Oppositional Fuzzy Image Thresholding

Fares S. Al-Qunaieer
Systems Design Engineering
University of Waterloo
200 University Ave. West
Waterloo, ON N2L 3G1, Canada
Email: falqunai@engmail.uwaterloo.ca

Hamid R. Tizhoosh
Systems Design Engineering
University of Waterloo
200 University Ave. West
Waterloo, ON N2L 3G1, Canada
Email: tizhoosh@uwaterloo.ca

Shahryar Rahnamayan
Electrical and Computer Engineering
University of Ontario Institute of
Technology, 2000 Simcoe Street North
Oshawa, ON L1H 7K4, Canada
Email: shahryar.rahnamayan@uoit.ca

***Abstract*—In many image processing applications, image thresholding is considered to be an important task. Opposition-Based Learning (OBL) was recently introduced and used to enhance different computation algorithms. In this paper, a new thresholding algorithm is proposed by utilizing the concept of opposite fuzzy sets. The algorithm is applied on general set of images and compared with the previous opposition-based thresholding algorithm [1] and a commonly used thresholding method, namely the Otsu method. The most reliable results on the test data are achieved using the proposed algorithm.**

I. Introduction

The process of segmenting the region of interest in an image is considered a very important step in many image processing applications. This arises from the fact that the extracted region will be used by the remaining processing algorithms, such as feature extraction and object recognition. If there is an error in the segmentation phase or if the result is not accurate, the error will be propagated to the subsequent phases. Bilevel thresholding could be considered as the simplest form of segmentation because it is concerned with only two classes (e.g. object and background). Intensive research has been conducted to design and implement better thresholding techniques, over 40 of them have been reviewed and compared by Sezgin and Sankur in [2].

Opposition-Based Computing (OBC) [3] has been recently introduced and used to improve soft computing techniques, such as neural networks [4], evolutionary algorithms [5], [6] and reinforcement learning [7], [8]. In this paper, we discuss different ways to incorporate OBC concepts into the design of image thresholding algorithms. A new image thresholding algorithm based on opposite fuzzy sets is proposed. Experimental results using a general set of images are provided and compared with the results of a recently introduced opposition-based image thresholding algorithm [1] and Otsu method [9].

The remaining of the paper is organized as follows: a quick review of previous thresholding algorithms is presented in Section II. Section III provides an overview of Opposition-Based Learning. Then, opposite fuzzy sets are introduced in Section IV. Methods to apply opposition concepts to image processing applications are discussed in Section V. After that, in Section VI, a thresholding algorithm based on opposite fuzzy sets is proposed. In Section VII, the proposed method is tested and compared with two other methods. Finally, Section VIII concludes this paper.

II. Image Thresholding

Researchers have put huge amount of time and efforts to develop efficient image thresholding algorithms. As pioneers, the work of Otsu, Kapur, and Kittler are described first, then some of the recent works are reviewed in this section.

Proposed Otsu method[9] is considered the most popular thresholding algorithm in literature. From the histogram of image, Otsu proposed an evaluation criterion based only on the zeroth and the first order statistics. The aim of the algorithm is to minimize the measure of separability between the classes. Kapur et al. [10] proposed an algorithm based on entropy definition. They defined the entropies corresponding to the distributions of the different classes. Then, in order to achieve the maximum information between the classes (hence, the optimal thresholding), the sum of the defined entropies is maximized. Kittler and Illingworth [11] modeled the classes (e.g. object and background) as mixture of Gaussians obtained from the histogram of the image. The objective of their suggested algorithm is to optimize the classification error rate to obtain the minimum thresholding error.

Various techniques were utilized to design image segmentation algorithms. Tao et al. [12] proposed the use of normalized graph cut. The weights of the graph were computed from the grey-levels of the image. Yang et al. [13] proposed Spatially Weighted Fuzzy C-Means (SWFCM) algorithm. The authors incorporated the spacial information into Fuzzy C-Mean (FCM) clustering algorithm.

Computational intelligence techniques have been used for image segmentation. Kang and Zhang [14] utilized Cellular Neural Network (CNN) associated with histogram analysis to find the best threshold value. Genetic algorithm was used by Ren [15] to estimate the optimal threshold value. Particle Swarm Optimization was utilized for image thresholding by Lin et al. [16]. They considered each pixel as a particle and the optimal threshold as the food source.

III. Opposition-Based Learning

The concept of Opposition-Based Learning (OBL) was recently introduced by Tizhoosh [3]. The basic idea behind OBL is whenever we are searching for a solution, or we are approximating a solution by looking at a guess in a certain direction in the search space, we should always look at the opposite guess, or we should examine the opposite

978-1-4244-8126-2/10/$26.00

direction as well. Many machine intelligence algorithms consider finding the solution of a given problem as function approximation. Thus, if the objective is to search for the solution x, the algorithm makes an estimation $\hat{x}$ which in process of search/learning/optimization should converge to x. Such algorithms can be computationally expensive. Among others, starting points of search can dramatically affect the accuracy of the found solution (e.g. due to local maxima or minima) and the convergence time. In many cases, starting points are chosen randomly, such as weights of a neural network, initial population of evolutionary algorithms, and action policy of reinforcement agents. If the starting point delivers an estimate which is close to the optimal solution, this results a fast convergence. On the other hand, if the initial search starts from a point which is very far from the optimal solution, such as opposite location in worst case, the convergence will take much more time or finding the solution can even become intractable. Looking simultaneously in both current and opposite positions/guesses/estimates may help to solve this problem.

Definition – Let x be a real number defined on the interval $[a, b]$. The *opposite number* $\breve{x}$ is defined as follows [3]

$$\breve{x} = a + b - x. \tag{1}$$

For $a = 0$ and $b = 1$, we have

$$\breve{x} = 1 - x. \tag{2}$$

In the same manner, the opposite number in a multidimensional search space can be defined.

Definition – Let $P(x_1, x_2, ..., x_n)$ be a point in a n-dimensional coordinate system with $x_1, ..., x_n \in \Re$ and $x_i \in [a_i, b_i]$. The opposite point $\breve{P}$ is defined by its coordinates $\breve{x}_1, ..., \breve{x}_n$ where [3]

$$\breve{x}_i = a_i + b_i - x_i \qquad i = 1, ..., n \tag{3}$$

Opposition-Based Learning (OBL) – Let $f(x)$ be the function in focus and $g(.)$ a proper evaluation function. If $x \in [a, b]$ is an initial (random) guess and $\breve{x}$ is its opposite value, then in every iteration we calculate $f(x)$ and $f(\breve{x})$. For a maximization problem, the learning continues with x if $g(f(x)) > g(f(\breve{x}))$, otherwise with $\breve{x}$.

Ventresca and Tizhoosh [4] investigated the use of opposition based computing to improve the performance of backpropagation neural networks. In their approach, they considered the opposite of the transfer function for a subset of neurons. The opposite of transfer function $f(x)$ is defined as $\breve{f}(x) = f(-x)$. The opposite neural network has the same weights as the original ones and at least one neuron with an opposite transfer function. An extension to reinforcement learning based on opposition based computation was proposed by Tizhoosh [7], [8]. The main idea of the algorithm is to consider actions and opposite actions and/or opposite states. This makes the traversal path of the state space shorter, which means faster convergence. Rahnamayan et al. [5], [6] proposed the inclusion of opposition-based computing in evolutionary algorithms. The authors proposed population initialization based on opposition concept. This is performed by initializing a random population $P(n)$ and calculating the opposite population $OP(n)$. The fittest individuals are selected from both $P(n)$ and $OP(n)$. In addition, based on a jumping rate, the opposite of the current population is calculated and the fittest individuals are selected from both populations.

Opposition-Based Differential Evolution (ODE) was applied for the task of finding the best threshold for images [17]. The process of finding the best threshold was considered as an optimization problem, and the objective function,

$$f(T) = \sum_{i=1}^{M} \sum_{J=1}^{N} |I_{ij} - B(T)_{ij}| \tag{4}$$

was employed, where$M$ and N are the dimension of the image I, and $B(T)$ is the corresponding thresholded image generated by applying the threshold T. The goal is to minimize the objective function. In order to do so, ODE with very small population size (=5), which called *micro*-ODE, was used.

Khalvati et al. [18] considered the use of OBC to enhance window memoization, a technique to accelerate window-based methods in image processing via exploiting image redundancies. The case study was gray-scale morphological algorithms that use 3×3 non-flat structuring elements. A lookup technique that uses multi-thresholding was developed to increase the reuse rate. Each time a lookup is performed on a window, the response of the opposite window is calculated as well. This results in reducing the number of calculations.

The concept of opposition has been applied also for initialization, generation jumping and determining the best member of Particle Swarm Optimization (PSO) [19]. Five methods for employing opposition concept were considered to extend the construction phase of Ant Colony Optimization [20]. Furthermore, opposite concept was used to determine the reliability of Fuzzy c-mean Clustering algorithm [21].

IV. Opposite Fuzzy Sets

Fuzzy sets have been frequently used for constructing new thresholding algorithms, such as in [22]. The notion of opposition has always been a part of fuzzy sets since they were introduced. It is a misconception, however, to consider *negation* as an equivalent to opposition. The negation of a set A is $\neg A$ which means "*not* A", this is too general to catch the oppositeness of A. For example, "not very young" could be any value between "young" and "very old", while its opposite is "very old". Very limited studies of opposition in fuzzy sets have been conducted in literature. These studies mainly originate from pure linguistic perspective by using *antonyms* [23], [24]. Antonyms logic was formalized by Golota [25]. Trillas et al. [26] applied antonyms on fuzzy sets in details. Still, antonyms can be considered only as a special case of opposition, namely linear opposition. Nonlinear opposition can not be modeled by antonyms. Tizhoosh defined a preliminary formal framework for opposite fuzzy sets [1]. Linear fuzzy opposition was defined as "type I opposite fuzzy sets" and nonlinear as "type II opposite fuzzy sets".

Let X be the universe of discourse and $x \in X$ the elements of the objects to be classified.

Definition (Fuzzy Set) – A fuzzy set $A \subset X$ with membership function $\mu_A(x)$ is defined as

$$A = \{(x, \mu_A(x)) \mid x \in X, \mu_A(x) \in [0,1]\} \tag{5}$$

The membership function is given as $\mu_A(x) = f(x; a, \delta)$ where $\mu_A(a) = 1 \quad \forall a_i \in a$ and δ is the somatic parameter that changes the shape of the membership function.

Definition (Opposite Fuzzy Set) [1]– Given a fuzzy set $A \subset X$, the opposite fuzzy set $\breve{A} \subset X$ with membership function $\mu_{\breve{A}}(x)$ is defined as

$$\breve{A} = \{(x, \mu_{\breve{A}}(x)) \mid x \in X, \mu_{\breve{A}}(x) \in [0,1]\} \tag{6}$$

where $\mu_{\breve{A}}(x) = f(x; \breve{a}, \breve{\delta})$.

The vector $a = (a_1, a_2, \ldots)$ and its opposite vector $\breve{a} = (\breve{a}_1, \breve{a}_2, \ldots)$ represent the points on different locations of the universe of discourse with $\mu(a_i) = \mu(\breve{a}_i) = 1$; $\breve{\delta}$ is generally optional for linear opposition and will cause the opposite shape modification compared to original δ.

Based on the defined concepts, a new image thresholding approach was proposed [1]. The algorithm defines a set A as dark pixels, and calculates the entropy of A. Then, it iteratively defines $\breve{A}$, the candidate opposite fuzzy set of A with different sizes starting from the brightest region, calculates the entropy of $\breve{A}$ and the difference between the entropies. The minimum difference indicates that $\breve{A}$ is the most probable opposite of A. An extension to this method was proposed by Tizhoosh and Sahba [27]. First, the center of the object of interest is determined interactively via user input. A window is constructed around the central point, and its size is increased incrementally in each iteration. Each iteration, dark pixels fuzzy set is determined for current window, and its opposite bright fuzzy set is found. The location of the window with maximum entropy difference between the two fuzzy sets is found. Then, the threshold is calculated as the average of representative numbers of both fuzzy sets of the selected window.

Opposition is a multi-faceted phenomenon and can defined in different ways. Hence, different ways exist to use the concept of opposite fuzzy sets for introducing new thresholding algorithms. In the following sections we examine another idea to incorporate opposite fuzzy sets within the image segmentation procedure.

V. Opposition in Image Processing

Utilizing oppositional concepts in image processing and computer vision applications opens new perspectives on how to look at and how to solve problems. Still, the meaning of opposite point or pixel is vaguely defined. Let us consider three types of opposition concept in an image. The first is with respect to color or intensity, the second is with respect to location, and the third is with respect to the direction.

Opposite color or intensity: the opposite of a pixel could be defined as the pixel that has the opposite color or intensity. So, the opposite of black is white, and the opposite of dark in gray-level images is bright (Figure 1). Opposite of color could be considered as the complement of it. For instance, in RGB the opposite of red is cyan and so on.

Opposite location: it may be difficult in some cases to have a concrete definition of the opposite pixel with respect to location. Having an image with one object, the opposite of a pixel inside the object could be a pixel in the background. The opposite of a pixel with respect to a reference point (*e.g.* an edge) is the pixel that has the same distance but in the opposite direction.

Opposite direction: consider a 3×3 window, the opposite of the point that is above the center is the point below the center. Similarly, the right point is opposite to the left point, and upper-right is opposite to lower-left (Figure 2).

Figure 1. Opposite intensity: point $\breve{x}$ (light) is opposite to x (dark). $\breve{x}$ can be defined as described in Section II. Nonlinear opposite intensities can only be defined if domain knowledge is available or can be extracted from data in an online manner or via offline simulation.

1	2	3
4	5	6
7	8	9

Figure 2. Opposite directions in a 3×3 window. Number 6 is opposite to 4, 8 is opposite to 2, 7 is opposite to 3, and 9 is opposite to 1. Number 5 is a special case, it can be considered as an opposite of itself or an opposite to all other cases.

VI. Proposed Algorithm

Using opposite fuzzy sets, a new image thresholding algorithm is described in this section. Oppositional Fuzzy Thresholding (OFT1) algorithm described in [1] used two fuzzy sets, where one of them is fixed while the other is changing. In our proposed algorithm (OFT2), the two sets are changing at the same time and can move in both directions. This way, the chance of finding the two sets that have minimum similarity is greater. A similarity measure is used to calculate the similarity of fuzzy sets in each iteration (this is another difference to OFT1 which uses entropy to recognize opposites). The aim of the algorithm is to minimize the similarity between the two sets, so they can be considered as opposites. Two fuzzy sets, A and B, are initialized based on the histogram of the input image. A^*, the center of A, between minimum and mean intensities, and B^*, the center of B is between mean and maximum intensities. In each iteration, four fuzzy sets are constructed, two around each existing sets with distance of Δ: AR and AL will be to the right and left of A, respectively and BR and BL will be to the right and left of B as shown in Figure 3. Similarities between fuzzy sets (A, AR, AL) and (B, BR, BL) are measured using the following similarity function [28]:

$$\eta = 2 - d((A \cap B), [1]) - d((A \cup B), [0]) \qquad (7)$$

where,

$$d(A, B) = \frac{1}{n} \sum_{i=1}^{n} |\mu_A(x_i) - \mu_B(x_i)| \qquad (8)$$

The sets with minimum similarity are chosen as the best approximation of the opposition. If the similarity is less than previous iteration, then current A^* and B^* points are replaced with the centers of sets that have minimum similarity. On the other hand, if the similarity is equal to the previous iteration, then the algorithm is considered to have reached the optimal solution. These steps are presented in Algorithm 1. In line 2, Δ is the distance of the centers of constructed sets from A^* and B^*, δ is the bandwidth of the membership function, α is the similarity of previous iteration, it is initialized with a large number (from empirical experimentation, 1000 is a good initialization). Initializing Δ to 1 gives the best results. The value of δ is determined dynamically depending on the width of the histogram, empirical experiments show that dividing by 50 obtain good results. Δ and δ are illustrated in Figure 4. Starting points A^* and B^* are initialized in lines 3-4. The points AR^*, AL^*, BR^*, and BL^* are calculated in lines 7-10. Fuzzy sets are constructed in line 11, using triangular membership functions. In lines 12-20, similarities between different fuzzy sets are calculated. Minimum similarity is found in line 22.

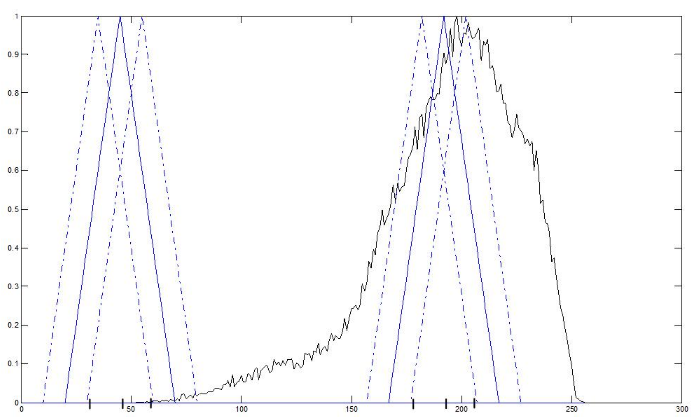

Figure 3. Proposed oppositional fuzzy set image thresholding algorithm, the six different sets (A, AR, AL, B, BR, BL), with their centers (A^*, AR^*, AL^*, B^*, BR^*, BL^*) are illustrated. The distance between $A^*$and $AR^*$is Δ, analogously for A^* and AL^*, B^* and BR^*, B^* and BL^*. The width of all fuzzy sets is 2δ.

Algorithm 1 Proposed image thresholding algorithm

```
1.  h = histogram of the image
2.  Δ = 1, δ = (max(h) − min(h))/50, α = 1000
3.  A* = (mean(h) + min(h))/2
4.  B* = (mean(h) + max(h))/2
5.  change = true
6.  Iterate
7.      AR* = A* + step
8.      AL* = A* − step
9.      BR* = B* + step
10.     BL* = B* − step
11.     Construct six fuzzy sets: A, AR, AL, B, BR, BL
12.     η1 = similarity(A,B)
13.     η2 = similarity(A, BR)
14.     η3 = similarity(A, BL)
15.     η4 = similarity(AR, B)
16.     η5 = similarity(AR, BR)
17.     η6 = similarity(AR, BL)
18.     η7 = similarity(AL, B)
19.     η8 = similarity(AL, BR)
20.     η9 = similarity(AL, BL)
21.     Find the sets Am&Bm with minimum similarity
22.     λ = min(η1, η2, . . . , η9)
23.     if λ < α
24.         A = Am
25.         B = Bm
26.         α = λ
27.     elseif λ = α
28.         change = false;
29.     end
30. until change = false
```

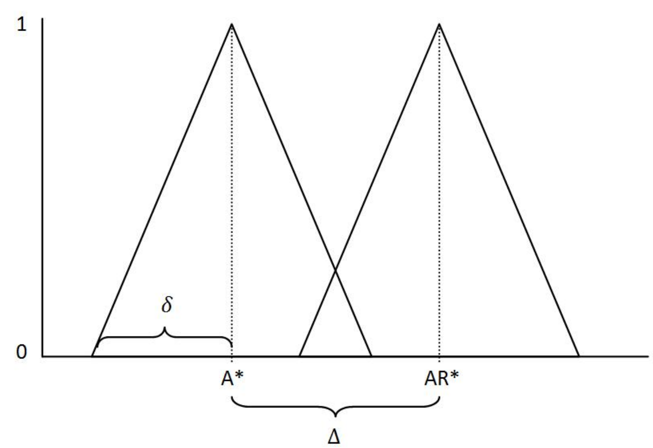

Figure 4. Illustration of Δ and δ. Δ is the distance between the centers of the original set and created sets to the right and left. δ is half the width of the fuzzy set.

VII. Experimental Results

In this section, the performance of the proposed algorithm is investigated. A test set of 10 different images was employed in the testing process. The images contain various objects, different levels of noisiness and clear or fuzzy boundaries. For each image, manually segmented image (GOLD) is provided. Performance is measured by comparing GOLD image with the binary image resulting from the thresholding algorithm. The performance measure is defined by

$$\gamma = 100 \times \left(1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} |I_G(i,j) - I_T(i,j)|}{N \times M}\right) \quad (9)$$

where N and M are height and width of the images, respectively, I_G is the GOLD image and I_T is the result of thresholding algorithm. The performance of both oppositional fuzzy thresholding algorithm OFT1 and the well-known Otsu algorithm [9] are compared with the proposed algorithm (OFT2). The test images with their corresponding GOLD images and the results of the three algorithms are shown in Figure 5. The performance measure for the three algorithms is presented in Table I. It can be observed that current algorithm achieved the highest average performance (94.73%) with the lowest standard deviation (2.42%). The low standard deviation indicates that OFT2 algorithm is better for general images than other two algorithms (higher applicability).

Table I
PERFORMANCE MEASURE OF OFT1, OTSU, AND OFT2 (PROPOSED)

Image	OFT1	Otsu	OFT2
Block	93.69	**94.60**	93.69
Fleck	15.10	95.93	**96.45**
Potato	**99.61**	98.03	97.37
Rad	98.52	98.17	97.76
Rice	93.15	93.78	93.58
Shadow	78.02	90.50	90.50
Stones	85.23	**95.22**	94.91
Text	90.55	78.21	**91.65**
Zimba	96.08	**97.47**	96.68
News	86.35	**95.07**	94.73
m	83.63	93.70	**94.73**
σ	24.97	5.90	**2.42**

VIII. CONCLUSIONS

Image thresholding is one of the most important stages in image processing applications, but yet it considered to be a challenging problem. Many thresholding algorithms have been proposed in literature, each based on a different approach. In this paper, the problem was considered from a new perspective which was introduced recently. By utilizing opposition concepts and fuzzy sets, the proposed algorithm searches for the two fuzzy sets that have the minimum similarity, which can be considered the opposites of each other. Such technique could obtain good results for general types of images. As a future work, the performance of using different membership function shapes, other than triangular, will be investigated. Also, the result of choosing different starting points will be studied. Furthermore, the algorithm could be tested as a local thresholding technique rather than global one.

ACKNOWLEDGMENT

The authors would like to thank King Abdulaziz City for Science and Technology for their scholarship and for their valuable support.

REFERENCES

[1] H.R.Tizhoosh, "Opposite fuzzy sets with applications in image processing," in *Proceedings of IFSA World Congress (International Fuzzy Systems Association)*, pp. 36–41, 2009.

[2] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–168, 2004.

[3] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet*, vol. 1, pp. 695–701, 2005.

[4] M. Ventresca and H. R. Tizhoosh, "Improving the convergence of backpropagation by opposite transfer functions," in *IEEE International Conference on Neural Networks*, pp. 4777–4784, 2006.

[5] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution algorithms," in *IEEE Congress on Evolutionary Computation*, pp. 2010–2017, 2006.

[6] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[7] H. R. Tizhoosh, "Reinforcement learning based on actions and opposite actions," in *Proc. of the first ICGST International Conference on Artificial Intelligence and Machine Learning AIML 05*, vol. 05, pp. 94–98, 2005.

[8] H. R. Tizhoosh, "Opposition-based reinforcement learning," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 10, no. 4, pp. 578–585, 2006.

[9] N. Otsu, "Threshold selection method from gray-level histogram," *IEEE Trans. Sys. Man Cybern.*, vol. SMC-9, no. 1, pp. 62–66, 1979.

[10] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Graph. Models Image Process*, vol. 29, pp. 273–285, 1985.

[11] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1985.

[12] W. Tao, H. Jin, and L. Liu, "A new image thresholding method based on graph cuts," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. I–605 – I–608, 2007.

[13] Y. Yang, C. Zheng, and P. Lin, "Image thresholding based on spatially weighted fuzzy c-means clustering," in *The Fourth International Conference on Computer and Information Technology, CIT*, pp. 184 – 189, 2004.

[14] J. Kang and W. Zhang, "An approach for image thresholding using cnn associated with histogram analysis," in *International Conference on Measuring Technology and Mechatronics Automation, ICMTMA*, pp. 421 – 424, 2009.

[15] X. Ren, "An optimal image thresholding using genetic algorithm," in *International Forum on Computer Science-Technology and Applications, IFCSTA*, pp. 169 – 172, 2009.

[16] Z. Lin, Z. Wang, and Y. Zhang, "Image thresholding using particle swarm optimization," in *International Conference on MultiMedia and Information Technology, MMIT*, pp. 245 – 248, 2008.

[17] S. Rahnamayan and H. R. Tizhoosh, "Image thresholding using micro opposition-based differential evolution (micro-ode)," in *IEEE Congress on Evolutionary Computation, CEC*, pp. 1409–1416, 2008.

[18] F. Khalvati, H. R. Tizhoosh, and M. D. Aagaard, "Opposition-based window memoization for morphological algorithms," in *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing, CIISP*, pp. 425–430, 2007.

[19] H. Lin and H. Xingshi, "A novel opposition-based particle swarm optimization for noisy problems," in *Proceedings - Third International Conference on Natural Computation, ICNC*, vol. 3, pp. 624–629, 2007.

[20] A. R. Malisia and H. R. Tizhoosh, "Applying opposition-based ideas to the ant colony system," in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium, SIS*, pp. 182–189, 2007.

[21] K. Yuchi, E. Yasunori, and M. Sadaaki, "Fuzzy c-means algorithms for data with tolerance based on opposite criterions," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 10, pp. 2194–2202, 2007.

[22] H. R. Tizhoosh, "Image thresholding using type ii fuzzy sets," *Pattern Recognition*, vol. 38, no. 12, pp. 2363–2372, 2005.

[23] V. NovÃąk, "Antonyms and linguistic quantifiers in fuzzy logic," *Fuzzy Sets and Systems*, vol. 124, no. 3, pp. 335–351, 2001.

[24] E. Trillas and S. Guadarrama, "What about fuzzy logic's linguistic soundness?," *Fuzzy Sets and Systems*, vol. 156, no. 3, pp. 334–340, 2005.

[25] Y. Y. Golota, "On a certain formalization of antonyms logic," *Fuzzy Sets and Systems*, vol. 45, no. 3, pp. 335–340, 1992.

[26] E. Trillas, C. Moraga, S. Guadarrama, S. Cubillo, and E. CastiÃśeira, "Computing with antonyms," *Studies in Fuzziness and Soft Computing*, vol. 217, pp. 133–153, 2007.

[27] H. R. Tizhoosh and F. Sahba, "Quasi-global oppositional fuzzy thresholding," in *International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1346–1351, 2009.

[28] D. H. Park, S. H. Lee, E. H. Song, and D. Ahn, *Similarity computation of fuzzy membership function Pairs with similarity measure*, vol. 4682 LNAI of *Lecture Notes in Computer Science*. Springer-Verlag, 2007.

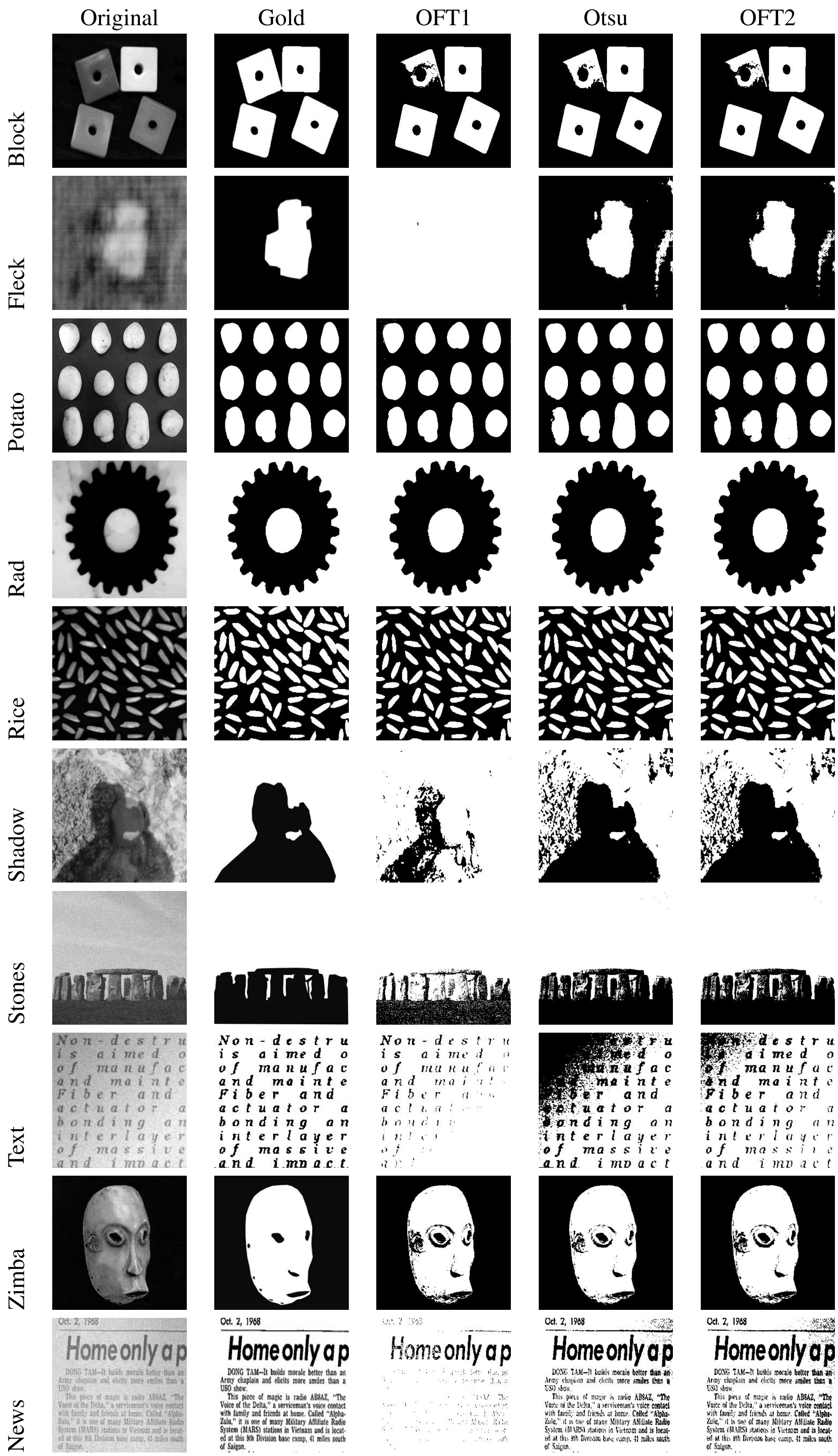

Figure 5. Test set of general images and corresponding results for the three algorithms, OFT1, Otsu method, and OFT2 (proposed).