Multi-scale edge finding in noisy pictures
Antonio V. Buemi\textsuperscript{1}, Giovanni Gallo\textsuperscript{2}, Filippo Stanco\textsuperscript{3}
Università di Catania, Dipartimento di Matematica e Informatica

Abstract
Finding edges in digital images is a very basic and important task. This paper describes a new technique to address this problem. The proposed technique tries to solve two of the main difficulties that edge finding algorithms must face: uncertainty due to noise artifacts and combination of details detected at different scales. In particular we adapt a fuzzy arithmetic approach called “Fuzzy Marching Squares” (G.Gallo, S.Spinello, Thresholding and fast iso-contour extraction with fuzzy arithmetic, Pattern Recognition Letters 21 (2000) 31-44) to obtain the edge map of a digital picture. Our map assembles together information about uncertainty and scale. Comparisons with results obtained with Canny’s algorithm show that our approach is particularly suitable in presence of additive noise. As for complexity the proposed approach is comparable with the application of traditional median filter.

Keywords: Digital image processing, Edge finding, Fuzzy arithmetic, Multi scale analysis.

1 Introduction
Edge finding is a basic building block of any computer vision algorithm. For this reason it has attracted an enormous amount of research. Important results have been achieved and good algorithms to routinely perform edge finding tasks are well known and widespread (see for example [1]). At the same time some particular issues that arise when dealing with pictures that are characterized by strong noise and presence of details at a large variety of scales are still largely unsolved. A common approach to alleviate the effects of noise is to smooth out the image, but this produces, in turn, loss of details. At the same time linear filters with kernel of different sizes may be adopted to detect edges at different scales, but there is no obvious way to combine together the responses obtained by such filters.

In this paper we apply a new technique, introduced in [2], to obtain an algorithm for edge finding that is, at the same time, robust to noise and integrates in a somehow natural way multi scale information. The technique in [2] builds a summarized representation of the picture under examination, using ideas from statistics and fuzzy arithmetic and propose an efficient way to interrogate such summary.

In this paper we propose to apply the above methodology in order to locate regions of a digital picture where the absolute value of the gradient exceeds a given threshold with a prescribed presumption level. The technique moreover may be used to carry out the detection task at different scales. It is hence possible to obtain several maps of the edges in a digital picture relatively to different presumption levels and to different scale. We propose simple morphological operators to combine together the relevant information in these maps in order to obtain a unique, visually expressive, edge picture. Comparison of some of the results obtained shows that the novel approach may provide more reliable information.

The rest of this paper is organized as follows: Section 2 introduces the basic ideas about fuzzy arithmetic and its use in digital image threshold; Section 3 describes the step of our algorithms; Section 4 presents our implementation and our experimental results. A final Conclusion Section ends the paper.

2 Fuzzy arithmetic and digital pictures
In this Section we give a short self-contained presentation of the fuzzy arithmetic ideas needed to understand the proposed algorithm. For notations and definition not explicitly mentioned here the reader is referred to [3].

2.1 Fuzzy arithmetics
A fuzzy real number $F$ is an interval $[a, b]$ of the real line together with a “membership function”, $m(t)$ from the set of the real numbers to the unit interval $[0,1]$ such that:

- $m(t) = 0$ for $t$ in $\mathbb{R} \setminus [a, b]$ ;
• There is at least a point \( c \) in \([a, b]\) such that \( m(c) = 1 \).

\( m(t) \) can be a general function, however, for our application only triangular fuzzy numbers are used. A triangular fuzzy number \( F = ([a, b], m(t)) \) is a fuzzy number such that there is only one point \( c \) in \([a, b]\) such that \( m(c) = 1 \) and the function \( m(t) \) is linear and monotonically increasing from \( a \) to \( c \) and linear and monotonically decreasing from \( c \) to \( b \). A triangular fuzzy number is hence completely known from the 3-tuple \((a, b, c)\).

Given a real number \( s \) in \([0,1]\) the interval \( F_s \) is the subinterval \([l_s, h_s]\) of \([a, b]\) such that for every \( t \) in \([l_s, h_s]\), \( m(t) \) is greater or equal to \( s \). \( F_s \) is said “s-cut” of the number \( F \). This interval is particularly simple to compute for the case of fuzzy triangular numbers.

Although it is possible to introduce a large variety of arithmetic operators over fuzzy numbers, here we use just two operations: linear interpolation and windowing.

Given two fuzzy triangular numbers described by the 3-tuple \((a, b, c)\) and \((d, e, f)\), the fuzzy linear segment joining the two numbers is given as the set of fuzzy numbers \((a+t(d-a), b+t(e-b), c+t(f-c))\) obtained as \( t \) ranges in \([0, 1]\). A graphical representation of a linearly interpolated segment between two fuzzy triangular numbers is reported in Fig.1.

![Fig. 1. The linear interpolation between fuzzy triangular number (0,6,3) and (6,9,7.5). Different presumption levels are shown at different gray levels.](image)

Given a fuzzy triangular number \( F=(a, b, c) \) and a pair of real numbers \( T_1<T_2 \), we say that \( F \) is in the window \([T_1, T_2]\) with presumption level \( s \) if the intersection of interval \( F_s \) and interval \([T_1, T_2]\) is not empty. Fig.2 shows pictorially the result of windowing a fuzzy segment.

![Fig.2. Pictorial results of windowing a fuzzy segment linearly interpolating two triangular fuzzy numbers.](image)

2.2 Fuzzification of digital pictures

A preliminary operation to process digital picture using fuzzy technique is to build a “fuzzy” representation of it. Given a digital image \( I \), made of \( M \times N \) pixels, each of intensity level in the range \([0, 255]\), a fuzzy representation of it is a an array of \( M/h \times N/h \) fuzzy numbers obtained as follows:

The original image \( I \) is partitioned into \( M/h \times N/h \) squares of \( 2h \times 2h \) pixels, partially overlapping. For each of this square the following three simple statistics are computed:

\[ A = \text{first quartile gray value of pixels in the square}; \]
\[ B = \text{median gray value of pixels in the square}; \]
\[ C = \text{third quartile gray value of pixels in the square}; \]

The image fuzzification algorithm may hence be summarized as follows:

**ALGORITHM: Picture Fuzzification**

- **INPUT:** A gray level digital picture of size \( M \times N \), a lattice step \( h \).
- **OUTPUT:** An array of size \( M/h \times N/h \) of triangular fuzzy numbers.

**STEP 1:** Divide the input picture in \( M/h \times N/h \) overlapping squares of size \( 2h \times 2h \);

**STEP 2:**

FOR every square produced in Step 1, COMPUTE

\[ A = \text{First quartile gray value of pixels in the square}; \]
\[ B = \text{third quartile gray value of pixels in the square}; \]
\[ C = \text{Median gray value of pixels in the square}; \]

SET the corresponding entry in the output array to the fuzzy number \((A,B,C)\);
2.3 Querying a fuzzy picture

The fuzzy image obtained as in the previous subsection can be now efficiently interrogated relatively to a window of intensity values \([T_1, T_2]\) using a “Marching cubes”-like fashion ([4]). More precisely the fuzzy values at the vertexes of the discrete lattice are linearly interpolated along the arcs of the lattice and these fuzzy segments are windowed according with the procedure described in subsection 2.1.

Finally for each square the segments that have been found in the windowing process are joined together with the standard computation of their fuzzy hull. This final set of pixels is the answer to the windowing query.

This operation can be performed very efficiently and it is pictorially described in Fig.3 to 5. More details on this procedure may be found in [2] where the reader may find a through analysis about limits and usefulness of this approach. In summary the querying is described as follows:

**Algorithm: Querying a gray interval over a lattice square.**

- **Input:**
  - four vertexes UL, UR, LL, LR of the lattice of the fuzzy number summarizing the picture;
  - an interval \([a,b]\) of gray values;
  - a finite set \(\{\alpha_1 \ldots \alpha_n\}\) of presumption values;
- **Output:**
  - a fuzzy set where the picture may possibly assume values in the interval \([a,b]\) with presumption levels \(\{\alpha_1 \ldots \alpha_n\}\).

**FOR** every presumption level in \(\{\alpha_1 \ldots \alpha_n\}\).

- **Compute** the intersection of \([a,b]\) with the four linear function interpolating the fuzzy values in UL, UR, LL, LR along the edges UL-UR, UL-LR, LL-LR, UL-LL;
- **Set** the convex hull of the four (possibly empty) intersections found above as the level set at the corresponding presumption level of the fuzzy set in output;

**End Algorithm**

What is important to remark here is that the suggested procedure combines in a unique step a regularization procedure (smoothing through statistical operations), a scale selection (through the adoption of a suitable parameter \(h\)) and a possibilistic analysis of picture details (through the use of fuzzy numbers). The answer to the query gives the set of pixel that with several degree of presumption my fall within the prescribed window, but this information is relative to the specific scale considered. Frequently, especially in medical images, a detail cannot be correctly and completely detected at a single scale. It is hence crucial to have a meaningful synthesis of the information gathered at different scales. The rest of this paper shows that it is possible to combine the results obtained at consecutive scales (i.e., using successive values for the parameter \(h\)) using simple morphological operations.

![Fig.3. Four fuzzy numbers at the vertexes of the lattice representing the fuzzified picture.](image)

![Fig.4. Threshold of the four linear interpolation along the arc of the lattice.](image)

![Fig.5. Convex hull of the results of the threshold operation in Fig.4.](image)

3 Edge finding using fuzzy arithmetic

Edges in a digital picture are the boundaries between homogeneous regions. This concept is not univocally defined: it depends on the vague concept of “homogeneous region”. For example a homogenous
collection of pixels may mean “pixels coming from the same physical object”, or may mean “pixels with the same intensity value”, or “pixels whose statistics, as a population, obey to a given law”. In this paper we are concerned with boundaries between regions whose pixels have intensities within a small range. Starting from this assumption a high gradient value is a strong cue to the existence and location of an edge. The most widely known and used algorithm to find edges, in this case, is an algorithm proposed by Canny in 1986 [1]. The algorithm proposes a sophisticated analysis of pixels with high gradient values. In presence of moderate to strong noise, however, gradient information becomes unreliable and a lot of spurious “edges”, not corresponding to any real boundary, arise.

The technique proposed in this paper starts from the same basic idea: regions where, with different level of presumption, has been estimated a high gradient value are considered edges at different level of presumption. Since the estimation is done using statistics that are robust to noise our technique is able to detect reliably real boundary, arise.

The successive step of our algorithm requires to classify as edge/not-edge the pixels of the original picture. The whole construction is shown, for a test image, in Fig. 8.

### 3.2 Merging consecutive scales

In this step of the procedure for a fixed presumption level the three edge/not-edge maps described above are merged using a standard morphological intersection. More precisely a pixel is kept as an edge pixel if and only if it is and edge pixel in all three edge maps obtained at consecutive scales. This allows to remove spurious edge pixels and to generate a reasonably smooth edge map that synthesize the information gathered at different scales.

This step may be repeated for all the presumption levels needed. In our experiments we have obtained satisfactory results using the presumption levels $\alpha = 0$ and $\alpha = 0.9$ (see Fig. 9).

A final merging of the edge maps obtained in so far is shown in Fig. 10 and it is straightforward because of the following Lemma, whose simple proof is not reported.

**Lemma.** If $\alpha_t < \alpha_c$ the set of edge pixels obtained at presumption level $\alpha_t$, merging scales $h_t$, $h_2$ and $h_3$ contains the set of edge pixels obtained at presumption level $\alpha_c$, merging scales $h_1$, $h_2$ and $h_3$.

The merging algorithm may hence be summarized as follows:

```plaintext
ALGORITHM: Merging the answers obtained at consecutive h scale values.

- **INPUT:** three fuzzy sets $F_{h_1}$, $F_{h_2}$, $F_{h_3}$ described by their respective level sets at presumption levels $\{\alpha_1, \ldots, \alpha_n\}$;
- **OUTPUT:** A fuzzy set $G$, described by the level sets at presumption levels $\{\alpha_1, \ldots, \alpha_n\}$ merging the edge information in the input sets.

FOR every presumption level $\alpha$ in the set $\{\alpha_1, \ldots, \alpha_n\}$

SET the $\alpha$-level set of $G$ as the intersection of the $\alpha$-level sets of $F_{h_1}$, $F_{h_2}$, $F_{h_3}$

END ALGORITHM
```

### 4 Experimental results

We have implemented the ideas reported above into a user friendly application written in C++ using the GUI libraries provided with the Borland C++ Builder compiler.
We report some examples that provide a qualitative evidence of the good results obtained in so far.

Fig. 11 and Fig.12 shows the difference between the edge map obtained using our algorithm and a simple gradient magnitude threshold on the standard Lena image and on a section of the Visible Human Data Set. Our maps show a better coherence and refinement of the found edges. Moreover the different gray levels used in the maps to mean different presumption levels are very expressive cues to the relevance of the observed edges.

Fig.13 shows how additive Gaussian noise affects our method compared with Canny’s algorithm. Although a loss of precision happens with both approaches the new algorithm proposed in this paper is still able to provide a reasonable answer even in this difficult situation.

In order to get more quantitative results we have measured and compared the average time required to perform fuzzification and querying of a gray level picture at different values of the parameter h for one fixed presumption level. Similarly we have measured the average time required to perform traditional median filter at different values of parameter h and to threshold the result. The averages have been computed over a set of about 100 pictures of several different sizes. The results are reported in Fig.6 and Fig.7. Although the proposed approach is slightly slower for small values of h it becomes faster that the traditional median technique as h increases. This advantage does not hold if fuzzy processing is done using too many presumption levels. This suggests that for real applications of the proposed method a choice of at most three different presumption levels is advisable.

5 Conclusion

We have presented an application of fuzzy arithmetic to edge finding in digital pictures. Qualitative analysis provides already strong evidence that the proposed approach may be very useful in low level processing of noisy pictures.

We have also proposed a simple but effective way to visualize uncertainty in pictures using different gray levels for different presumption levels of a fuzzy set.

Further research will try to validate quantitatively this findings and to extend the present result to the case of 3D-data and to the case of temporal image sequences.

Acknowledgements

We wish to thank Salvatore Spinello for letting us use part of his software and for many helpful suggestions.

References


![Fig.6. Comparison between average time required to obtain a fuzzy version of the image and the time required to query the result.](image6.png)

![Fig.7. Comparison between average total time required by the fuzzy approach and the traditional median approach.](image7.png)
Fig. 8. A section from the Visible Human Data Set (a); the picture has been fuzzy thresholded at presumption level 0, with $h=3$ (b), $h=5$ (c) and $h=7$ (d); the picture has also been fuzzy thresholded at presumption level 0.9, with $h=3$ (e), $h=5$ (f) and $h=7$ (g).
Fig. 9. Morphological intersection of images in figures 7b-d (a); Morphological intersection of images in figures 7e-h (b).

Fig. 10. Final edge map obtained from image in fig. 7a using our fuzzy approach.

Fig. 11. Input image (a); edge map obtained using our algorithm (b); edge map obtained using gradient magnitude threshold (c).
Fig. 12. Input image (a); edge map obtained using our algorithm (b); edge map obtained using Canny’s algorithm with similar parameter values (c).

Fig. 13. Same image in Fig. 11 with 10% gaussian noise added (a); edge map obtained using our algorithm (b); edge map obtained using Canny’s algorithm with similar parameter values (c).