

Region-based Image Segmentation by Watershed Partition and DCT Energy Compaction

Chi-Man Pun*, Ning-Yu An and C. L. Philip Chen

Department of Computer and Information Science, University of Macau
Macau SAR, China

E-mail: {cmpun, ya974141, philipchen}@umac.mo

Received 21 March 2011

Accepted 17 November 2011

Abstract

An image segmentation approach by improved watershed partition and DCT energy compaction has been proposed in this paper. The proposed energy compaction, which expresses the local texture of an image area, is derived by exploiting the discrete cosine transform. The algorithm is a hybrid segmentation technique which is composed of three stages. First, the watershed transform is utilized by preprocessing techniques: edge detection and marker in order to partition the image in to several small disjoint patches, while the region size, mean and variance features are used to calculate region cost for combination. Then in the second merging stage the DCT transform is used for energy compaction which is a criterion for texture comparison and region merging. Finally the image can be segmented into several partitions. The experimental results show that the proposed approach achieved very good segmentation robustness and efficiency, when compared to other state of the art image segmentation algorithms and human segmentation results.

Keywords: image segmentation, energy compaction, cosine transform, watershed.

1. Introduction

In the literature, segmentation has been known as the process of dividing an image into a subset of connected regions based on an application defined criteria. For natural scenery images, it usually means the separation of the image into its constituent objects or feature types (e.g. sky, trees, buildings, etc) ². Despite the plenty of research outcome on such topic, it still continues to be a challenging problem in image processing and computer vision². The difficulties mainly results from the fact that different people can define textures in application-dependent ways or with

different perceptual motivations, and there is no generally agreed-upon definition.

Among the different segmentation methods, unsupervised image segmentation algorithms have been widely applied for its generalization, which can be broadly divided in three categories: region-based, graph-based and model-based ones. More specifically, region-based image segmentation approaches ³⁻⁶ try to separate connected image regions by grouping neighboring pixels based on brightness, color, and texture. Afterwards, adjacent regions are merged, under some criterion involving homogeneity or region boundary sharpness. In the past, many research efforts have been done regarding graph-based image segmentation algorithms ⁷⁻⁹. These approaches use an

* Corresponding author.

undirected graph to represent the image pixels. For each graph edge, there is an edge weight representing pair wise pixels similarity. Therefore, the objective is to minimize the cost of splitting the graph into a number of disjoint subsets. Differently, the model based methods strive to capture the underlying structure of the texture and classify image pixels into different segmented regions¹⁰⁻¹². And a variety of different models are employed, including Hidden Markov Models (HMM), anisotropic diffusion model, and general statistical models based on filter responses.

In this paper, our work focuses on the region-based algorithm, and a novel approach is proposed, which combines watershed and region merging by intensity and texture extraction. To the best knowledge of ours, the most widely investigated texture representation methods have been windowed Fourier transforms, the Gabor representation^{13, 14}, Wavelet transforms^{15, 16}, local histograms, and the local structure tensor, etc. However, the main drawback of Gabor filtering is that the excessive computational effort can't be avoided. To address these limitations, the discrete cosine transform is adopted for texture representation, resulting in a fast segmentation and energy compaction approach.

Furthermore, the main procedure of the presented method can be summarized into three steps. Firstly, the improved watershed transform¹⁷⁻¹⁹ is implemented in the input gray-scale image and the image is divided into several disjoint regions, while each region is labeled by a unique number. Second, the regions are combined according to their similarities which are calculated by intensity mean, intensity variance and region pixels until they reach a certain amount. Finally, the discrete cosine transform²⁰⁻²² is utilized to obtain the spectral information in a maximum rectangle of each region^{23,24}. The residual regions are merged corresponding to region similarities by involving texture information and region intensity. The flow chart of our method is given in Fig 1.

The rest of the paper is organized as follows. In Section 2, we give a brief review on improved watershed transformation as well as the knowledge about discrete cosine transform. In Section 3, details of the proposed algorithm for cost computing and region merging will be described. Experimental results are presented in Section 4. Conclusions are given in Section 5.

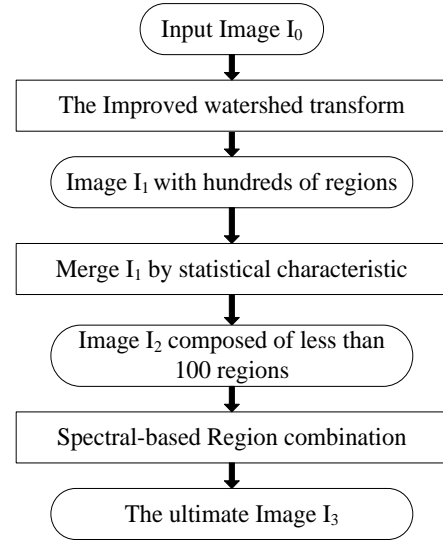


Fig 1. The flow chart of our proposed algorithm.

2. The Related Work

2.1. The Watershed Transform and Improvement

In geography, a watershed is the ridge that divides areas drained by different river systems. A catchment basin is the geographical area draining into a river or reservoir. The well-known watershed segmentation algorithm is the classic discrete Vincent-Soille flooding watershed algorithm¹⁷⁻¹⁹.

Define D as a set of gray-scale image pixels. Let $f : D \rightarrow \mathbb{N}$ be a function assigning an integer value to each $p \in D$, and h_{\min} and h_{\max} be the minimum and maximum gray value. Then, the recursion with the gray level h increases from h_{\min} to h_{\max} , where the implementation of the algorithm, can be defined as

$$X_{h_{\min}} = \{p \in D \mid f(p) = h_{\min}\} = T_{h_{\min}} \quad (1)$$

$$X_{h+1} = \text{MIN}_{h+1} \cup IZ_{T_{h+1}}(X_h), \quad h \in [h_{\min}, h_{\max}) \quad (2)$$

where X_h denotes the union of the set of basins computed at level h .

$$T_h = \{p \in D \mid f(p) \leq h\} \quad (3)$$

MIN_{h+1} denotes the union of all regional minima at altitude h , $IZ_{T_{h+1}}(X_h)$ is the union of the geodesic influence zones of the connected components of X_h . The watershed of (Wshed) f is the complement of $X_{h_{\max}}$ in D :

$$(\text{Wshed}) f = D \setminus X_{h_{\max}} \quad (4)$$

Next, the output of the watersheds algorithm is the input image into its different catchment basins, each of which is characterized by a unique label. At the same time, watershed points are equidistant to two nearest connected catchment basins that are characterized by 0.

However, the main disadvantage of this approach is that the watershed transformation may produce excessive over-segmentation for most natural images (Fig 4.a). Therefore, some pre-processing may be necessary to produce a segmentation that better reflects the arrangement of objects within the image. To achieve such purpose, we compute the gradient magnitude of the image which has high pixel values along object edges, and low pixel values everywhere else via Sobel liner filtering method (See Fig 2. and Eq.(5)), which can detect edge strength and direction resulting in a better noise-suppression (smoothing) characteristic.

Z ₁	Z ₂	Z ₃
Z ₄	Z ₅	Z ₆
Z ₇	Z ₈	Z ₉

Fig 2. Image pixel neighborhood

The Sobel filter is

$$g = [G_x^2 + G_y^2]^{1/2} = \{ [(Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)]^2 + [(Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)]^2 \}^{1/2} \quad (5)$$

After using the filter to enhance the gradient, the marker-controlled method is employed to eliminate the ambiguous magnitude and reduce the number of regions. Firstly, the locations of all regional minima in an image are searched, and these extraneous minima are eliminated with the function which computes the set of "low spots" in the image that are deeper (set the height threshold 10) than their immediate surroundings. Then, the minima imposition technique modifies a gray-scale image so that regional minima occur only in marked locations. Finally, the watershed transformation is performed for marker-modified gradient image.

The procedures of our improved watershed transform are shown in Fig 3, which is a detailed first step in Fig 1. Then the comparison between applying watershed transform directly and watershed after noise

elimination is provided in Fig 4. From Fig. 4 we can see that most of the insignificant regions in b are reduced contrast to a, and the segmentation performance of b is much better.

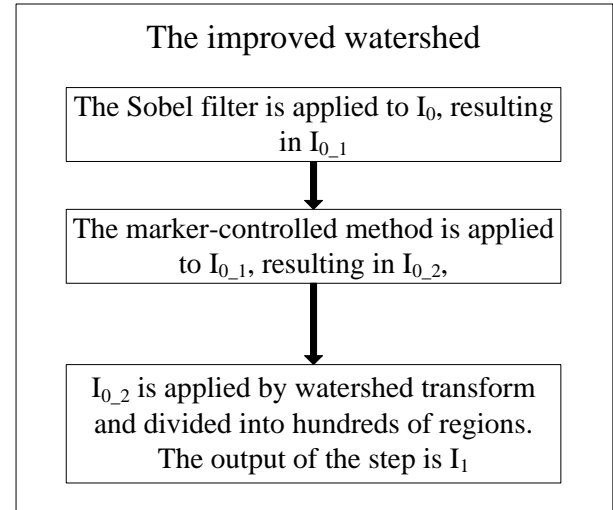


Fig 3. The procedures of the improved watershed transform.

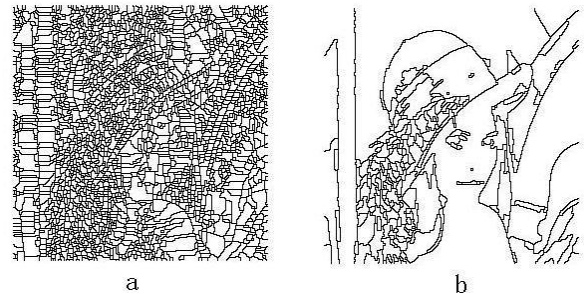


Fig 4. (a) is the image "Lena" applied watershed transform immediately, (b) is "Lena" applied improved watershed transform.

2.2. The Maximum Rectangle in an Irregular Region

After watershed transform, the image is substituted by region adjacency graph and each region is disjointed with the others. The regions have irregular shapes but the DCT transform need to be implemented on rectangle, so we get the maximum rectangle in a region for substituting the texture of the whole region. Apparently, some deviation is brought with the substitution, but fortunately the primary features in the regions can be preserved^{23, 24}.

In terms of this, one can firstly imagine a ball with a diameter of one pixel rolls pixel by pixel just inside the boundary (which means the rolling path borders on boundary of the region and included in the boundary) and allow the ball roll around clockwise until the ball encounters the start point again. The rolling path is composed of a series of pixels whose coordinates and the relative positions to the boundary are memorized.

Suppose that one pixel (x_0, y_0) in the rolling path is on the inner position of the boundary, the initial distance between two obstacles along X direction is defined as w and Y direction is defined as l .

Then, we extend the width from the pixel along the direction of X^+ and X^- , until the boundary obstacles are encountered. Afterwards, the l along the direction of Y^+ and Y^- is expanded pixel by pixel, each step the measure of width w long the direction of X^+ and X^- is performed over again once the length l is increased, and the rectangle area is computed. The rectangle area equals $l * w$. The length is extended until it meets boundary obstacle. After the comparison of all the rectangles along the Y direction, the maximum rectangle at the position (x_0, y_0) and its width and length are memorized.

Next, by comparing all the maximum rectangle areas of each pixel on the rolling path, the largest area among all those rectangle areas can be determined. The result of the algorithm is shown in Fig 5.

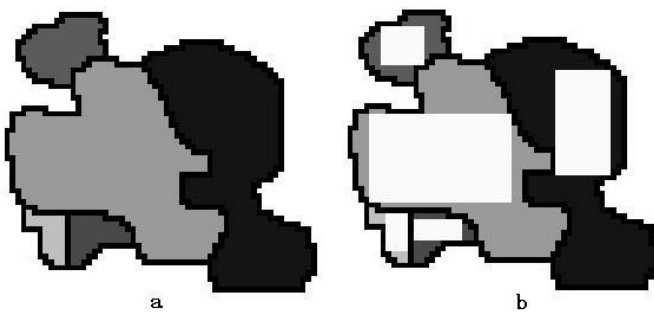


Fig 5. (a) includes 5 irregular regions, and (b) is the expression of the algorithm while the white blocks are 5 maximum rectangles in the 5 regions.

2.3. The Discrete Cosine Transform and its application

2.3.1. The DCT introduction its characteristic

The most common 2-D DCT definition²⁰⁻²² is

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{6}$$

$$\alpha(u) = \begin{cases} \sqrt{1/N} & \text{for } u = 0 \\ \sqrt{2/N} & \text{for } u \neq 0 \end{cases} \tag{7}$$

The definition of $\alpha(v)$ is the same as $\alpha(u)$.

The DCT transform transforms the whole image to a coefficient energy matrix by intensities, and also it depicts excellent energy compaction for highly correlated images. Since the uncorrelated image has more sharp intensity variations than the correlated image, the former has more high frequency content than the latter.

For low frequency image, DCT operation provides ideal energy compaction in the low frequency region. Furthermore, when images have progressively high frequency and spatial contents, the transform coefficients spread over low and high frequencies. Some examples are shown in Fig 6.

Fig 6 comprises of 4 image classes. The left images are the original images while the right images are energy compaction pictures. Comparing Fig 6.a with Fig 6.b, the former image has more sharp intensity variations than the latter one. Consequently, transform coefficients of the former spread over middle and high frequencies, while the ones of the latter spread over low and middle frequencies. Since the original images reside in periodicity, the pixels in energy matrix corresponding to particular frequency and intensity variation periodicity have more energy magnitude.

The left image of Fig 6.c is a stochastic image, so it has progressively high frequency. Also, Fig 6.d has high spatial contents and more variation texture, therefore the energy is distributed over most part of the corresponding energy image.

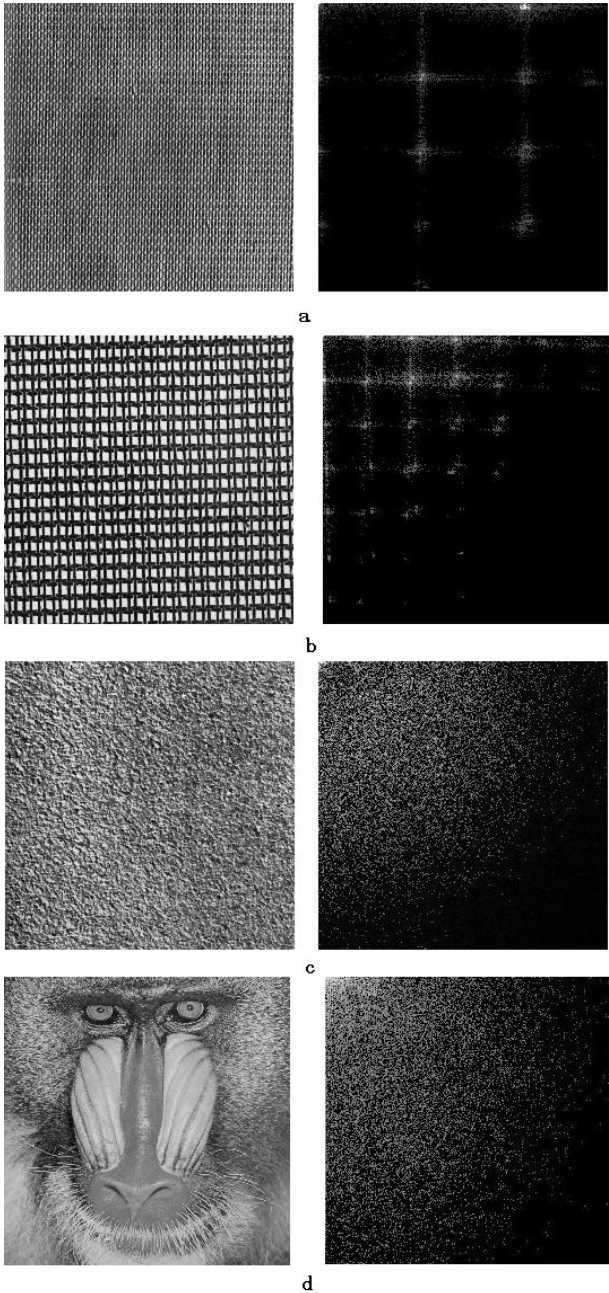


Fig 6. Energy compaction of DCT

2.3.2. The application of DCT transform in our method

As the texture images in Fig 6.a, b and c, we choose 40 synthetic texture images to test the DCT's performance of energy compaction.

The resolution of the original images is 512*512. Since the size of different rectangles is unequal, so we get portion of the each original image by the resolution

of 300*300, and discard the residual areas. Then the new image set is compared with the original one to match the textures.

The matrices after image DCT transform are noted as C_i , which have the same sizes as before transformation.

d_1 and d_2 are the width and length of one matrix C_1 with the resolution 512*512, d_3 and d_4 are the width and length of DCT matrix C_{1_r} of the 300*300 image. Owing to the DCT matrix has frequencies spread from low to high, we consider each side to be cut into 3 parts which represents the low frequency, middle frequency and high frequency, and they are denoted as d_{i_3} ($i=1...4$). d_1 and d_2 are not divisible, the rounding integers are obtained, so the d_{1_3} and d_{2_3} are 170. Based on the sides cutting, the DCT matrices can be divided into 9 parts averagely. The top left of the matrix has the lowest frequencies; the top middle of the matrix has the low and middle frequencies, and etc. Therefore, the contrast of two DCT matrices is established between the corresponding two parts.

Then sum the absolute value of each part in the DCT matrix, we can get the DCT frequency energy. Such as the left top part of C_1 is given below,

$$\text{Vector}_{1_1} = \sum_{i=1}^{d_{1_3}} \sum_{j=1}^{d_{2_3}} |C_1(i, j)| \quad (8)$$

Here Vector_{1_i} ($i = 1, 2, \dots, 9$) is defined as the i part of DCT energy in C_1 . Similarly, Vector_{2_i} is the i part of DCT energy in C_{1_r} .

According to our experiments, Vector_{1_i} and Vector_{2_i} don't have correlations directly, they need be normalized by their size, which is formulated by formula (9).

$$\begin{aligned} \text{Vector}_{1_i} &= \text{Vector}_{1_i} / (d_1 * d_2) \\ \text{Vector}_{2_i} &= \text{Vector}_{2_i} / (d_3 * d_4) \end{aligned} \quad (i=1...9) \quad (9)$$

Finally, the difference of two DCT matrices is given as:

$$\eta = \sum_{i=1}^9 (\text{Vector}_{\max_i} / \text{Vector}_{\min_i}) \quad (10)$$

Here Vector_{\max_i} denotes the larger value between Vector_{1_i} and Vector_{2_i} , Vector_{\min_i}

denotes the smaller value between $Vector_{1_i}$ and $Vector_{2_i}$.

η can estimate the similarity of two texture regions. If two regions have analogous textures, their corresponding vectors will have approximate value and the ratio is close to 1, otherwise, the ratio is larger. Hence η will have smaller value when two compared region textures are similar, or it will hold a larger value when two regions have great discrepancies.

Based on the characteristic of η , the two synthesis image sets, each of which contains 40 images, are used for texture matching. Each of images in the set C_{i_r} is matched with all the images in the set C_i , and totally 40*40 times matching are required. The result is shown in Table I.

The Corresponding Texture matching value η	Times	Ratio
The minimum value	34	85%
The three minimum value	2	5%
The five minimum value	3	7.5%
others	1	2.5%

Table I. The texture matching by DCT energy.

In Table I first row, the Corresponding Texture matching value means that each matrix in C_{i_r} corresponds to one matrix in C_i which has the same texture and is the original image's DCT matrix of C_{i_r} . Also, C_{i_r} is matched with the other matrices in C_i and the relative magnitude of the corresponding value is extracted. Table I demonstrates that 34 matrices in the set C_{i_r} can find their same texture matrices in the set C_i because they have the minimum value η , meanwhile only one image can't find its corresponding picture within 5 minimum η which means the 5 most similar images.

Therefore, according to the characteristic of DCT transform, we can extract texture and distinguish regions with different texture by comparing their high, middle or low frequency energy compaction.

3. Fast Region Merging Algorithm

As introduced in Fig 1, the image denoted by I_1 is divided into hundreds of regions after improved watershed transform. The following are the fast region combination, an important aspect of which is how to define the similarities (We call it region cost)

Here the region merging part in our algorithm which is composed of the next two stages, is described as follows: in stage 2, the over-segmented image I_1 is merged according to two region features: pixel intensity mean and pixel intensity variance, until a threshold T_1 is reached. In stage 3, as the procedures in stage 2, the residual regions are incorporated sequentially. But the difference is the definition of region discrepancies which is region pixel intensity and texture similarity by DCT energy computation proposed in last section.

3.1. Region Merging By Intensities

In the stage, the initial image is I_1 , which can be represented by an undirected adjacency graph, and the data structure is:

$$I_1 = (V, E, W) \tag{11}$$

Where $V = \{v_1, v_2, \dots, v_n\}$, V is the set of nodes representing different regions in graph I_1 , and v_i memorizes three features in the region i : the number of pixels, the mean, and the variance in region i .

$$E = \{e_{i,j} = (V_i, V_j), i, j \in (1, 2, \dots, N), i \neq j\} \tag{12}$$

E denotes the set of edges between the adjacent regions; obviously it records the neighboring regions of each region.

$$W = \{w_{i,j} = \cos t(e_{i,j}), i, j \in (1, 2, \dots, N), i \neq j\} \tag{13}$$

Here $w_{i,j}$ is the weight of each $e_{i,j}$ and it denotes the adjacent region dissimilarity (region cost).

The procedure of this stage in a flow chart is given in Fig 7, and the detailed description of Fig 7 is provided below.

Input: I_1 is represented by a graph, the pixels in each region (v_i) is labeled by a unique number, different regions are labeled from 1 to N , and the pixels partition different regions are labeled by 0.

Step 1: Search the whole graph and find the right or lower neighbors of each region as $e_{i,j}$.

Step 2: Here T_1 denotes the region threshold and usually it is set to 50 to 100 for the reason that the small patches can be combined with other regions just by their size and intensity. In this step, we judge whether the region amount is less than the threshold. If not, iterate step3 to step5.

Step 3: For each pair of adjoining regions, calculate the mean of pixels, the variance and the pixel's amount in each region. The variance is defined as

$$\sigma = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \quad (14)$$

Where, X_i is the intensity of a pixel i and \bar{X} denotes mean pixel value in a region. Variance is a measure of pixel intensities' variation and deviation in a region. And the formulation of the region cost between adjacent regions a and b is

$$\begin{aligned} diff_m &= |\mu_a - \mu_b| \\ diff_v &= |\sigma_a - \sigma_b| \end{aligned} \quad (15)$$

$$w_{a,b} = (n_a \times n_b)^{rV1} \times (w_m \times diff_m + w_v \times diff_v) \quad (16)$$

Here, n_a denotes the number of pixels in region a , μ_a denotes the mean of pixel intensities in region a , and σ_a denotes the variance in region a . Then $diff_m$ and $diff_v$ are the difference of mean and variance between neighbor regions a and b . In Eq. (16), w_m and w_v indicate the weight of $diff_m$ and $diff_v$, and usually $diff_m$ is several-fold bigger than $diff_v$. In terms of our experiments, w_m and w_v , which are set to 1.5 and 0.2 separately, generate the better region combination results among all the weight values.

Also, the parameter $rV1$ is the weight of region size. When either of n_a and n_b is less than T_1 , $rV1$ equals to 1, otherwise $rV1$ equals to 1.5, such that the small patches are assigned smaller region cost for combined easier.

Eventually, region cost of regions a and b is calculated and denoted by $w_{a,b}$.

Step 4: In this procedure, the minimal $w_{a,b}$ among all the region cost $w_{i,j}(i, j \in 1, \dots, N)$ is selected for merging. The region a, b and their boundary pixels are re-labeled by the smaller tag between a and b .

Step 5: $\{V, E, W\}$ are updated during the procedure. n, μ, σ and $e_{i,j}, w_{i,j}$ of the merged regions need to be re-calculated.

From Eq. (16), we can find that the cost of merging is in direct relation to the area of regions, and the texture of small patches is more changeful, thus the texture will not be involved in the stage. Finally, the image is divided into T_1 regions.

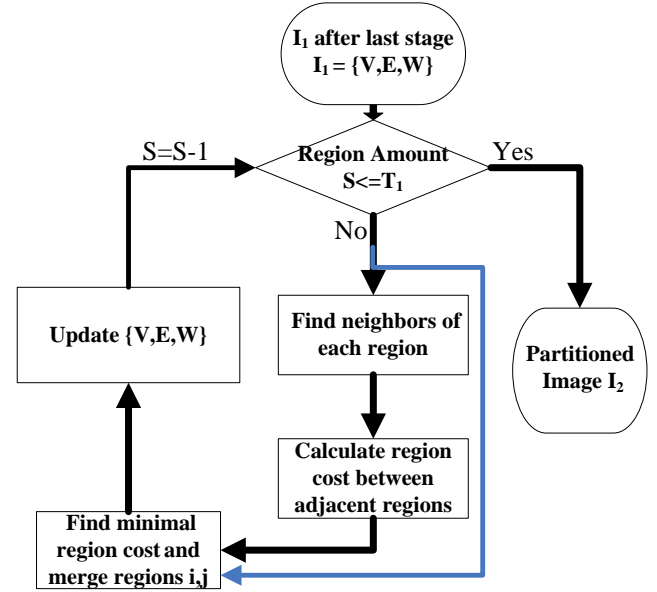


Fig 7. Region cost calculation flowchart

3.2. Merging Considering Region Textures

Followed from stage 2, the initial condition in the stage is image I_2 composed of about tens of regions and the primary merging steps are the same as stage 2.

However, the dissimilarity is the computation of regions cost. The DCT energy comparison in maximum rectangles, which is proposed in subsection 2.3.2, is used for compared two texture regions.

The maximum rectangle covers most area of a region, so most textures in a region are taken into account, and the region cost here between two regions a and b is defined:

$$w_{a,b} = (n_a \times n_b)^{rV2} \times (w_{m,2} * diff_m + W_c \times \eta_{a,b}) \quad (17)$$

The definition of n_a, n_b and $diff_m$ is the same as Eq (16). $\eta_{a,b}$ denotes DCT energy difference between a and b . The parameter $rV2$ is the weight of region size. In the stage, region merging mainly depends on texture

than region size, so $rV2$ is reduced and set to 0.5. $w_{m,2}$ and w_c indicate the weight of $diff_m$ and $\eta_{a,b}$, and they are set to 1 and 5 respectively in our program. At last, the same as Eq (16), $w_{a,b}$ is region cost.

In the stage, the threshold T_2 which denotes the final output region number is also needed and input by user. Finally, the program can output the segmented image I_3 , which has required number of regions T_2 .

4. Experimental Results

4.1. Segmentation results of the first and second stages

The results of the first stage are the images after improved watershed transform. During the second stage, the threshold T_1 need be fixed. Generally, the range of T_1 is about 50 to 100. If the initial image's resolution is high, E. g. 512*512, T_1 can be larger since that each patch in I_2 is enough for feature extraction; otherwise T_1 should be smaller. In the following experiments, the original images' resolution is 256*256, and T_1 is set to 50. The two stages' results of "Lena", "baboon" and "jet" are shown in Fig 8. Comparing the left images and right ones, many fragments that don't have great difference with their neighbors in intensity will get the high merging priority.

From the results of the second stage, the regions are relative larger, consequently they are convenient to extract textures in next stage. Yet, we can find out that some regions, such as the hair in picture "Lena" and the tree in picture "Lake", that has different mean gray value but analogous textures, aren't able to be combined with each other. Then the stage 3 is required.



Fig 8. The results of three images "Lena", "Lake" and "Jet" from top to bottom. The left column is segmentation results after stage 1, the improved watershed transform, and the right column is the results after stage 2.

4.2. The final results and comparing with human segmentations

The proposed technique is applied on the Berkeley segmentation dataset²⁵ which is constituted of a wide range of pictures from landscapes, animals to human beings, various objects. Moreover, the dataset include ground truth images which is hand-labeled by subjects.

The comparison between the proposed segmentation obtained and the ground truth segmentation is obtained by computing the Precision and Recall, as defined in²⁶. Let S_M be boundaries obtained by the proposed method for segmentation and S_G be boundaries of the ground truth segmentation, then the precision P is defined as

$$P = \frac{\#(S_M \cap S_G)}{\#(S_M)} \quad (18)$$

where $\#(S)$ is the number of pixels of the boundaries in the segmentation S and $(S_M \cap S_G)$ is the intersection of S_M and S_G . The recall is defined as

$$R = \frac{\#(S_M \cap S_G)}{\#(S_G)} \quad (19)$$

The measure F is defined as:

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} \quad (20)$$

where α equal to 0.5 is an overall measurement of the quality of segmentation. A good segmentation has values of P, R and F close to 1.

The implementation of our method is good at partitioning the graphs which have distinct objects, but our segmentations will not so accurate when the objects and background which are difficult to distinguish.

Some comparisons between human segmentation results and our benchmark results are provided in Fig 9 and Fig 10. In the two figures, the left column is the human segmentations and the right column is results of the proposed method which is based on the original images. Fig 9 and Fig 10 shows that by controlling the final region number parameter T_2 , our method can find the whole object from the background, however some details are ignored in our methods and the superiority of human segmentation is evident: indeed, in all images, humans provide a segmentation where objects are correctly extracted and identified from the background

Table II illustrates a quantitative comparison between the proposed algorithm and the human segmentations. For each image of our proposed method results, we compare it with all of the human segmentations from the original image. The human segmentations of Table II images a-d are the left column of Fig 9.a-d, and the human segmentations of Table II images e-g are the left column of Fig 10.e-g. Here the region number threshold T_2 is set to 15 for the reason that this threshold gives the better F value.

The results of image a and g are best and F value is greater than 0.6, while image e has the worst segmentation result since the human segmentations of Fig 10.e has many textures and patterns which are hard to find in our method. Generally, most results of our method can give ideal segmentations.

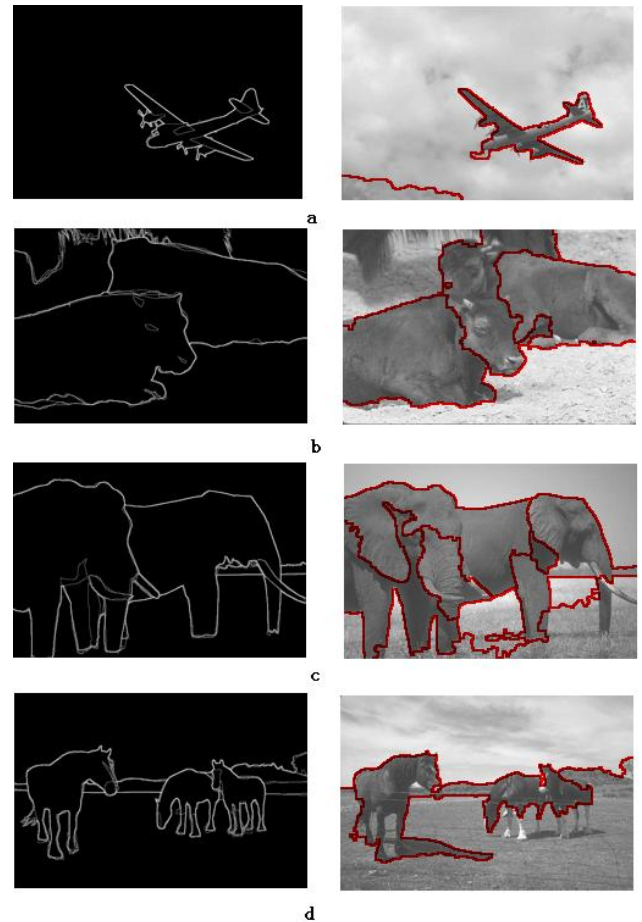


Fig 9. The left column is the human segmentations and the right one is our segmentation results

Images	P	R	F
a	0.61	0.91	0.73
b	0.43	0.62	0.51
c	0.39	0.88	0.54
d	0.52	0.69	0.59
e	0.31	0.42	0.36
f	0.50	0.81	0.62
g	0.33	0.73	0.45

Table II. Quantitative precision and recall comparison between our method and human segmentations

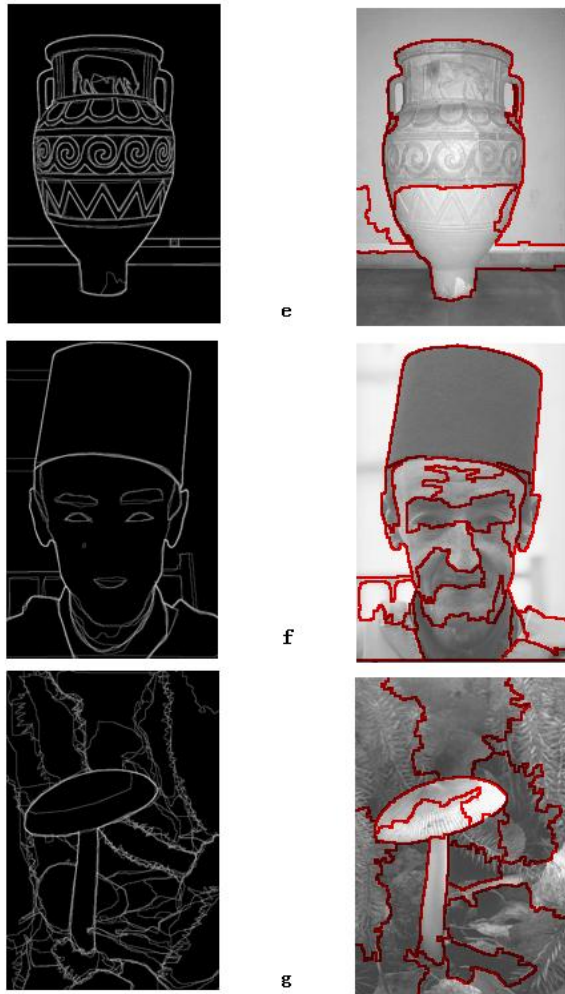


Fig 10. The left column is the human segmentations and the right one is our segmentation result

4.3. The results and comparing with N-Cut method

The obtained segmentation of ‘Lena’ between our method and Normalized-Cut algorithm is shown in Fig 11. **The N-cut method segment the image with the resolution of 128*128. We implement the program with the resolution of 128*128.** Our method partitions the object more accurately, which improves performance and convenient for some operations subsequently.

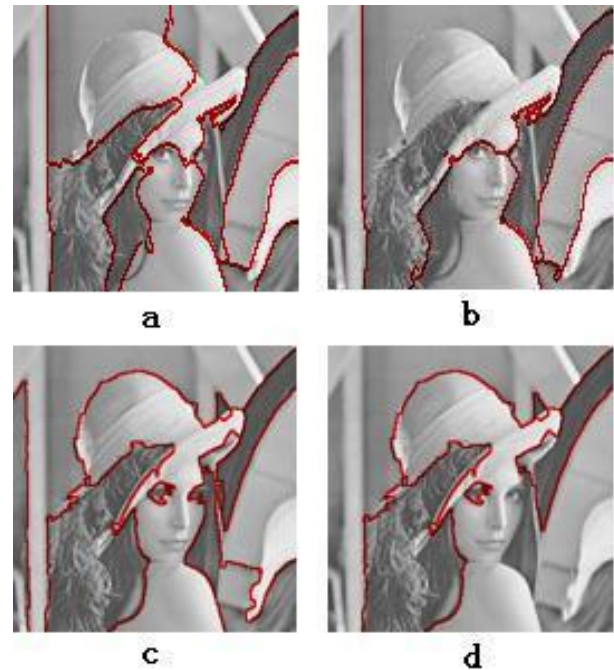


Fig 11. The comparison of segmentation of ‘Lena’ between our method and Normalize Cut algorithm. a and b are divided into 10 and 6 regions respectively by N-Cut algorithm. c and d are divided into 10 and 6 regions respectively by our method.

Besides, the comparison of the image “Baboon” and “Lake” between our method and N-Cut algorithm are shown in Fig 12 and 13. The intensity’ difference of regions is little in Fig 12; therefore the distinction of each part is evident by texture. From the results, it is obvious that our method is able to partition the different texture parts into the accurate regions compared with the other. Also in Fig 13, from visual perception perspective, the partitions of our method present more precise and reasonable.

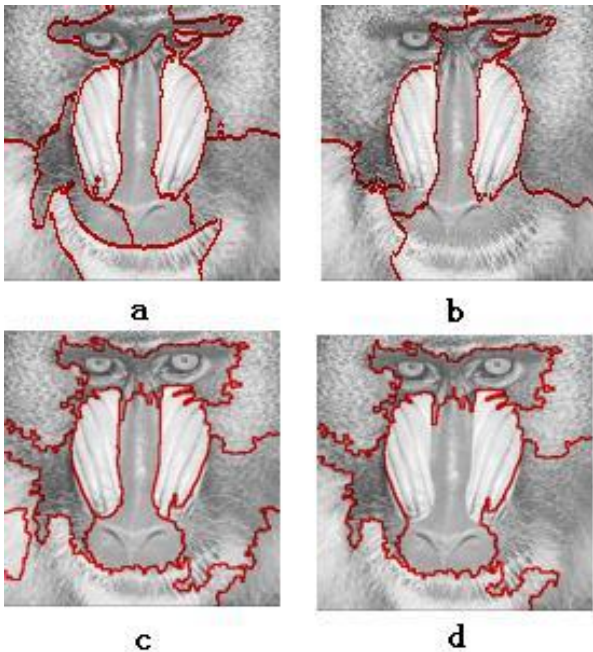


Fig 12. The comparison of segmentation of ‘Baboon’ between Normalize Cut (a, b) algorithm and our method (c, d). Figure a, c are 10 regions while b, d are 6 regions.

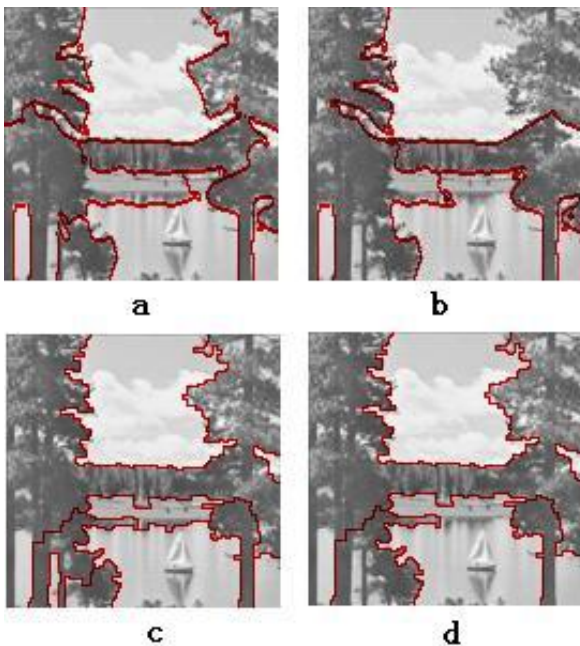


Fig 13 The comparison of segmentation of ‘Baboon’ between our method (c, d) and Normalize Cut (a, b) algorithm. Figure a, c are 10 regions while figure b, d are 6 regions.

5. Conclusion

In this paper, we propose a novel image segmentation approach based on watershed transformation, DCT transform and region merging. We

improve the watershed transform by preprocessing techniques: edge detection and marker. Then during region merging, the region size, mean and variance features are used to calculate region cost for combination in the first stage, and in the second merging stage we utilize DCT transform for texture comparison. Finally the image is segmented into several partitions which can be controlled easily.

The proposed algorithm is evaluated for several grayscale images, and the results are attractive with respect to most cases, which can accurately extract the segmentation. We also give a comparison between our method and N-Cut method. The experimental results demonstrate our method is more precise and reasonable, and is comparable with the human segmentation results.

In the future work, the extension for the segmentation of color images is a possible research direction, and parameters adjustment using more adaptive approaches can also be studied.

Acknowledgements

The authors would like to thank the referees for their valuable comments. This work was supported in part by the MYRG grant of Research Committee of the University of Macau.

References

1. Bomans, M., et al., 3-D segmentation of MR images of the head for 3-D display. *Medical Imaging, IEEE Transactions*, 1990. 9(2): p. 177-183.
2. C.-M. Pun, H-M Zhu and W. Feng, " Real-Time Hand Gesture Recognition using Motion Tracking," *International Journal of Computational Intelligence Systems*, 4(2), pp. 277 - 286, 2011..
3. Karoui, I., et al. Region-Based Image Segmentation Using Texture Statistics And Level-Set Methods. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference 2006*.
4. Y. Deng, and B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 2001. 23(8): p. 800-810.
5. Y. Zhu, and N. X. Xiong, A Two-stage Image Segmentation Method Based on Watershed and Fuzzy C-Means. *2008 IEEE Asia-Pacific Services Computing Conference, APSCC, 2008*, P. 1550-1555.

6. N. Li, M. M. Liu and Y. F. Li, Image segmentation algorithm using watershed transform and level set method, *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2007, p.613-616.
7. Jianbo, S. and J. Malik, Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 2000. 22(8): p. 888-905.
8. Rotem, O., H. Greenspan, and J. Goldberger. Combining Region and Edge Cues for Image Segmentation in a Probabilistic Gaussian Mixture Framework. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on. 2007.*
9. Alzate, C. and J. A. K. Suykens. Image Segmentation using a Weighted Kernel PCA Approach to Spectral Clustering. *Computational Intelligence in Image and Signal Processing, 2007. CIISP 2007. IEEE Symposium on. 2007.*
10. Jain, J.M.a.A., Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognit*, 1992. 25: p. 173-188.
11. Comaniciu, D. and P. Meer, Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 2002. 24(5): p. 603-619.
12. Junqing, C., et al., Adaptive perceptual color-texture image segmentation. *Image Processing, IEEE Transactions*, 2005. 14(10): p. 1524-1536.
13. Hofmann, T., J. Puzicha, and J.M. Buhmann. An optimization approach to unsupervised hierarchical texture segmentation. *Image Processing, Proceeding., International Conference on. 1997.*
14. Clausi, D.A. and D. Huang, Design-based texture feature fusion using Gabor filters and co-occurrence probabilities. *Image Processing, IEEE Transactions*, 2005. 14(7): p. 925-936.
15. C.-M. Pun and M.-C. Lee, "Log-Polar Wavelet Energy Signatures for Rotation and Scale Invariant Texture Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 590-603, May 2003.
16. Hsi-Chia, H., Texture segmentation using modulated wavelet transform. *Image Processing, IEEE Transactions*, 2000. 9(7): p. 1299-1302.
17. Vincent, L. and P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 1991. 13(6): p. 583-598.
18. Soille, P., *Morphological Image Analysis: Principles and Applications*, Springer-Verlag, 2003.
19. Vincent, L., Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *Image Processing, IEEE Transactions*, 1993. 2(2): p. 176-201.
20. Ahmed, N., T. Natarajan, and K.R. Rao, Discrete Cosine Transform. *Computers, IEEE Transactions*, 1974. C-23(1): p. 90-93.
21. Strang, G., The Discrete Cosine Transform. *SIAM Review*, 1999. 41: p. 135-147.
22. Khayam, S., The Discrete Cosine Transform (DCT): Theory and Application. *WAVES lab technical report*, May, 2004.
23. C. C. Cheng; J. S. Tsai; Global cutting for the maximum rectangular block from arbitrary closed region. *International Journal of Machine Tools & Manufacture*, 2004; p. 1423-1430.
24. Takahashi, T. Dropping method for rectangle packing problem. in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on. 2000.*
25. D. Martin, C. Fowlkes, D. Tal, and J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in *Proc. 8th Int. Conf. Computer Vision*, vol. 2, pp. 416-423, Jul. 2001.
26. S. D. R. Martin, C. Fowlkes, and J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, *Pattern Analysis and Machine Intelligence, IEEE Transactions*. Jun, 2004, 26(5), p. 530-539.