A New Multi-Layered Fuzzy Image Filter
for Removing Impulse Noise

Russel J STONIER

Faculty of Informatics and Communication, Central Queensland University,
Rockhampton, QLD 4702, Australia

and

Mohamed M ANVER

Faculty of Informatics and Communication, Central Queensland University,
Rockhampton, QLD 4702, Australia

ABSTRACT

In this paper we develop a fuzzy image filter which consists of a multi-layered fuzzy structure based on the weighted fuzzy blend filter for the removal of noise from images heavily corrupted by impulse noise, while preserving the intricate details of the image. The introduction of multi-layered fuzzy systems substantially decreases the number of rules to be learnt. We then show how Evolutionary Algorithms (EAs) can be used to effectively learn the fuzzy rules in each knowledge base. Results are presented for impulse noise corruption of the well-known 'Lena' image.

Keywords: Fuzzy filter, image enhancement, multi-layered fuzzy system, evolutionary learning, evolutionary algorithm.

1. INTRODUCTION

Conventional image enhancement techniques such as mean and median filtering have been employed in various applications in the past and are still being used. However techniques using Fuzzy Logic (FL) which mimics human reasoning and tolerates ambiguities well are increasingly being looked into as alternatives to these conventional techniques.

When using FL to any practical application, the question arises as to how the fuzzy rule base should be decided. Usually, human intuition is used to decide these rule bases with the intention of seeing a human-like behaviour from the system. However EAs which belong to the category of artificial intelligence techniques is proving to be very powerful when determining the fuzzy rules in the knowledge base rather than employing human experience and knowledge. This is specifically true in automated processes where the human intervention is a minimum.

In this paper, both FL and EAs are employed to show how they could be used in a practical digital image processing system to remove heavy impulse noise from corrupted images. It is shown how a Multi-Layered Fuzzy Logic (MLFL) structure can be developed as a fuzzy image filter to reduce the high computational power required otherwise.

Section 2 of the paper gives an introduction to MLFL systems. In Section 3 we present a simple analysis of the Weighted Fuzzy Blend Filter presented in [1]. It will be used to construct a two layered fuzzy image enhancement algorithm in Section 4, whose fuzzy rules in each layer will be learnt from corrupted data by an evolutionary algorithm. Application will be made to well-known 'Lena' image and comparisons are made where applicable.

2. MULTI-LAYERED FUZZY LOGIC SYSTEMS

A key consideration when designing FL systems is the size of the rule base, which increases exponentially with the number of inputs - the so called 'curse of dimensionality'. For example consider the 8 input - single output single-layer FL system shown in Figure 1 and the MLFL structure shown in Figure 2. In the MLFL structure the output of the first layer is combined with the output of the second layer to obtain the final output.
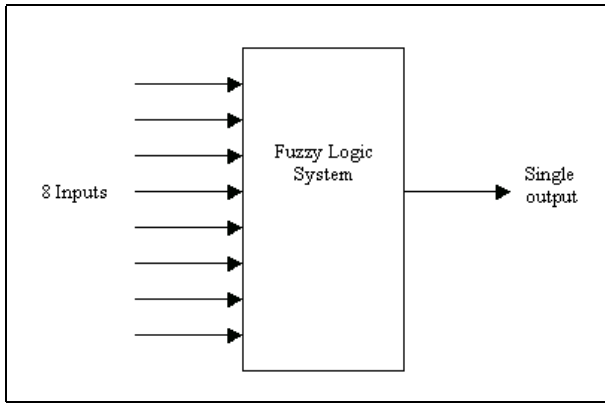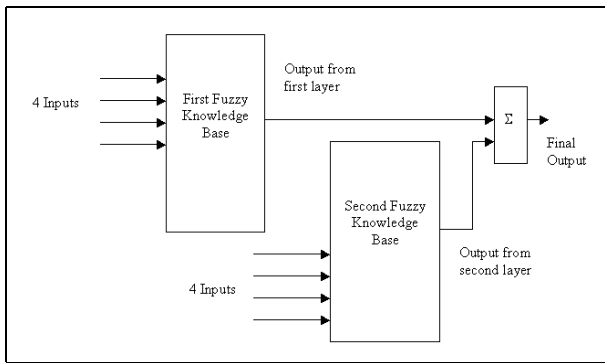
Figure 1: Single-Layer Fuzzy Logic System

Figure 2: A Multi-Layered Fuzzy Logic System

The Fuzzy Knowledge Base (FKB) for the single-layer fuzzy system consists of $2^8 = 256$ rules assuming each input is represented using a membership function having two fuzzy sets. For the MLFL system shown in Figure 2, the first layer will have $2^4 = 16$ rules and the second layer will have another $2^4 = 16$ rules, making the total number of rules to 32. The total number of rules in the FKB has been reduced by a factor of 8 in this example, by employing a multi-layered approach. A question that arises when designing a multi-layered structure is how do we break down the inputs among different layers. There are two possible ways to solve this problem according to recent papers published in the area of MLFL systems. One method is to break up the inputs so that a physically meaningful output is produced from the first layer and then use it together with the second layer to obtain the final output. This kind of approach has been proven to be successful in [3] where a MLFL controller has been developed for the control of a robot. Here the output of the first layer is a control variable giving the direction of a robot from an initial position. Then it is input to the second layer to control the speed of the moving robot. But obtaining a meaningful output variable from the first layer is not essential to design

a successful MLFL system, see [4], where the balancing of an inverted pendulum is tackled using a multi-layered fuzzy approach. How to structure the inputs to a MLFL system among different layers is an open research question.

Different applications that show how multi-layered fuzzy systems and EAs can be used to find fuzzy rule bases are described in [2], [5], [6], [7] and [8].

3. THE WEIGHTED FUZZY BLEND FILTER

We begin with a discussion of the Weighted Fuzzy Blend Filter (WFBF) presented in [1]. From this filter our multi-layered fuzzy filter will be developed.

The WFBF consists of two components to recover original pixel values from a corrupted image. The first component is aimed at detecting noisy pixels by comparing the intensity differences between a selected pixel and the neighbouring pixels in a sliding window of 3×3. The second component concentrates on the pixel data re-construction when necessary. Both components use fuzzy reasoning to detect if the pixel is noisy and to reconstruct the pixel by using mean or median filters. A 3×3 sliding window is shown below.

$$\begin{bmatrix} I_1 & I_2 & I_3 \\ I_4 & I_0 & I_5 \\ I_6 & I_7 & I_8 \end{bmatrix}$$

Let us suppose the pixel to be processed is I_0 and pixels I_1 to I_4 have been processed. The input variables to the first FKB are the $n = 4$ intensity differences given by,

$$x_j = abs(Ij^{'} - I_0) \quad j = 1, \cdots, 4 \qquad (1)$$

Since the difference values for pixels I_5 to I_8 are uncertain as they may be highly corrupted by noise, they are not used as fuzzy inputs. Two input fuzzy sets, named DH = Difference High and DL = Difference Low, are defined with membership functions in terms of constants $c_1 = 10$ and $c_2 = 40$, given as follows:

$$U_{DH}(x) = \begin{cases} 0 & x \leq c_1, \\ (x - c_1)/(c_2 - c_1) & c_1 < x < c_2, \\ 1 & x \geq c_2. \end{cases}$$

$$U_{DL}(x) = \begin{cases} 1 & x \leq c_1, \\ (c_2 - x)/(c_2 - c_1) & c_1 < x < c_2, \\ 0 & x \geq c_2. \end{cases}$$

The input variables lie in the interval $[0, 255]$ for the processing of grey scale images. The output variable of

the fuzzy inference engine $out \in [0,1]$, is a member of two fuzzy sets, $B^1 = VL$ and $B^2 = VH$ respectively. The centres of these sets are taken to be $y^1 = 0$ and $y^2 = 1$. There are $M = 16$ rules in the fuzzy rule base. The rules are built using intuition of how the intensity differences determine the existence of a noisy pixel. For example rule ℓ may be of the form:

$$\text{If } (x_1 \text{ is } A_1^\ell) \text{ and } (x_2 \text{ is } A_2^\ell) \text{ and } (x_3 \text{ is } A_3^\ell)$$
$$\text{and } (x_4 \text{ is } A_4^\ell) \text{ then } (out \text{ is } B^\ell)$$

where on the antecedent side of the rule $A_1^\ell = DH$, $A_2^\ell = DH$, $A_3^\ell = DH$ and $A_4^\ell = DH$, and in the consequent $B^\ell = VL$. Each rule has been designed to deal with a particular pattern of intensity difference among the neighbouring pixels.

Given a fuzzy rule base with M rules, the output out as given in Equation 2 uses a singleton fuzzifier, Mamdani product inference engine and centre average defuzzifier, see [10].

$$out = \frac{\sum_{\ell=1}^{M} \overline{y}^\ell (\prod_{i=1}^{n} \mu_{A_i^\ell}(x_i))}{\sum_{\ell=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^\ell}(x_i))} \qquad (2)$$

where $\overline{y}^\ell$ are centres of the output sets B^ℓ.

The enhanced pixel output is given by,

$$I_{enhanced} = out \times I_0 + (1 - out) \times I_{recons} \qquad (3)$$

where $I_{enhanced}$ is the output of the pixel to be estimated, out is the fuzzy output, I_0 is the input value of the pixel and I_{recons} is the reconstructed value for that pixel.

Two recovery methods are used to obtain I_{recons}, one is the median value when intensity gradient is high, and the other is the mean value when the intensity gradient is low. The intensity gradients are defined in 4 directions:

$$\begin{aligned} G(1) &= abs(I_4' - I_5), \\ G(2) &= abs(I_2' - I_7), \\ G(3) &= abs(I_3' - I_6), \\ G(4) &= abs(I_1' - I_8) \end{aligned}$$

and the minimum gradient is defined as,

$$G_{min} = \min\{G(1), G(2), G(3), G(4)\} \qquad (4)$$

The median value I_{median} is calculated using a 3×3 sliding window, while the mean value I_{mean} is calculated using the 2 outer pixels corresponding to the minimum gradient direction given by G_{min}. For example, if the minimum gradient is in the horizontal direction the mean will be calculated as $(I_4 + I_5)/2$.

Another fuzzy membership function GL (Low Gradient), with $c_1 = 40$ and $c_2 = 80$, is introduced for determining which method should be used,

$$U_{GL}(x) = \begin{cases} 1 & x \leq c_1, \\ (c_2 - x)/(c_2 - c_1) & c_1 < x < c_2, \\ 0 & x \geq c_2. \end{cases}$$

Fundamentally this second level in the WFBF incorporates, components of the traditional mean and median filters. We take weights for both mean and median to be simply $U_{GL}(x)$ and $U_{\overline{GL}} = 1 - U_{GL}(x) = U_{GH}$, respectively. Then the reconstructed value is obtained from,

$$I_{recons} = U_{GL}(x) \times I_{mean} + U_{\overline{GL}} \times I_{median} \qquad (5)$$

This filter has been shown to be capable of removing noise from images corrupted by impulse noise up to 30%, see details in [1].

The fuzzy rule base in the WFBF filter was created using human reasoning. A simple iterative search can be undertaken on all possible ($2^{16} = 65,536$) fuzzy rule bases to find the best rule base. The Mean Square Error (MSE) between the uncorrupted image and the enhanced version of the corrupted image was used as the criteria to determine the quality of the image. A 256×256 image corrupted by 30% impulse noise was used and the minimum MSE obtained was 90.12. This MSE was lower than that obtained by the WFBF which shows that it is possible to find a 'better' WFBF by a direct search of all the possible knowledge bases.

In this paper we propose to develop a second layer which has as output the reconstructed pixel value I_{recons} which can be immediately input into Equation (3).

4. MULTI-LAYERED FUZZY IMAGE FILTER

Multi-Layered Fuzzy System Design

A multi-layered fuzzy logic system is now designed to remove impulse noise in similar manner to the WFBF. This particular structure, shown in Figure 3, has two FKBs. The first layer has 4 input intensity differences given by Equation 1 and an output variable $out \in \{0, 1\}$ representing whether a selected pixel is noisy or not as described in Section 3. The fuzzy membership function for the output variable out is as given in Section 3. The number of rules in the first FKB is $2^4 = 16$. The second FKB has 3 inputs; they are:

(i) Minimum gradient (as calculated in the previous section) fuzzified using 2 given fuzzy sets.

(ii) Median of the pixels in the 3×3 sliding window fuzzified using 11 triangular fuzzy sets shown in Figure 4.

(iii) Mean in the direction of the minimum gradient (as calculated in WFBF) fuzzified using the 11 triangular fuzzy sets.
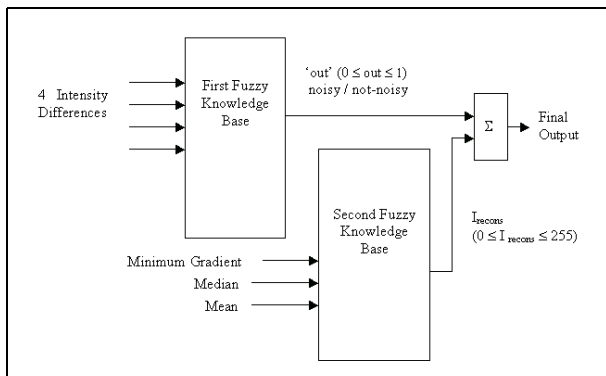

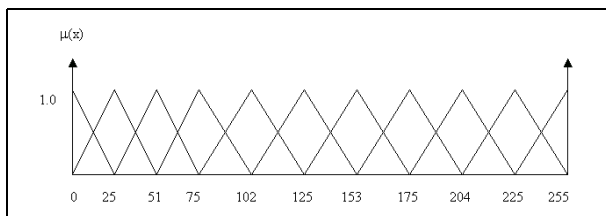Figure 3: Multi-Layered Fuzzy Image Filter

Figure 4: Fuzzy sets for fuzzification of median and mean values

Each fuzzy set in Figure 4 can be described by the following equation where x is the value of the variable to be fuzzified, L is the 'left corner' of a fuzzy set, C is the 'center' of a fuzzy set and R is the 'right corner' of a fuzzy set.

$$\mu(x) = \begin{cases} (x-L)/(C-L) & L < x \leq C \\ (R-x)/(R-C) & C < x < R \\ 0 & else \end{cases}$$

The output of the second layer is taken as the 're-constructed' integer value of the pixel ($I_{recons} \in \{0, \cdots, 255\}$). The number of rules in the second FKB is $2^1 \times 11^2 = 242$. The final output of the enhanced pixel is calculated according to Equation 3.

Evolutionary Learning of the Fuzzy Rule Bases
We show now how to apply an EA [11, 12], to learn the fuzzy rules in the two knowledge bases. Each individual string in the evolutionary population is to uniquely represent the multi-layered structure. This is achieved as follows. In the first knowledge base each

fuzzy rule is uniquely defined by the consequent part, it being represented by either a 0 or 1 depending on whether the output variable *out* belongs to the fuzzy set VL or VH respectively. In similar manner each fuzzy rule in the second knowledge base is defined by its consequent.

The two fuzzy rule bases can therefore be represented as a string of $M = 258$ consequents,

$$\underset{\sim}{x}_k = \{a_1, \cdots, a_{16}, a_{17}, \cdots, a_{258}\},$$

where $a_j \in \{0, 1\}$ for $j = 1, \cdots, 16$, are the consequents for the rules in the first knowledge base and $a_j \in \{0, 255\}$ for $j = 17, \cdots, M$, are the consequents for the rules in the second knowledge base. In this representation the output centres for sets VL and VH in the first layer are represented by 0 and 1, and in the second layer we take as output centres, the full set of integers in the range from 0 to 255.

Each such string forms an individual in the evolutionary population and a possible solution to finding the "best" fuzzy filter. The population at generation t, $P(t) = \{\underset{\sim}{x}_k \; : \; k = 1, \cdots, N\}$, where $N = 100$ is the number of individuals in the population, the population size. We define the fitness for each individual f_k, as the MSE between the uncorrupted image and the enhanced version of the 256×256 'Lena' image corrupted by 30%. The initial generation was formed by placing random elements in each individual string. Subsequent generations were built using a full replacement policy including an elitism strategy of size 2 and tournament size $n_T = 3$.

Mutation probability was set to $p_M = 0.03$. Normal binary mutation was used for the first 16 elements of a string and the mutation method shown in the following code segment was used for the remaining 242 elements. Here pv is the present value of a string component and mv is the mutated value.

```
if (flip(p_{M}) == true)
{ if (pv  ==  0)
    {mv = pv + rand2limit(1,128);}
  else f (pv == 255)
    {mv = pv - rand2limit(1,128);}
  else
    {
    if (flip(0.5) == true)
      {mv = pv + rand2limit(1,128);}
    else
      {mv = pv - rand2limit(1,128);}
  }
} else
    {mv = pv;}
if (mv <  0 ) { mv =  0 ; }
if (mv > 255) { mv = 255; }
```

Remark: This algorithm says that if the present

value of an element is 0 then a number between 1 and 128 should be added to the present value. If the present value of an element is 255 then a number between 1 and 128 is subtracted from the present value. Else a value between 0 and 128 is either added or subtracted with a probability of 0.5. If the calculated value lies outside $[0, 255]$ then the value is clipped to the boundary limits.

The crossover strategy consisted of a blend of simple one-point crossover and a modified version of arithmetic crossover with a probability of $p_C = 0.6$. Arithmetic crossover with different parameter (α) values was used as shown by the following code:

```
double alpha1 = rand0_1();
double alpha2 = rand0_1();
for (i = 0; i < length_of_chromosome; i++) {
    c1[i] = round(alpha1 * p2[i]
            + (1 - alpha1) * p1[i]);
    c2[i] = round(alpha2 * p2[i]
            + (1 - alpha2) * p2[i]); }
```

The EA was set to terminate either after a pre-set number of generations or when the difference between minimum fitness value and the maximum fitness value was smaller than a pre-determined value. It was found that for this
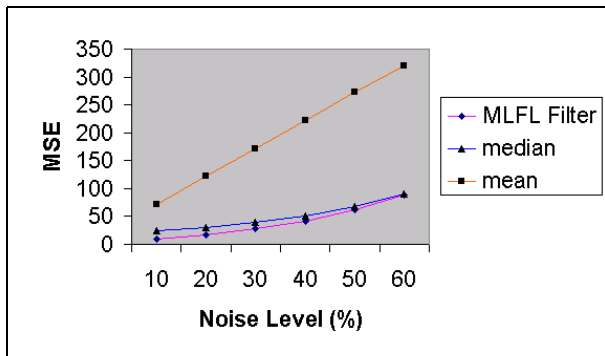

Figure 5: Graph showing the MSE vs Noise Percentage

multi-layered structure the EA converged around the 280th generation to a fuzzy rule base which gave a minimum MSE value of 67.16. The algorithm was re-run with different random initializing seeds achieving minimum MSE around the same value.

Experimental Results
This filter was implemented on a 'Lena' test image having 30% corruption and size 256×256 pixels to reduce the computational time required and then tested on a 512×512 'Lena' image. The results for the 512×512 image are shown in Figures 6 and 7 for corruption levels of 30% and 40% and the MSE values obtained for corruption levels ranging from 10% to 60% are shown in Figure 5.

It is observed that this new MLFL system has sufficient knowledge learnt from enhancing a 256×256 image corrupted by 30% impulse noise, to remove noise as high as 40% from the 512×512 image.

5. CONCLUSION

In this paper we have presented a multi-layered fuzzy image filter based on the weighted fuzzy blend filter. The rules in the layers have been learnt directly from data using an appropriately defined evolutionary algorithm with modifications to the basic mutation and crossover operators. It has been shown that this new fuzzy image filter performs better than conventional mean and median filters. Indeed the MLFL system having been learnt by enhancing a reduced 256×256 'Lena' image corrupted by 30% impulse noise, is able to remove noise as high as 40% in a 512×512 'Lena' image.

The filter is seen to preserve intricate features of the image while removing heavy impulse noise where as the conventional mean and median filters fail in this context even at low corruption levels. That is the granularity is well maintained in the enhanced images obtained through the fuzzy processing which is not reflected through the plots of MSE values for median filter and the MLFL structure as shown in Figure 5.

The learning of fuzzy rules in a fuzzy image filter with a true hierarchical fuzzy logic structure where the output of the first layer is fed in to the second layer to obtain an 'improved' final output [13], is being currently studied.

6. REFERENCES

[1]   X. Mao, R.J. Stonier and P. Thomas, "An Effective Fuzzy Image Filter", Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA), M. Mohammadian (Ed), Las Vegas, July 2001, pp. 280-285.

[2]   R. J. Stonier and M. M. Anver, "A Multi-Layered Fuzzy Image Filter for Removing Impulse Noise", accepted for presentation at and publication in the Proceedings of the International Conference on Intelligent Systems (ICIS) 2002, Boston, July, 2002.

[3]   Masoud Mohammadian and R.J. Stonier, "Adaptive Two Layer Fuzzy Logic Control of a Mobile Robot System", Proceedings of IEEE Conference on Evolutionary Computing, Vol.1, Perth WA Australia, December 1995, pp. 204-209.

[4]   R.J. Stonier, A.J. Stacey and C.Messom, "Learning fuzzy controls for the inverted pendulum", Proceedings of ISCA 7th International Conference on Intelligent Systems, Melun, Paris, July 1998, pp. 64-67.

Figure 6: (a) Corrupted image (30%) (b) Restored image (30%) - MLFL Structure

Figure 7: (a) Corrupted image (40%) (b) Restored image (40%) - MLFL Structure

[5] M.Mohammadian and R.J. Stonier, "Hierarchical Fuzzy Control", Proceedings of the 7th International conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, Paris, 1998, pp. 621-629.

[6] Masoud Mohammadian, Mark Kingham and R. J. Stonier, "Prediction of Interest Rate using Neural Networks and Fuzzy Logic", Proceedings of ISCA 7th International Conference on Intelligent Systems, Melun, Paris, 1998, pp. 72-75.

[7] P.J. Thomas and R.J. Stonier, "Evolutionary learning of fuzzy control in robot-soccer", Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA), M. Mohammadian (Ed), Las Vegas, July, 2001, pp. 494-503.

[8] I. Nainar, M. Mohammadian, R.J. Stonier and J. Millar, "An adaptive fuzzy logic controller for control of traffic signals", Proceedings of the 4th International Conference on Control, Automation, Robotics and Computer Vision (ICARCV'96), Singapore, December 3-6th, 1996, pp. 578-582.

[9] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Massachusetts: Addison Wesley, 1992.

[10] L. Wang, A course in fuzzy systems and control, USA: Prentice Hall, 1997.

[11] D. Goldberg, Genetic Algorithms in Search, Optimisation and Machine Learning, Massachusetts: Addison Wesley, 1989.

[12] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, 1994.

[13] G.V.S. Raju and J. Zhou, Adaptive Hierarchical Fuzzy Controller, IEEE Transactions on Systems, Man and Cybernetics, 1993, pp. 973-980.