0.0885	10.5451	0.4745
Sobel + T1 FSs	0.0789	11.2124	0.6398
Sobel + IT2 FSs	0.0766	11.3453	0.6406
Sobel + GT2 FSs	0.0749	11.4521	0.6542

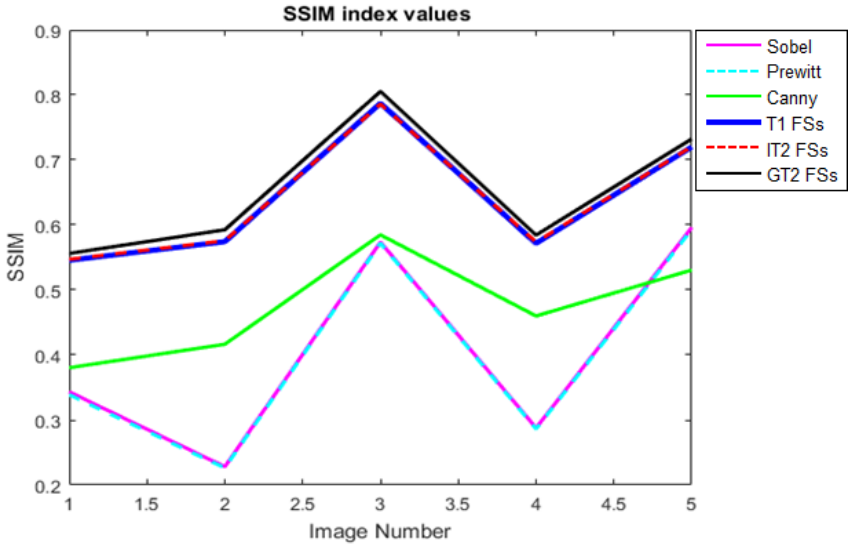


Figure 7.6 SSIM value achieved by the edge detectors Sobel, Prewitt, Canny, Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs

In another test the synthetic images of Table 7.1, were corrupted by different Gaussian noise levels (20, 30, 40 and 50 dbi's). The idea is to measure the performance of edge detectors under uncertainty (noise). The PSNR, MSE and SSIM values obtained by the Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs are shown in Tables 7.20, 7.21 and 7.22 respectively.

For more details of this test; in Table 7.23, a comparative analysis is presented in which the averages of Tables 7.20, 7.21, 7.22 are included and of course the results also achieved by Sobel, Prewitt and Canny edge detectors.

Table 7.20 PSNR, MSE and SSIM values achieved by Sobel + T1 FSs under Gaussian noise in synthetic images

Sobel + T1 FSs												
Image	**20 dbi**			**30 dbi**			**40 dbi**			**50 dbi**		
	MSE	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**
Sphere	0.0669	11.7445	0.3064	0.0599	12.2275	0.3626	0.0609	12.1523	0.4301	0.0615	12.1123	0.5152
Doughnut	0.0652	11.8548	0.3964	0.0686	11.6395	0.4189	0.0699	11.5558	0.4711	0.0706	11.5145	0.5448
Peaks	0.0427	13.6983	0.2549	0.0275	15.6115	0.3097	0.0253	15.9641	0.4431	0.0256	15.9098	0.6865
Harmonic	0.0733	11.3468	0.3471	0.0715	11.4547	0.3777	0.0727	11.3824	0.4324	0.0732	11.3540	0.5356
Hat	0.0558	12.5375	0.2422	0.0507	12.9510	0.2639	0.0504	12.9762	0.3859	0.0505	12.9710	0.6269

Table 7.21 PSNR, MSE and SSIM values achieved by Sobel + IT2 FSs under Gaussian noise in synthetic images

Sobel + IT2 FSs												
Image	**20 dbi**			**30 dbi**			**40 dbi**			**50 dbi**		
	MSE	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**
Sphere	0.0651	11.8637	0.3088	0.0589	12.2981	0.3644	0.0609	12.1523	0.4214	0.0610	12.1449	0.5124
Doughnut	0.0705	11.5203	0.3845	0.0685	11.6403	0.4164	0.0694	11.5889	0.4710	0.0697	11.5654	0.5481
Peaks	0.0418	13.7888	0.2551	0.0502	12.9936	0.2631	0.0257	15.9030	0.4307	0.0256	15.9140	0.6832
Harmonic	0.0699	11.5560	0.3522	0.0704	11.5232	0.3798	0.0716	11.4494	0.4363	0.0722	11.4119	0.5382
Hat	0.0457	13.4055	0.2511	0.0269	15.7031	0.3095	0.0501	12.9985	0.3739	0.0500	13.0108	0.6261

Table 7.22 PSNR, MSE and SSIM values achieved by Sobel + GT2 FSs under Gaussian noise in synthetic images

Sobel + GT2 FSs												
Image	**20 dbi**			**30 dbi**			**40 dbi**			**50 dbi**		
	MSE	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**
Sphere	0.0507	12.9517	0.3425	0.0555	12.5602	0.3667	0.0560	12.5199	0.4304	0.0562	12.5015	0.5168
Doughnut	0.0588	12.3087	0.4128	0.0606	12.1779	0.4361	0.0627	12.0251	0.4818	0.0632	11.9960	0.5622
Peaks	0.0265	15.7672	0.2835	0.0219	16.5904	0.3300	0.0211	16.7571	0.4572	0.0210	16.7830	0.7009
Harmonic	0.0587	12.3137	0.3761	0.0646	11.8949	0.3905	0.0661	11.7981	0.4458	0.0663	11.7837	0.5422
Hat	0.0443	13.5380	0.2588	0.0449	13.4804	0.2759	0.0451	13.4616	0.3651	0.0452	13.4519	0.6231

According with the information presented in Table 7.23, we can notice that under noise levels the Sobel + GT2 FSs achieved better MSE, PSNR and SSIM values than all others the

edge detectors; however, Sobel + IT2 FSs was better than Sobel + T1 FSs. Besides this, the Canny edge detector improved on the results achieved by Sobel and Prewitt. In Figure 7.7, the PSNR values of Table 7.23 for the different dbi levels are presented.

Table 7.23 Comparison results achieved by Sobel, Prewitt, Canny, Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs color edge detectors applied on synthetic images with Gaussian noise

Edge Detector	20 dbi			30 dbi			40 dbi			50 dbi		
	MSE	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**
Sobel	0.1726	7.6333	0.1124	0.1686	7.7355	0.1370	0.1448	8.4041	0.1606	0.0683	11.9208	0.4954
Prewitt	0.1708	7.6794	0.1140	0.1667	7.7851	0.1401	0.1390	8.5837	0.1647	0.0678	11.9604	0.5184
Canny	0.1267	9.1154	0.2185	0.1221	9.2840	0.2372	0.0719	11.6468	0.3937	0.0599	12.3428	0.5759
Sobel + T1 FSs	0.0608	12.2364	0.3094	0.0556	12.7768	0.3466	0.0559	12.8062	0.4325	0.0563	12.7723	0.5818
Sobel + IT2 FSs	0.0586	12.4269	0.3103	0.0550	12.8317	0.3466	0.0555	12.8184	0.4267	0.0557	12.8094	0.5816
Sobel + GT2 FSs	0.0478	13.3758	0.3348	0.0495	13.3408	0.3599	0.0502	13.3124	0.4360	0.0504	13.3032	0.5890

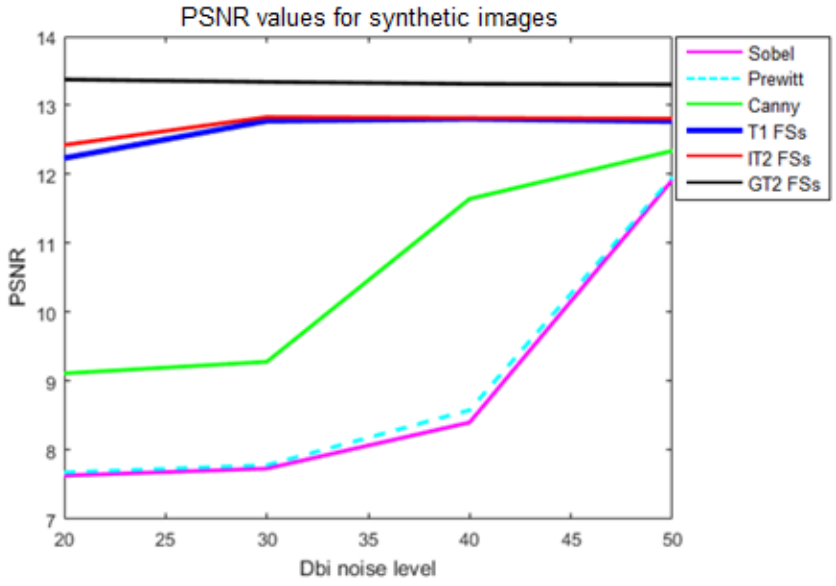


Figure 7.7 PSNR values achieved by Sobel, Prewitt, Canny, T1 FSs, IT2 FSs and GT2 FSs for synthetic images with Gaussian noise

7.3.2 Simulation results using real color images

The three fuzzy edges detection methods based on type-1, interval type-2 and generalized type-2 FSs are now applied to the real color images database of Table 7.2.

In the first test we apply the fuzzy edge detection based on Sobel + T1 FSs. In this test we applied different dbi's noise levels on the real color images. The MSE, PSNR and SSIM measures obtained by the Sobel + T1 FSs using 20, 30, 40 and 50 dbi's values are presented in Table 7.24.

Table 7.24 PSNR, MSE and SSIM values using Sobel + T1 FSs color edge detection in real color images

Sobel + T1 FSs												
Image	**20 dbi**			**30 dbi**			**40 dbi**			**50 dbi**		
	MSE	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**
House	0.0375	14.2621	0.4013	0.0293	15.3266	0.5400	0.0293	15.3314	0.6448	0.0294	15.3222	0.6850
Pepper	0.0254	15.9510	0.3909	0.0163	17.8668	0.6125	0.0146	18.3678	0.6944	0.0149	18.2754	0.6953
Lena	0.0229	16.3926	0.4786	0.0207	16.8481	0.6296	0.0217	16.6291	0.6638	0.0215	16.6687	0.6737
Orangu	0.0358	14.4574	0.5208	0.0362	14.4093	0.6199	0.0375	14.2546	0.6349	0.0376	14.2479	0.6370
Forest	0.0435	13.6115	0.3204	0.0305	15.1581	0.5161	0.0309	15.1070	0.6139	0.0302	15.2023	0.6394
Plane	0.0278	15.5632	0.3582	0.0215	16.6758	0.5065	0.0212	16.7271	0.6388	0.0214	16.6913	0.6748

In another test, the edge detector was applied using Sobel + IT2 FSs, for these experiments we used the same method for the images of Table 7.2 and we applied the same dbi's noise levels. The MSE, PSNR and SSIM values achieved by this fuzzy edge detection method are shown in Table 7.25.

Finally, we considered the experiment using our proposed edge detection method using Sobel + GT2 FSs. The MSE, PSNR and SSIM results are shown in Table 7.26.

Table 7.25 PSNR, MSE and SSIM values using Sobel + IT2 FSs color edge detection in real color images

Sobel + IT2 FSs												
Image	**20 dbi**			**30 dbi**			**40 dbi**			**50 dbi**		
	MSE	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**
House	0.0406	13.9098	0.3908	0.0294	15.3169	0.5400	0.0293	15.3267	0.6438	0.0296	15.2910	0.6804
Pepper	0.0220	16.5759	0.4133	0.0152	18.1882	0.6239	0.0149	18.2787	0.6871	0.0146	18.3608	0.7003
Lena	0.0228	16.4149	0.4791	0.0223	16.5208	0.6114	0.0218	16.6242	0.6676	0.0217	16.6330	0.6702
Orangu	0.0405	13.9212	0.4912	0.0343	14.6482	0.6307	0.0379	14.2085	0.6319	0.0373	14.2805	0.6382
Forest	0.0312	15.0596	0.3616	0.0279	15.5378	0.5266	0.0295	15.2991	0.6195	0.0291	15.3545	0.6442
Plane	0.0281	15.5159	0.3572	0.0215	16.6741	0.5052	0.0212	16.7360	0.6396	0.0214	16.6971	0.6746

Table 7.26 PSNR, MSE and SSIM values using Sobel + GT2 FSs edge detector in real color images

	Sobel + GT2 FSs											
Image	**20 dbi**			**30 dbi**			**40 dbi**			**50 dbi**		
	MSE	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**
House	0.0249	16.0463	0.4570	0.0179	17.4707	0.6266	0.0185	17.3313	0.7093	0.0187	17.2860	0.7454
Pepper	0.0126	18.9830	0.5116	0.0100	19.9892	0.6878	0.0097	20.1345	0.7495	0.0094	20.2761	0.7631
Lena	0.0093	20.3278	0.6387	0.0120	19.1920	0.7304	0.0128	18.9185	0.7617	0.0127	18.9772	0.7701
Cangu	0.0157	18.0291	0.6536	0.0204	16.8970	0.7278	0.0199	17.0186	0.7572	0.0204	16.9108	0.7560
Forest	0.0165	17.8134	0.4281	0.0127	18.9761	0.6100	0.0126	18.9830	0.7455	0.0128	18.9355	0.7760
Plane	0.0188	17.2549	0.4111	0.0131	18.8164	0.6077	0.0116	19.3714	0.7588	0.0127	18.9777	0.7796

The averages of the simulation results obtained in Tables 7.24, 7.25 and 7.26 are presented in Table 7.27; moreover, in these tables are also included the results achieved by the Sobel, Prewitt and Canny edge detectors.

In the results of Table 7.27 we can notice that the proposed method based on Sobel + GT2 FSs is better than Sobel + IT2 FSs for all cases of noise levels; on the other hand, the MSE, PSNR and SSIM values obtained by Sobel + IT2 FSs are better than Sobel + T1 FSs. With regards to the other edge detectors, in this case the Sobel, Prewitt and Canny the performance was very similar; however these are below the fuzzy edge detectors. The performance of the

PSNR values described in Table 7.27 are shown in Figure 7.8, and we can appreciate that the PSNR values achieved by the proposed Sobel + GT2FSs edge detector are the highest.

Table 7.27 Comparison results achieved by Sobel, Prewitt, Canny, Sobel + T1FSs, Sobel + IT2FSs and Sobel + GT2FSs color edge detection in real color images with Gaussian noise

Edge Detector	20 dbi			30 dbi			40 dbi			50 dbi		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Sobel	0.0478	13.2927	0.1062	0.0498	13.1087	0.1068	0.0501	13.0843	0.1066	0.0501	13.0802	0.1065
Prewitt	0.0479	13.2894	0.1053	0.0496	13.1289	0.1059	0.0498	13.1082	0.1055	0.0499	13.1020	0.1054
Canny	0.0505	13.0861	0.2114	0.0503	13.1247	0.2256	0.0506	13.0999	0.2245	0.0508	13.0869	0.2241
Sobel + T1 FSs	0.0322	15.0396	0.4117	0.0258	16.0475	0.5708	0.0259	16.0695	0.6485	0.0258	16.0680	0.6675
Sobel + IT2 FSs	0.0309	15.2329	0.4155	0.0251	16.1477	0.5729	0.0258	16.0789	0.6483	0.0256	16.1028	0.6680
Sobel + GT2 FSs	0.0163	18.0758	0.5167	0.0144	18.5569	0.6650	0.0142	18.6262	0.7470	0.0144	18.5605	0.7650

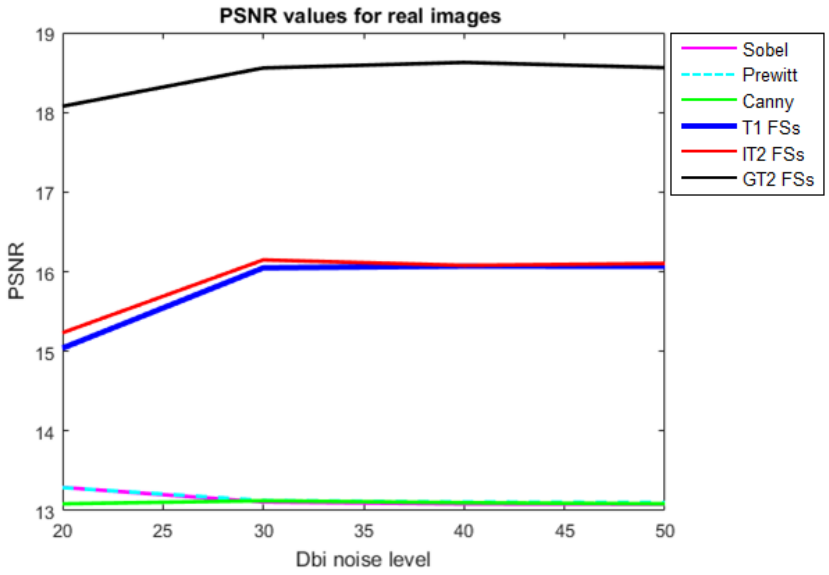


Figure 7.8 PSNR value achieved by Sobel, Prewitt, Canny, T1 FSs, IT2 FSs and GT2 FSs for real images with Gaussian noise

Based on the PNSR values shown in Table 7.27, we can apply a Wilcoxon's rank-sum test. In the first evaluation we considered the PSNR values obtained by the Sobel + IT2FSs and Sobel + T1FSs; this has an associated p value of 0.1; therefore, the hypothesis that the interval type-2 is

better than type-1 is rejected because there is not sufficient statistical evidence to support this statement.

In another test we considered the PSNR values achieved by the Sobel + GT2FSs and Sobel + T1FSs; with a p value of 0.0143. Additionally, we applied the Wilcoxon's rank-sum test on the results obtained by the Sobel + GT2FSs and Sobel + IT2FSs; in the same manner the p value was of 0.0143. Both results imply that there is sufficient evidence to state that there is a significant advantage in using the generalized type-2 Sobel method. In other words, the difference in results shown in Table 7.27 is sufficient statistical to say that the GT2 FS is better than interval type-2 and type-1 fuzzy logic for this process of edge detection.

The visual edge detection for the Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + IT2 FSs methods are presented in Tables 7.28 and 7.29. Of course, graphically we cannot clearly appreciate the difference in the detected edges and for this reason the MSE and PSNR measures were applied, but the statistical tests support the conclusion that generalized type-2 fuzzy logic can outperform interval type-2 and type-1 in edge detection for color images under different levels of noise in the process.

Table 7.28 Color edge detection using the Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs methods applied on real images

Sobel + T1 FSs	**Sobel + IT2 FSs**	**Sobel + GT2 FSs**
		

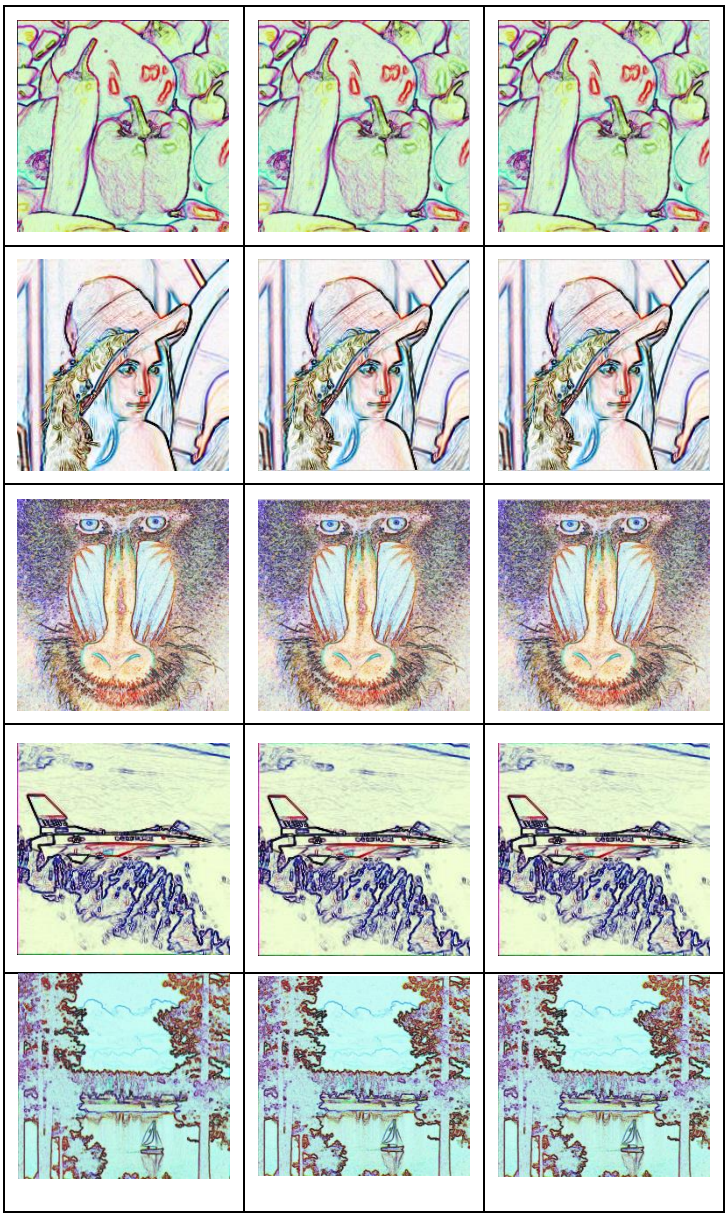
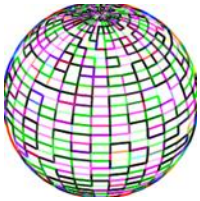
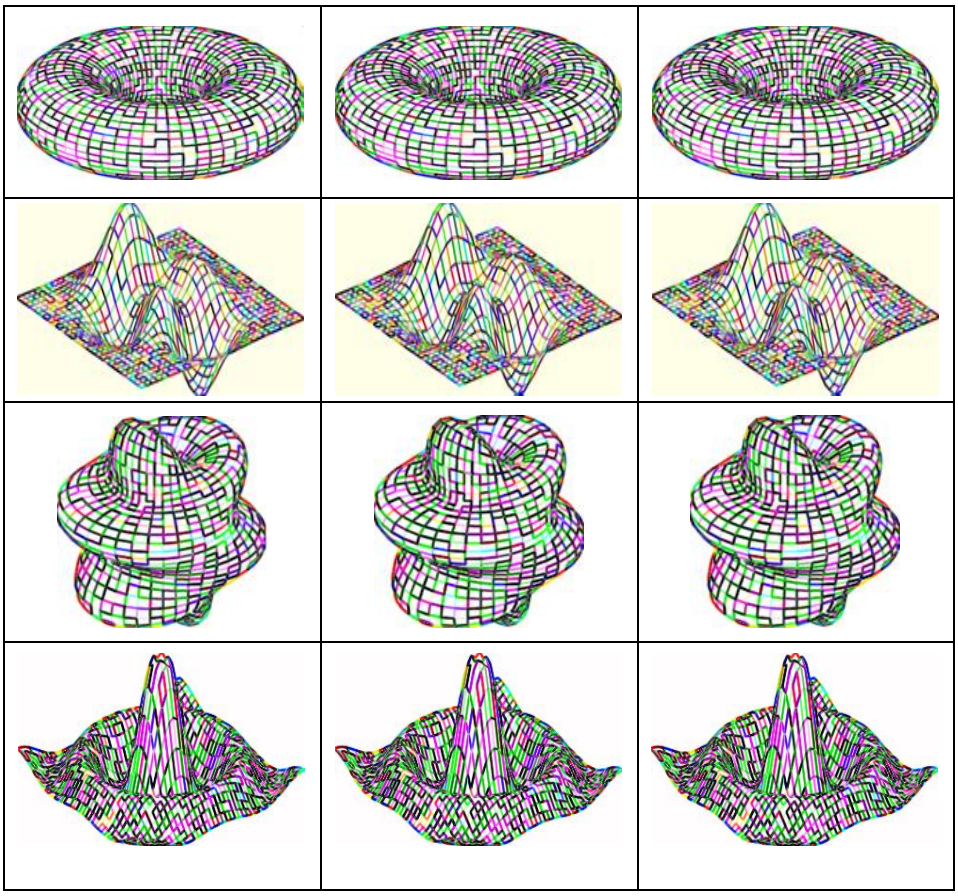


Table 7.29 Color edge detection using the Sobel + T1 FSs, Sobel + IT2 FSs and Sobel + GT2 FSs methods applied on synthetic images

Sobel + T1 FSs	Sobel + IT2 FSs	Sobel + GT2 FSs
		



7.4. Edge detection method applied on GT2 fuzzy images

This section shows the simulation results obtained with the edge detection method, which is applied on fuzzy images; all the details about the implementation of this methodology were previously described in Section 5.4; therefore, a comparative analysis is presented where the image fuzzification process is implemented using T1, IT2 and GT2 membership functions. These methods were applied on the synthetic images illustrated in Table 7.1 and the accuracy achieved by the methods is measured with the FOM metric and the results are explained below.

Table 7.30 shows the FOM values achieved by the edge detection method in which are included the traditional MG, T1 FSs, IT2 FSs and GT2 FSs; according with these results we noticed that the mean value of (0.9140) achieved by the IT2 fuzzy edge detection method was

better than the mean value of the T1 fuzzy edge detection; nevertheless, the GT2 improved on the results obtained by the IT2 with a better mean value of 0.9485.

Table 7.30 FOM values achieved by the MG, T1 FSs, IT2 FSs and GT2 FSs on the fuzzy images

Figure Number	FOM Values			
	Edge Detection method			
	MG	T1 FSs	IT2 FSs	GT2 FSs
1	0.8724	0.8724	0.9296	0.9562
2	0.8605	0.8605	0.8378	0.9464
3	0.8982	0.8982	0.9476	0.9426
4	0.8636	0.8636	0.9561	0.9526
5	0.8528	0.8528	0.8758	0.9474
6	0.8531	0.8531	0.8866	0.9450
7	0.8676	0.8676	0.9502	0.9439
8	0.8837	0.8837	0.9527	0.9442
9	0.7921	0.7921	0.8477	0.9558
10	0.9195	0.9195	0.9557	0.9511
Mean	0.8663	0.8663	0.9140	0.9485

After obtaining the ten experiments using the different fuzzy edge detection methods, the *t* student tests are performed, because it is necessary to find out if a significant difference exist when a GT2 fuzzy edge detection is applied. The *t* student tests are presented in Tables 7.31, 7.32 and 7.33 respectively. The value *t* student test uses a distribution to prove if a hypothesis is null or not, the "*p*" value is the probability that the decision of the correct hypothesis is not valid.

In Tables 7.31 and 7.32, the test shows that there is statistical evidence to say that there is significant difference in the results achieved by the GT2 FSs with respect to the T1 and IT2 FSs (with more than 95% confidence). Otherwise, in the tests shown in Table 7.33 significant differences were also observed in the results obtained with the IT2 FSs with regards to T1 FSs.

Table 7.31 Results for the *t* student test simulation for edge detection method based on IT2 and GT2 FSs

Edge detection method	N	Mean	StDev	Std error mean
IT2 FSs	10	0.9140	0.0473	0.0150
GT2 FSs	10	0.9485	0.0050	0.0016

Results T-value = -2.30, P-value=0.024

Table 7.32 Results for the *t* student test for the edge detection method based on T1 and GT2 FSs

Edge detection method	N	Mean	StDev	Std error mean
T1 FSs	10	0.8663	0.0335	0.011
GT2 FSs	10	0.9485	0.0050	0.0016

Results T-value = -7.67, P-value=0.000

Table 7.33 Results for the *t* student test for the edge detection method based on T1 and IT2 FSs

Edge detection method	N	Mean	StDev	Std error mean
T1 FSs	10	0.8663	0.0335	0.011
IT2 FSs	10	0.9140	0.0473	0.0150

Results T-value = -2.60, P-value=0.010

7.5. Simulation results achieved by the face recognition system

This section provides a comparison of the recognition rates achieved by the face recognition system when different fuzzy edge detectors were applied.

In the experimental results several edge detectors were analyzed in which the Sobel operator, Sobel combined with T1 FSs, IT2 FSs and GT2 FSs are included. Besides these, the experiments were also evaluated without using any edge detector.

The tests were executed using the ORL [66], the Cropped Yale [67-68] and FERET [69] databases; an example of these face databases is shown in Table 7.34. The parameters used in the monolithic neural network are described in Section 6.2. Otherwise, the training and testing set that we considered in the experiments are presented in Table 6.1, and these values depend on the database size used.

Table 7.34 Face databases

Database	Examples
ORL	
Cropped Yale	
FERET	

In the first test, the face recognition system was performed using the Sobel GT2 fuzzy edge detector. This test was applied on the ORL data set. The mean rate, standard deviation and max rate values achieved by the system are shown in Table 7.35.

Table 7.35 Recognition rate for the ORL database using Sobel GT2 fuzzy edge detector

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
Sobel + GT2 FSs	93.67	0.01	96.50

In another test, the system also considered the Cropped Yale database. The numeric results for this experiment are presented in Table 7.36.

Table 7.36 Recognition rate for the Cropped Yale database using GT2 fuzzy edge detection

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
Sobel + GT2 FSs	98.11	0.03	100

In another test the recognition system was applied to the FERET database, the results for this simulation are presented in Table 7.37.

Table 7.37 Recognition rate for FERET database using GT2 fuzzy edge detection

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
Sobel + GT2 FSs	87.50	0.08	92.50

As part of the goals of this work, the recognition rate values achieved by the system when the GT2 fuzzy edge detector is used, were compared with the results obtained when the neural network is trained without edge detection; also, to the Sobel operator, the Sobel edge detector combined with T1 and IT2 FSs. The results for the ORL, Cropped Yale and FERET databases are shown in Tables 7.38, 7.39 and 7.40 respectively.

The results obtained on the ORL database are presented in Table 7.38; so, in this Table we can notice that the mean rate value is better when the Sobel GT2 fuzzy edge detector is applied with a percentage value of 93.67 and max rate of 96.50.

Table 7.38 Recognition rate for the ORL database

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
None	2.59	0.0022	5.00
Sobel	2.50	0	2.50
Sobel + T1 FSs	87.25	0.03	91.25
Sobel + IT2 FSs	90.75	0.04	95.00
Sobel + GT2 FSs	93.67	0.01	96.50

Otherwise, the results achieved when the Cropped Yale database is used are shown in Table 7.39. In this Table we observed that the best performance (mean rate) of the neural network is obtained when the Sobel + GT2 FSs was applied with a mean percentage rate of 98.11; nevertheless, we can notice than the max rate values obtained by all the fuzzy edge detectors are of 100.

Table 7.39 Recognition rate for the Cropped Yale database

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
None	2.83	0.00	6.57
Sobel	2.63	0	2.63
Sobel + T1 FSs	97.52	0.02	100
Sobel + IT2 FSs	97.70	0.03	100
Sobel + GT2 FSs	98.11	0.03	100

Finally the recognition rates achieved by the system when the FERET database is used are presented in Table 7.40. These results also show that the mean rate value is better when the Sobel + Sobel + GT2 FSs is applied, with a mean percentage rate of 87.50 and max rate of 92.50.

Table 7.40 Recognition rate for FERET database

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
None	0.8	0.02	1.00
Sobel	1.0	0.03	1.50
Sobel + T1 FSs	82.77	0.68	83.78
Sobel + IT2 FSs	84.46	0.32	87.84
Sobel + GT2 FSs	87.50	0.08	92.50

Based on the simulation results presented in Tables 7.38, 7.39 and 7.40 we can conclude that the edge detection based on GT2 FS represent a good way to improve the performance in face recognition systems.

In general, the results achieved in the simulations were better when the fuzzy edge detection was applied specially with the Sobel + GT2 FSs; since the results were very low when the monolithic neural network was applied without any edge detection method; even so, when the traditional Sobel edge detector was applied.

8. Conclusions and future work

In summary in this work, multiple approaches for edge detection edge detection method based on generalized type-2 fuzzy logic has been proposed.

In the results presented in Section 7.1 when the proposed method based on MG + GT2 FSs is applied, better results are obtained, and the main reason is that the uncertainty in edge detection is modeled more closely with generalized type 2 fuzzy logic. This is in contrast to the traditional morphological gradient method, or even its type-1 and interval type-2 versions that are not able to cope with higher degrees of uncertainty.

Otherwise, the experimental results presented in Sections 7.2 and 7.3 show that the use of generalized type-2 fuzzy inference systems can improve the performance in edge detection with respect to interval type-2 fuzzy systems. After evaluating the detected edges with the metrics Pratt's FOM, MSE, PSNR and SSIM we can notice the advantages of the GT2 FSs to model uncertainty in the gray tone values for the edges. The Sobel + GT2 FSs method obtains better edges than when using the Sobel + IT2 FSs on grayscale and color images, because it preserves more details of the original image; on the other hand, the Sobel + IT2 FSs can improve the performance with respect to the Sobel + T1 FSs.

In Section 7.4, a novel edge detection method using GT2 fuzzy images was presented. According with the results achieved in this section we can conclude that the proposed method is a good way to model the uncertainty when the image are corrupted by noise. Additionally, in this methodology, the fuzzy inference system is no necessary because we only used the fuzzification process, and this minimizes the computing time; so, this methodology can be used in real applications on image processing systems.

Based on the results presented in Section 7.5 when the edge detection was applied in the

image databases before the training phase of the monolithic neural network, we notice that the proposed edge detector based on GT2 FSs improved on the results achieved by the IT2 FSs and T1 FSs. The best recognition rate obtained by the neural network was when the GT2 fuzzy edge detector was applied; so we can conclude that the edge detection based on GT2 FS represents a good way to improve the performance in a face recognition system.

In general, this work leads to the conclusion that the use of generalized type-2 fuzzy systems can be a good choice when there is a high level of uncertainty in the problem. In other words, generalized type-2 fuzzy logic allows for better modeling of uncertainty, because it gives more degrees of freedom in comparison to interval type-2 and type-1 fuzzy logic. The complex nature of the uncertainty encountered in the real world indicates that generalized type-2 is needed in real-world devices and applications, in particular in the image processing area that is the case study in this thesis, because the devices that capture digital images are always exposed to external interference adding high noise levels or uncertainty to the images.

Based on the results of this thesis we can notice the advantages of using the generalized type-2 FSs, allowing us for greater control of the uncertainty that may exist in image processing systems and to help achieve satisfactory results. However, the proposed method may not be applied for real time applications tasks at this moment because edge detection in images is required in a very short time in real world applications. So, our next goal will be improving each step of our method to reduce the complexity and the computing time. In particular, we would also need exploiting parallel or distributed computing for achieving faster results for generalized type-2 fuzzy logic.

In future work, we are planning to implement optimization techniques that would help finding the optimal parameter values of the membership functions and the optimal number of

alpha planes for the automatic implementation of the proposed method. Finally, we look forward to improve the generalized type-2 fuzzy logic algorithms to consider other areas of application.

9. References

[1] V. Torre, T. A. Poggio, On Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, no. 6. 1986, pp. 147–163.

[2] M. Setayesh, M. Zhang, and M. Johnston, "Effects of static and dynamic topologies in Particle Swarm Optimisation for edge detection in noisy images," in *2012 IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.

[3] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 2. 1986, pp. 679–698.

[4] T. Shimada, F. Sakaida, H. Kawamura, and T. Okumura, "Application of an edge detection method to satellite images for distinguishing sea surface temperature fronts near the Japanese coast," Remote Sensing of Environment, vol. 98, no. 1, pp. 21–34, 2005.

[5] A. A. Goshtasby, 2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications. 2005, pp. 34–39.

[6] A. Kazerooni, A. Ahmadian, N. D. Serej, H. S. Rad, H. Saberi, H. Yousefi, and P. Farnia, "Segmentation of brain tumors in MRI images using multi-scale gradient vector flow," in Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE, 2011, pp. 7973–7976.

[7] B. Siliciano, L. Sciavicco, L. Villani, and G. Oriolo, Robotics: Modelling, Planning and Control. 2010, pp. 415– 418.

[8] Z. Hocenski, S. Vasilic, and V. Hocenski, "Improved Canny Edge Detector in Ceramic Tiles Defect Detection," in IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on, 2006, pp. 3328 –3331.

[9] C. Tao, W. Thompson, and J. Taur, "A fuzzy if-then approach to edge detection," Fuzzy Systems, Second IEEE International Conference on, pp. 1356–1360, 1993.

[10] Z. Talai and A. Talai, "A fast edge detection using fuzzy rules," 2011 International Conference on Communications, Computing and Control Applications (CCCA), pp. 1–5, Mar. 2011.

[11] L. Hu, H. D. Cheng, and M. Zhang, “A high performance edge detector based on fuzzy inference rules,” Information Sciences, vol. 177, no. 21, pp. 4768–4784, Nov. 2007.

[12] T. Pham and L. Van Vliet, “Blocking artifacts removal by a hybrid filter method,” Proc. ASCI, pp. 249–253, 2005.

[13] O. P. Verma and R. Sharma, “An optimal edge detection using universal law of gravity and ant colony algorithm,” 2011 World Congress on Information and Communication Technologies, pp. 507–511, Dec. 2011.

[14] P. Agrawal, S. Kaur, H. Kaur, and A. Dhiman, “Analysis and Synthesis of an Ant Colony Optimization Technique for Image Edge Detection,” 2012 International Conference on Computing Sciences, pp. 127–131, Sep. 2012.

[15] H. Bustince, E. Barrenechea, M. Pagola, and J. Fernandez, “Interval-valued fuzzy sets constructed from matrices: Application to edge detection,” Fuzzy Sets and Systems, vol. 160, no. 13, pp. 1819–1840, Jul. 2009.

[16] O. Mendoza, P. Melin, and G. Licea, “A New Method for Edge Detection in Image Processing Using Interval Type-2 Fuzzy Logic,” 2007 IEEE International Conference on Granular Computing (GRC 2007), pp. 151–151, Nov. 2007.

[17] O. Mendoza, P. Melin, and G. Licea, “Interval type-2 fuzzy logic for edges detection in digital images,” International Journal of Intelligent Systems (IJIS), vol. 24, no. 11, pp. 1115–1133, 2009.

[18] P. Melin, O. Mendoza, and O. Castillo, “An improved method for edge detection based on interval type-2 fuzzy logic,” Expert Systems with Applications, vol. 37, no. 12, pp. 8527–8535, Dec. 2010.

[19] R. Biswas and J. Sil, “An Improved Canny Edge Detection Algorithm Based on Type-2 Fuzzy Sets,” Procedia Technology, vol. 4, pp. 820–824, Jan. 2012.

[20] A. N. Evans and X. U. Liu, "A morphological gradient approach to color edge detection," in IEEE Transactions on Image Processing, vol. 15, no. 6, pp. 1454-1463, 2006.

[21] I. Sobel, “Camera Models and Perception,” Ph.D. thesis, Stanford University, Stanford, CA, 1970.

[22] R. Kirsch, “Computer determination of the constituent structure of biological images,” Computers and Biomedical Research, vol. 4, pp. 315–328, 1971.

[23] J. M. S. Prewitt, “Object enhancement and extraction,” Picture Analysis and Psychopictorics, Academic Press, New York, NY, pp. 75–149, 1970.

[24] M. A. Soyturk, A. Basturk, M. E. Yuksel, "A Novel Fuzzy Filter for Speckle Noise Removal," Turkish Journal of Electrical Engineering and Computer Sciences, vol. 22, pp. 1367-1381, 2014

[25] M. T. Yildirim, A. Basturk and M. E. Yuksel, "Impulse Noise Removal from Digital Images by A Detail-Preserving Filter Based on Type-2 Fuzzy Logic," IEEE Transactions on Fuzzy Systems, vol. 16, no. 4, pp. 920-928, 2008.

[26] M. E. Yuksel, A. Basturk and E. Besdok, "Detail-Preserving Restoration of Impulse Noise Corrupted Images by A Switching Median Filter Guided by A Simple Neuro-Fuzzy Network," EURASIP Journal on Advances in Signal Processing, vol. 16, pp.2451-2461, 2004

[27] M. E. Yuksel and A. Basturk, "Efficient Removal of Impulse Noise from Highly Corrupted Digital Images by A Simple Neuro-Fuzzy Operator," AEU-International Journal of Electronics and Communications, vol. 57, no. 3, 214-219, 2003.

[28] L. A. Zadeh, "Fuzzy sets," Information and Control, Vol. 8, pp. 338-353, 1965.

[29] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," Information Sciences, vol. 8, no. 3, pp. 199-249, 1975.

[30] L. A. Zadeh, "Fuzzy Logic," Computer, vol. 1, no. 4, pp. 83-93, 1988.

[31] Q. Liang and J. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," IEEE Transactions Fuzzy Systems, vol. 8, no. 5, pp. 535–550, 2000.

[32] J. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, 2001.

[33] N.N. Karnik, J.M. Mendel and Q. Liang, "Type-2 fuzzy logic systems," Fuzzy Systems, IEEE Transactions on, vol.7, no.6, pp.643-658, 1999.

[34] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," IEEE Trans. Fuzzy Syst., vol. 10, no. 2, pp. 117–127, 2002.

[35] J.M. Mendel, Advances in type-2 fuzzy sets and systems, Information Sciences, vol. 177, no. 1, pp. 84-110, 2007.

[36] C. Wagner and H. Hagras, "Toward general type-2 fuzzy logic systems based on zSlices," Fuzzy Syst. IEEE Trans., vol. 18, no. 4, pp. 637–660, 2010.

[37] C. Wagner and H. Hagras, "Employing zSlices based general type-2 fuzzy sets to model multi level agreement," 2011 IEEE Symp. Adv. Type-2 Fuzzy Log. Syst., pp. 50–57, Apr. 2011.

[38] J. M. Mendel, L. Fellow, F. Liu, and D. Zhai, "α -Plane Representation for Type-2 Fuzzy Sets : Theory and Applications," IEEE Trans. Fuzzy Syst., vol. 17, no. 5, pp. 1189–1207, 2009.

[39] J. M. Mendel, "Comments on alpha-plane representation for type-2 fuzzy sets: theory and applications," IEEE Trans. Fuzzy Syst., vol. 18, no. 1, pp. 229–230, 2010.

[40] F. Liu, "An efficient centroid type-reduction strategy for general type-2 fuzzy logic system," Inf. Sci. , vol. 178, no. 9, pp. 2224–2236, 2008.

[41] M. A. Sanchez, O. Castillo, and J. R. Castro, "Generalized Type-2 Fuzzy Systems for controlling a mobile robot and a performance comparison with Interval Type-2 and Type-1 Fuzzy Systems," Expert Syst. Appl., vol. 42, no. 14, pp. 5904–5914, Aug. 2015.

[42] P. Melin, C. Gonzalez, J. R. Castro, O. Mendoza, and O. Castillo, "Edge Detection Method for Image Processing Based on Generalized Type-2 Fuzzy Logic," IEEE Trans. Fuzzy Syst., vol. 22, no. 6, pp. 1515–1525, 2014.

[43] Tai K., A. El-Sayed, M. Biglarbegian, C. I. Gonzalez, O. Castillo, S. Mahmud, "Review of Recent Type-2 Fuzzy Controller Applications", Algorithms, vol. 9, no. 2, pp. 1-19, 2016.

[44] D. Zhai and J. M. Mendel, "Uncertainty measures for general Type-2 fuzzy sets," Inf. Sci., vol. 181, no. 3, pp. 503–518, 2011.

[45] J. M. Mendel, "General Type-2 Fuzzy Logic Systems Made Simple: A Tutorial," in IEEE Transactions on Fuzzy Systems, vol. 22, no. 5, pp. 1162-1182, Oct. 2014.

[46] X. Liu, J. M. Mendel, and D. Wu, "Study on enhanced Karnik–Mendel algorithms: Initialization explanations and computation improvements," Inf. Sci., vol. 184, no. 1, pp. 75–91, Feb. 2012.

[47] J. M. Mendel and L. Fellow, "On KM Algorithms for Solving Type-2 Fuzzy Set Problems," IEEE Trans. Fuzzy Syst., vol. 21, no. 3, pp. 426–446, 2013.

[48] D. Wu, "Approaches for Reducing the Computational Cost of Interval Type-2 Fuzzy Logic Systems: Overview and Comparisons," IEEE Trans. Fuzzy Syst., vol. 21, no. 1, pp. 80–99, Feb. 2013.

[49] A. C. Bovik, The Essential Guide to Image Processing, pp. 498–500, 2009.

[50] Y. Becerikli and T. M. Karan, "A New Fuzzy Approach for Edge Detection," in computational intelligence and bioinspired systems. Berlin: LNCS, Springer Verlag, pp. 943–951, 2005.

[51] C. Jacques and A. Bauchspiess, "Fuzzy inference system applied to edge detection in digital images," in V Brazilian Conference on Neural Networks, Brazil, 2001.

[52] O. Mendoza, P. Melin, and G. Licea, "A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks and the Sugeno integral," Inf. Sci., vol. 179, no. 13, pp. 2078–2101, Jun. 2009.

[53] A.D. Kulkarni, Computer Vision and Fuzzy Neural Systems, Prentice Hall, 2001.

[54] W.K. Pratt, Digital Image Processing, 2nd Edition. John Wiley & Sons, Inc.: New York, 1991.

[55] O. Mendoza, P. Melin and O. Castillo, "Neural networks recognition rate as index to compare the performance of fuzzy edge detectors," in Neural Net-works (IJCNN), The 2010 International Joint Conference on, pp. 1-6, 2010.

[56] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, "Digital Image Processing using Matlab," in Prentice-Hall, 2004.

[57] F. Perez-Ornelas, O. Mendoza, P. Melin and J. R. Castro, "Interval type-2 fuzzy logic for image edge detection quality evaluation," in 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), 2012, no. 1, pp. 1–6.

[58] I. Abdou and W. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," Proceedings of the IEEE, vol. 67, no. 5, pp. 753–763, 1979.

[59] F. Perez-Ornelas, O. Mendoza, P. Melin, J.R. Castro, A. Rodriguez-Diaz, O. Castillo, "Fuzzy Index to Evaluate Edge Detection in Digital Images," PLoS ONE, vol. 10, no.6, pp. 1-19, 2015.

[60] S. Schulte, V. De Witte and E.E. Kerre, "A Fuzzy Noise Reduction Method for Color Images", IEEE Transactions on Image Processing, vol. 16, no. 5, pp. 1425–1436, 2007.

[61] Z. Wang, L. Lu and A. C. Bovik, "Video quality assessment based on structural distortion measurement", Signal Processing: Image Communication, vol. 19, no. 2, pp. 121-132, 2004.

[62] Z. Wang and Q. Li, "Information Content Weighting for Perceptual Image Quality Assessment," in IEEE Transactions on Image Processing, vol.20, no.5, pp. 1185-1198, 2011.

[63] J. Kacprzyk, Multistage Fuzzy Control, Wiley, Chichester, 1997.

[64] Z. Xu and M. Xia, "Distance and similarity measures for hesitant fuzzy sets," Inf. Sci., vol. 181, no. 11, pp. 2128-2138, 2011.

[65] E. Szmidt and J. Kacprzyk, "Distances between intuitionistic fuzzy sets," Fuzzy Sets and Systems, vol. 114, no. 3, pp. 505-518, 2000.

[66] "The USC-SIPI Image Database.," 1977. [Online]. Available: http://sipi.usc.edu/database/.

[67] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 643-660, 2001.

[68] K. C. Lee, J. Ho and D. Kriegman, "Acquiring Linear Subspaces for Face Recognition under Variable Lighting," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 684-698, 2005.

[69] P. J. Phillips, H. Moon, S. A. Rizvi and P. J Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no.10, pp. 1090–1104, 2000.

Appendix A

A.1 Function to calculate morphological gradient edge detection

```
function [d1,d2,d3,d4]=morphologicalgradient(image)
[r,c]=size(image);
    for x=2:r-1
        for y=2:c-1

            z1=double(image(x-1,y-1));
            z2=double(image(x-1,y));
            z3=double(image(x-1,y+1));
            z4=double(image(x,y-1));
            z5=double(image(x,y));
            z6=double(image(x,y+1));
            z7=double(image(x+1,y-1));
            z8=double(image(x+1,y));
            z9=double(image(x+1,y+1));

            d1(x,y)=sqrt((z5-z1)^2+(z5-z9)^2);
            d2(x,y)=sqrt((z5-z2)^2+(z5-z8)^2);
            d3(x,y)=sqrt((z5-z3)^2+(z5-z7)^2);
            d4(x,y)=sqrt((z5-z4)^2+(z5-z6)^2);
        end
    end
end
```

A.2. Function to define the FIS for the MG + GT2 FSs

```
function fis=deltaedgesgt2(clow,cmedium,chigh,cfondo,cborde,fpdelta,fpedges)

NumMf = [3 3 3 3 3];
MfType =
str2mat('gaussmgausstype2','gaussmgausstype2','gaussmgausstype2','gaussmgauss
type2', 'gaussmgausstype2');

fd=fpdelta/2;
fe=fpedges/2;
sd=chigh./4;
sefondo=abs(cfondo-cborde)/2;
seborde=abs(cfondo-cborde)/2;
rho=0.6;

MfParams{1}=[sd(1) clow(1)-fd(1) clow(1)+fd(1) rho;sd(1) cmedium(1)-fd(1)
cmedium(1)+fd(1) rho;sd(1) chigh(1)-fd(1) chigh(1)+fd(1) rho];
```

```
MfParams{2}=[sd(2) clow(2)-fd(2) clow(2)+fd(2) rho;sd(2) cmedium(2)-fd(2)
cmedium(2)+fd(2) rho;sd(2) chigh(2)-fd(2) chigh(2)+fd(2) rho];

MfParams{3}=[sd(3) clow(3)-fd(3) clow(3)+fd(3) rho;sd(3) cmedium(3)-fd(3)
cmedium(3)+fd(3) rho;sd(3) chigh(3)-fd(3) chigh(3)+fd(3) rho];

MfParams{4}=[sd(4) clow(4)-fd(4) clow(4)+fd(4) rho;sd(4) cmedium(4)-fd(4)
cmedium(4)+fd(4) rho;sd(4) chigh(4)-fd(4) chigh(4)+fd(4) rho];

MfParams{5}=[sefondo cfondo-fe cfondo+fe rho;seborde cborde-fe cborde+fe
rho];

fis =
newgfistype2('edges','mamdani','singleton','min','max','min','max','centroid'
);

%
fis = addgvartype2(fis,'input','D1',[-100 300]);
fis = addgvartype2(fis,'input','D2',[-100 300]);
fis = addgvartype2(fis,'input','D3',[-100 300]);
fis = addgvartype2(fis,'input','D4',[-100 300]);
fis = addgvartype2(fis,'output','E',[min([cfondo cborde]) max([cfondo
cborde])]);

%
fis = addgmftype2(fis,'input',1,'LOW',MfType(1,:),MfParams{1}(1,:));
fis = addgmftype2(fis,'input',1,'MIDDLE',MfType(1,:),MfParams{1}(2,:));
fis = addgmftype2(fis,'input',1,'HIGH',MfType(1,:),MfParams{1}(3,:));

%
fis = addgmftype2(fis,'input',2,'LOW',MfType(2,:),MfParams{2}(1,:));
fis = addgmftype2(fis,'input',2,'MIDDLE',MfType(2,:),MfParams{2}(2,:));
fis = addgmftype2(fis,'input',2,'HIGH',MfType(2,:),MfParams{2}(3,:));

%
fis = addgmftype2(fis,'input',3,'LOW',MfType(3,:),MfParams{3}(1,:));
fis = addgmftype2(fis,'input',3,'MIDDLE',MfType(3,:),MfParams{3}(2,:));
fis = addgmftype2(fis,'input',3,'HIGH',MfType(3,:),MfParams{3}(3,:));


fis = addgmftype2(fis,'input',4,'LOW',MfType(4,:),MfParams{4}(1,:));
fis = addgmftype2(fis,'input',4,'MIDDLE',MfType(4,:),MfParams{4}(2,:));
fis = addgmftype2(fis,'input',4,'HIGH',MfType(4,:),MfParams{4}(3,:));

fis = addgmftype2(fis,'output',1,'BACKGROUND',MfType(5,:),MfParams{5}(1,:));
fis = addgmftype2(fis,'output',1,'EDGE',MfType(5,:),MfParams{5}(2,:));


%% Define the fuzzy rules
ruleList= [ 3   3   3   3   2  1   2;
            2   2   2   2   2  1   2;
            1   1   1   1   1  1   1
    ];

fis=addgruletype2(fis,ruleList);

%% Plot the input and output GT2 MFs
figure
```

```
set(gcf,'Color',[1,1,1])
plotgmftype2(fis,'input',1); grid on
xlabel('D1');
figure
set(gcf,'Color',[1,1,1])
plotgmftype2(fis,'input',2); grid on
xlabel('D2');
figure
set(gcf,'Color',[1,1,1])
plotgmftype2(fis,'input',1); grid on
xlabel('D3');
figure
set(gcf,'Color',[1,1,1])
plotgmftype2(fis,'input',2); grid on
xlabel('D4');
figure
set(gcf,'Color',[1,1,1])
plotgmftype2(fis,'output',1); grid on
xlabel('E');

end
```

A.3. Function to obtain the Sobel operator

```
function [Gx,Gy]=sobeloperator(image,noise)

I=double(image);

%% add Gaussian noise to the image
I=awgn(I,dbi,'measured');

%% Obtain the row (r) and column(c) size
[r,c]=size(I);

%% Appy Sobel operator to obtain the horizontal and vertical gradients
for r=2:r-1
    for c=2:c-1
        Gx(r,c)= -1*I(r-1,c-1)-2*I(r-1,c)-I(r-1,c+1)...
                 + I(r+1,c-1)+2*I(r+1,c)+I(r+1,c+1);
        Gy(r,c)= -1*I(r-1,c-1)+I(r-1,c+1)-2*I(r,c-1)...
                 +2*I(r,c+1)-I(r+1,c-1)+I(r+1,c+1);
    end
end

end
```

A.4. Function to define the FIS for the Sobel + GT2 FSs

```
function fis=sobelg2fis(clow,cmedium,chigh,cfondo,cborde,fpdelta,fpedges)

NumMf = [3 3 2];
```

```
MfType = str2mat('gaussmgausstype2','gaussmgausstype2','gaussmgausstype2');

fd=fpdelta/2;
fe=fpedges/2;
sd=chigh./4;

sefondo=abs(cfondo-cborde)/2;
seborde=abs(cfondo-cborde)/2;
rho=0.6;

MfParams{1}=[sd(1) clow(1)-fd(1) clow(1)+fd(1) rho; sd(1) cmedium(1)-fd(1)
cmedium(1)+fd(1) rho; sd(1) chigh(1)-fd(1) chigh(1)+fd(1) rho];

MfParams{2}=[sd(2) clow(2)-fd(2) clow(2)+fd(2) rho; sd(2) cmedium(2)-fd(2)
cmedium(2)+fd(2) rho; sd(2) chigh(2)-fd(2) chigh(2)+fd(2) rho];

MfParams{3}=[sefondo cfondo-fe cfondo+fe  rho;seborde cborde-fe cborde+fe
rho];

fis =
newgfistype2('edges','mamdani','singleton','min','max','min','max','centroid'
);

%%
fis = addgvartype2(fis,'input','DH',[min(clow) max(chigh)]);
fis = addgvartype2(fis,'input','DV',[min(clow) max(chigh)]);
fis = addgvartype2(fis,'output','E',[min([cfondo cborde]) max([cfondo
cborde])]);

%%
fis = addgmftype2(fis,'input',1,'LOW',MfType(1,:),MfParams{1}(1,:));
fis = addgmftype2(fis,'input',1,'MIDDLE',MfType(1,:),MfParams{1}(2,:));
fis = addgmftype2(fis,'input',1,'HIGH',MfType(1,:),MfParams{1}(3,:));

%%
fis = addgmftype2(fis,'input',2,'LOW',MfType(2,:),MfParams{2}(1,:));
fis = addgmftype2(fis,'input',2,'MIDDLE',MfType(2,:),MfParams{2}(2,:));
fis = addgmftype2(fis,'input',2,'HIGH',MfType(2,:),MfParams{2}(3,:));

%%
fis = addgmftype2(fis,'output',1,'BACKGROUND',MfType(3,:),MfParams{3}(1,:));
fis = addgmftype2(fis,'output',1,'EDGE',MfType(3,:),MfParams{3}(2,:));


%% Define the matrix index rule
ruleList= [ 3 3, 2 1 2;
            2 2, 2 1 2;
            1 1, 1 1 1];

fis=addgruletype2(fis,ruleList);

%% Show the fuzzy rules
showrule(fis)

%% Plot the input and output GT2 MFs
figure
set(gcf,'Color',[1,1,1])
```

```
plotgmftype2(fis,'input',1); grid on
xlabel('DH');
figure
set(gcf,'Color',[1,1,1])
plotgmftype2(fis,'input',2); grid on
xlabel('DV');
figure
set(gcf,'Color',[1,1,1])
plotgmftype2(fis,'output',1); grid on
xlabel('EDGES');

end
```

A.5. Function to obtain edge detection method on GT2 fuzzy images

```
function [d1,d2,d3,d4,fgt2]= fuzzygt2mg(image)
[r,c]=size(image);
imageF=double(image);
u = linspace(0,1,100);
for x=2:r-1
    for y=2:c-1

        %% Fuzzifier the image with a GT2 MF
        mf1 = trigausstype2(imageF(x-1,y-1),u,[0 255 255 50 305 305 0.1]);
        mf2 = trigausstype2(imageF(x-1,y),u,[0 255 255 50 305 305 0.1]);
        mf3 = trigausstype2(imageF(x-1,y+1),u,[0 255 255 50 305 305 0.1]);
        mf4 = trigausstype2(imageF(x,y-1),u,[0 255 255 50 305 305 0.1]);
        mf5 = trigausstype2(imageF(x,y),u,[0 255 255 50 305 305 0.1]);
        mf6 = trigausstype2(imageF(x,y+1),u,[0 255 255 50 305 305 0.1]);
        mf7 = trigausstype2(imageF(x+1,y-1),u,[0 255 255 50 305 305 0.1]);
        mf8 = trigausstype2(imageF(x+1,y),u,[0 255 255 50 305 305 0.1]);
        mf9 = trigausstype2(imageF(x+1,y+1),u,[0 255 255 50 305 305 0.1]);

        %% Obtain the secondary MF
        z1 = mf1.fXU{1};
        z2 = mf2.fXU{1};
        z3 = mf3.fXU{1};
        z4 = mf4.fXU{1};
        z5 = mf5.fXU{1};
        z6 = mf6.fXU{1};
        z7 = mf7.fXU{1};
        z8 = mf8.fXU{1};
        z9 = mf9.fXU{1};
        fgt2{x,y}= z5;

        %% Calculate the fuzzy Euclidean distance
        d1(x,y)=sqrt(sum((z5(:,2)-z2(:,2)).^2)+sum((z5(:,2)-z8(:,2)).^2));
        d2(x,y)=sqrt(sum((z5(:,2)-z4(:,2)).^2)+sum((z5(:,2)-z6(:,2)).^2));
        d3(x,y)=sqrt(sum((z5(:,2)-z1(:,2)).^2)+sum((z5(:,2)-z9(:,2)).^2));
        d4(x,y)=sqrt(sum((z5(:,2)-z3(:,2)).^2)+sum((z5(:,2)-z7(:,2)).^2));

    end
end
end
```