Bitplane Based Area Morphology for CBIR

K. Kiran Kumar Chakravarthy Bhagvati A. K. Pujari B. L. Deekshatulu
Dept. of Computer and Information Sciences
University of Hyderabad, Hyderabad 500046
{chakcs,akpcs,bldcs}@uohyd.ernet.in

Abstract

Area morphology filters are capable of removing objects in an image based solely on the object area. These operators can be effectively used for content based image retrieval. However, the traditional level set based implementations of these operators are impractical because of their time complexities. Here we present a fast implementation of area morphology based on bitplane decomposition of the image. Our experiments indicate that the results of image segmentation are comparable to the original level set based approach and better than the existing fast area morphological techniques. The bitplane based area morphological segmentation technique is then applied for CBIR. Results on its use in CBIR are presented.

1. Introduction

Successful retrieval of relevant images from large-scale image collections is one of the current problems in the field of data management. Early image retrieval methods locate the desired images by matching keywords that are assigned to each image manually. However, manual processing has become impractical as a result of large numbers of images in collections today. Content based image retrieval (CBIR) has become a useful tool for management of digital image collections. CBIR, which is based on automatically extracted primitive features such as colour, shape, texture and even the spatial relationships among objects, has been employed since early 1990s. In the last decade, a great deal of research work in image processing has concentrated on CBIR technologies.

Area morphology is found to be useful for content based image retrieval. But the high computational costs associated with it make it infeasible for CBIR where we have to apply the technique to a large number of images. Fast algorithms were proposed in the past[1, 7, 9]. Although they decrease the processing times, the quality of results is not comparable to the original, slow implementations. In this paper, we propose a fast implementation of area morphology in bitplanes which speeds up the process while producing results comparable to the original implementation. The method is also applied to content based image retrieval.

The remainder of the paper is organised as follows. In Section 2, we discuss the existing area morphology implementations and their limitations. Section 3 describes our fast implementation and its application to image segmentation. Section 4 deals with application of bitplane based area morphology segmentation to CBIR. We conclude with a brief description of results in Section 5.

2. Area Morphology

Area morphology, proposed originally by Serra[8] and later extensively studied by Salembier, Crespo and Serra[4, 5, 7], is based on two algebraic operations: *area opening (AO)* and *area closing (AC)*. In contrast to the traditional morphological filters, area open and area close filters do not utilise any structuring element. These operations are amorphous, i.e., independent of shape or *shapeless*. We may consider area open operation as the process of removing bright objects that do not meet a specified minimum area. Likewise, the area close operator removes dark objects of insufficient area. The objects are connected components within the level sets of he image. A *level set* is a binary image obtained by thresholding the image at a particular level. For an image I with discrete domain D and image location p we define a level set S_l at level $l \in [0, L-1]$ as

$$S_l(p) = \begin{cases} 1 & \text{if } I(p) \geq l \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

In a level set S_l, the connected component $C_{s_l}(p)$ at p is given by

$$C_{s_l}(p) = \{q : \exists P_{I \geq l}(p, q)\} \qquad (2)$$

where $P_{I \geq l}(p, q)$ is an unbroken path between image locations p and q for which each element obeys $S_l(\cdot) = 1$ (hence

$I(\cdot) \geq l$). The neighbouring elements in such a path can be defined by 4- or 8-connectivity.

For level set S_l, the area open operation is given by

$$S_l \circ (a) = \{p : \exists |C_{s_l}(p)| \geq a\} \quad (3)$$

where a is the minimum area (in terms of the number of pixels). $C_{s_l}(p)$ is the connected component at p. The area of the connected component at p is $|C_{s_l}(p)|$. Area close is similarly defined using the boolean complement of S_l.

For grayscale images, we can define area open and area close by stacking the processed level sets. The reconstructed area-opened image at scale a is given by

$$I \circ (a) = \sum_{l=1}^{L-1} S_l \circ (a) \quad (4)$$

and the area closed image is given by

$$I \bullet (a) = \sum_{l=1}^{L-1} S_l \bullet (a) \quad (5)$$

Sequential application of area open and area closed operations results in useful multiscale operators. *Area open-close (AOC)* and *area close-open (ACO)* operations remove connected components with area less than a from a level set and its complement respectively. Figure 1 shows the "cameraman" image and Figure 2 shows the result after using AOC filter.

Acton's original implementation[3] of the AOC operation is time consuming as connected components have to be computed for all L level sets. It is not practical to apply this implementation to content based image retrieval. Some fast algorithms for area morphology were proposed in [1, 7, 9] but none of them are equivalent of the level set based implementation. The algorithm proposed by Vincent[9] uses standard morphological open operation in preliminary step to create a marker image M from the original image I. The partial connected components within the level sets of marker image M are fully reconstructed to yield the final reconstruction R.

The pyramidal algorithm proposed by Acton[1] is based on the notion that we can successively eliminate small regions in the image by creating coarser pyramid levels. Then, the surviving regions of insufficient area can be reconstructed in a coarse-to-fine manner. To create a marker image for an area open with area $a = 4^m$, we first create the $m + 1$ level erosion pyramid. The marker creation operates in two steps as described below.

- *Analysis Step*: Let $P_0 = I$. For level $m > 0$

$$P_m = (P_{m-1} \ominus F_0) \downarrow$$

Figure 1. Original Cameraman Image (128×128 pixels with 256 grayscales)

Figure 2. Result of AOC filter operation on Figure 1 using level set implementation

where F_0 is the 2×2 structuring element with the origin in the upper-left. The downsampling operation, denoted by $\downarrow$, is the injection operator, where the upper-left value of each 2×2 image subsection is sampled to create an image that is half as wide and half as high as that of the previous pyramid level. It is important to note that at level m, the value of the pixels will represent the highest level set for which a connected component of size $2^m \times 2^m$(area a) exists in the original image.

- *Synthesis Step*: We use the following relationship, starting at level $m + 1$, to recreate the connected components

$$P_m = [(P_{m+1}) \uparrow] \oplus F_0$$

where the upsampling operator $\uparrow$ simply injects the pyramid values in a matrix of zeros that is twice as wide and high. The dilation step serves as prolongation operation.

The image P_0 is used as the marker M after synthesis.

The partial connected components in the marker image M are reconstructed fully based on the intensities in the input image I. The components are reconstructed by selectively dilating these components (one pixel at a time) using a 3×3 cross-shaped structuring element with the origin at the centre of the cross. The result of applying pyramidal algorithm with area $a = 4$ on Figure 1 is shown in Figure 3. This method was found to give provably superior results among the existing fast algorithms for area morphology in terms of computational complexity[2].

Figure 3. Result of AOC operation using the fast pyramid algorithm on Figure 1

Although the fast algorithms improve substantially upon the computational cost, the results are not equivalent to the original level set based implementation of area open operation. The impact of the 3×3 cross-shaped structuring element is clearly visible as blockiness in the results of area open and area close operations. More importantly, the primary motivation for area morphology (i.e., removal of structuring element and its impact) is virtually ignored in these fast implementations.

3. Bitplane Based Area Morphological Segmentation

In this paper, we propose a novel fast algorithm that does not make use of any structuring element by decomposing the input image into bitplanes and then reconstructing the same from the bitplanes. Our approach, we argue, is truer to the philosophy of area morphology and show that it is about 30 times faster than the level set implementation for a 256 graylevel image. Our results are also more similar to the original implementation than those of [1, 2].

3.1. Area Operators

Here we use the concept of decomposing the image into bitplanes. Bitplane decomposition results in splitting a multilevel (monochrome or colour) image into a series of $m = \log_2 L$ binary images. The graylevels of an $m-$bit grayscale image can be represented in the form of a base-2 polynomial

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \ldots + a_1 2^1 + a_0 2^0 \quad (6)$$

A simple method of decomposing an image into a collection of binary images is to separate the m coefficients of the above polynomial. The 0^{th}-order bitplane is generated by collecting the a_0 bits of each pixel, while the $(m-1)^{st}$-order bitplane consists of the a_{m-1} bits or coefficients. In general, each bitplane is numbered from 0 to $m-1$ and is constructed by setting its pixels equal to the values of the appropriate bits or polynomial coefficients from each pixel in the original image.

Given the bitplane definitions, we can define area open and area close operations using the bitplanes. First, we define the operations on individual bitplanes and then for a grayscale image.

For a bitplane B_l, area open can be defined as

$$B_l \circ (a) = \{p : \exists |C_{B_l}(p)| \geq a\} \quad (7)$$

where a is the minimum area in terms of the number of pixels. The area of the connected component at p is $|C_{B_l}(p)|$ and this area may be 0. An area opened bitplane does not contain connected components with areas smaller than the minimum area a.

Similarly, an area close operation on a bitplane is given by

$$B_l \bullet (a) = \{p : \exists |C_{\overline{B_l}}(p)| \geq a\} \quad (8)$$

where the cardinality of the connected component is defined on the complement of bitplane B_l, i.e., where $B_l(p) = 0$. The area closed bitplane, thus, does not contain any connected components of 0s that are smaller in area than the minimum area a.

For grayscale images, we can define area open and area close by reconstructing the processed bitplanes. The reconstructed area-opened image at scale a is given by

$$I \circ (a) = \sum_{k=0}^{m-1} [B_k \circ (a)] 2^k \quad (9)$$

The area-closed image is given by

$$I \bullet (a) = \sum_{k=0}^{m-1} [B_k \bullet (a)] 2^k \quad (10)$$

The result of applying bitplane based AOC operation on Figure 1 is shown in Figure 4. Table 1 lists the time complexities and the actual average running times (in s) on a set of 128×128 pixel 256 grayscale test images. The bitplane based implementation is significantly faster than the original level set implementation but slower than the pyramid algorithm proposed in [1]. However, the resulting image is “cleaner” and does not show the blockiness evident in Figure 3. The same is substantiated by comparing the mean-square error between the level set implementation,

	Level set based implementation	Pyramidal algorithm	Bitplane based implementation
Time complexity	$O(LN)$	$O(N)$	$O(N \log_2 L)$
Time (in s) for 128×128 image	98s	0.5s	6s

Table 1. Timings and complexities of area open operations using different algorithms. L is the number of grayscales in the image.

and pyramid and bitplane algorithms respectively. For example, on the "cameraman" image in Figure 1, the mean-square error for the pyramid algorithm is 7630 while it is 153 for the bitplane algorithm. We found that the mean square errors are a factor of 30–50 times lower for the bitplane algorithm in our testing.

Figure 4. Result of AOC operation using our bitplane based algorithm on Figure 1

3.2. Scale-Space Creation

As with the case of level set implementation. we can create a scale space by sequential application of the area open and area close operations resulting in useful bias-reduced multiscale operators. Area open close (AOC) and area close open (ACO) are written $I \circ (\bullet(a))$ and $I \bullet (\circ(a))$ respectively. Both operations together remove bright and dark objects with area less than a. AOC and ACO produce image sacale spaces as the area parameter a is increased. AOC and ACO scale-spaces were found to have the following properties[3].

- *Fidelity*: This property ensures that the finest scale of scale-space contains the original input signal. Here $I_0 = I$ as the initial condition while creating the subspace will guarantee this property.
- *Euclidean Invariance*: This property ensures that the method of scale-space creation is invariant to translation and rotation of the image.
- *Causality*: It infers that a coarse scale representation can be recreated from any finer scale representation. Scale space produced by bitplane method is causal as I_s depends only on I_r for $s \geq r,\ r \geq 0$ where r, s are different scales in scale-space.

3.3. k-Means Clustering

The pixels within the scale-space corresponding to the same image locations form a scale-space vector. A scale-space vector therefore contains the intensity of a particular pixel for a given set of scales. The objective of scale-space classification is to group the scale-space vectors based on some similarity measure. The k-means clustering algorithm is a hard classifier which classifies the vectors based on the distance from the vector to the mean vector. It tries to minimise the squared error function

$$e^2 = \sum_s \sum_{i=1}^{k} ||d_i(x,y)||^2 \tag{11}$$

where s represents the number of scales in scale-space and k represents the number of clusters.

An initial seed vector representing cluster centres is arbitrarily specified for each class. The scale-space vectors are classified based on the minimum Euclidean distance from the cluster centres. A new set of cluster centres is then calculated from this reassignment. The iterative process continues until the net migration of cluster centres is insignificant.

The segmented images obtained with 3 clusters ($k = 3$) by the level set implementation as well as bitpane based implementation are shown in Figures 5 and 6 respectively. It may be seen that the bitplane based method introduces spurious segments in some, especially uniform, areas of the image. This over-segmentation is a result of reducing the number of level sets and thus losing the dynamic resolution. However, we found that the segmentation is of a better quality than that obtained when using the pyramid approximation.

4. Content-Based Image Retrieval

The segmented image obtained by bitplane based area morphology is used to extract and match image features in

Figure 5. Segmentation of Figure 1 with $k = 3$ using the level set AOC implementation

Figure 6. Segmentation of Figure 1 with $k = 3$ using our bitplane AOC implementation

a CBIR application on a Brodatz texture database of more than 3000 images. We are in the process of testing the same features on another database containing 10,000 images.

4.1. Feature Extraction

The segmentation obtained from clustering the scale-space vectors is used for the purpose of extracting local features at a segment level. These features are primarily for texture and include the mean, variance, uniformity, skewness and entropy defined below. In the following equations z is a random variable that represents the gray levels and $p(z_i)$ is the corresponding histogram.

- Mean, $\mu = \sum_{i=0}^{L-1} z_i p(z_i)$
- Variance, $\sigma^2 = \sum_{i=0}^{L-1} (z_i - \mu)^2 p(z_i)$
- Skewness, $\mu_3 = \sum_{i=0}^{L-1} (z_i - \mu)^3 p(z_i)$
- Uniformity, $U = \sum_{i=0}^{L-1} p_i^2(z_i)$
- Entropy, $E = -\sum_{i=0}^{L-1} p(z_i) \log(p(z_i))$

4.2. Matching

Integrated Region Matching (IRM)[6] is a novel similarity measure of images based on region segmentation and segment-level properties. We represent the image by a set of regions, roughly corresponding to the objects in the image, and a set of colour, texture, shape and location features are computed for each such region. The IRM measure for evaluating the overall similarity between images incorporates properties of all the regions in the images by a region-matching scheme. Compared with retrieval based on individual regions, the overall similarity approach reduces the influence of incaccurate segmentation and helps to clarify the sematics of a particular region. Each region is assigned a significance measure, indicating its importance in the matching for deciding the overall similarity between images.

In our experiments, we chose the texture features listed above as region features, and the area percentage (i.e., the fraction of the area of the entire image that is occupied by a particular region) as the significance measure. The intuition is that important or perceptually significant areas generally occupy large areas in an image.

5. Results

We implemented the level set based, pyramid and our bitplane based area morphological operations as well as the k-means clustering algorithm in C language using gcc compiler on a computer with a Pentium 4 processor at 1.6 GHz and 128 MB RAM. Segmenting the 128×128 pixel cameraman image shown in Figure 1 took nearly 40 minutes using the level set based implementation. It took only 2 minutes using our bitplane approach and about 12 seconds using the pyramid algorithm.

A Brodatz database containing more than 3000 images is used for testing our area morphological operation on a CBIR application. The database contains about 15 variants for each of the major texture classes. In CBIR, it is expected that these 15 images be retrieved when queried with an image from the database.

Initially, a database of all the images is created by running the AOC operation followed by k-means clustering to create a segmented image from which texture features (Section 4.1) were computed. Query processing and similarity measure are computed online and the best 10 database images are retrieved.

An example query and results are shown in Figure 7. The query image is shown in Figure 7(a). The image is already present in the database and therefore the best image retrieved from the database is the query image itself (Figure 7(b)). The other images retrieved, in the order of decreasing similarity measure, are shown in Figure 7(c)–(f). Of the 5 images shown in the Figure, four belong to the same category as the query image and should be regarded as *correct*. The other image (Figure 7(d)) is from a different category and may be construed as *incorrect*. In general, our experiments indicate that the retrieval accuracy is about 80% for the Brodatz database.

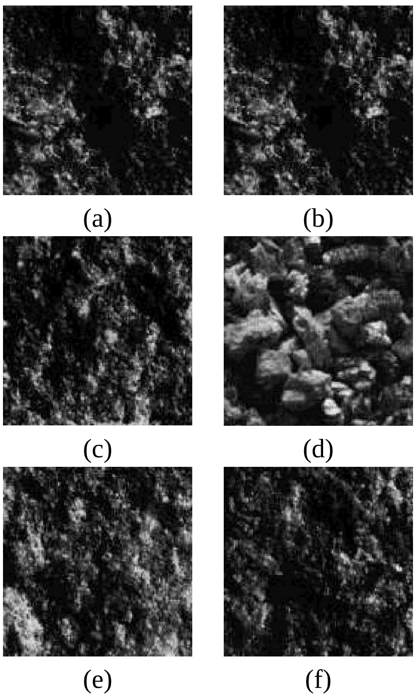

Figure 7. Example query and results. (a) Query image, (b)–(f) Top 5 matches

6. Conclusions

This paper described a novel bitplane based approach to area morphology. Our new implementation of the AOC and ACO algorithms run about 20 times faster than the original level set implementations. We found that the quality of results obtained by our approach is superior to that of existing fast algorithms as seen in a mean squared error that is lower by a factor of almost 30–50. The bitplane AOC and ACO operators do not introduce any spurious edges across multiple scales when used in creating scale-spaces. They also satisfy the three properties of scale-spaces (Section 3.2) that the original level set implementations do.

We used the new algorithms in a multiscale segmentation algorithm and showed that they result in small amounts of over-segmentation primarily due to reduced grayscale resolution. However, this result improves the quality obtained from the existing pyramid algorithm. The segmentation technique used is rotation invariant (due to the properties of area morphological operations).

The increased speed and higher degree of fidelity made it feasible to apply the bitplane area morphological operations in a CBIR application. Combined with IRM similarity measure, the results show that the performance is good and comparable to that of other CBIR approaches.

References

[1] Scott T. Acton. A Pyramidal Algorithm for Area Morphology. In *Proc. IEEE Int. Conf. on Image Processing*, Vancouver, Canada, 2000.

[2] Scott T. Acton. Fast Algorithms for Area Morphology. In *Digital Signal Processing II*. 2001.

[3] Scott T. Acton and D. P. Mukherjee. Scale Space Classification Using Area Morphology. *IEEE Trans. on Image Processing*, **9**(4):623–635, 2000.

[4] J. Crespo, R. Schafer, J. Serra, C. Gratin, and F. Meyer. The Flat Zone Approach: A General Low-level Region Merging Segmentation Method. *Signal Processing*, **62**:37–60, 1997.

[5] J. Crespo, J. Serra, and R. Schafer. Theoretical Aspects of Morphological Filters by Reconstruction. *Signal Processing*, **47**:201–225, 1995.

[6] Jia Li, J. Z. Wang, and Gio Wiederhold. IRM: Integrated Region Matching for Image Retrieval. In *Proc. of 8th ACM Int. Conf. on Multimedia*, pages 147–156, Los Angeles, California, 2000.

[7] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, 1998.

[8] J. Serra and P. Salembier. Connected Operators and Pyramids. In *Proc. of SPIE: Non-linear Algebra and Morphological Image Processing*, volume 2030, San Diego, 1993.

[9] L. Vincent. Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. *IEEE Trans. on Image Processing*, **2**(2), 1993.

ACKNOWLEDGMENTS: The authors wish to acknowledge the Department of Space, Govt of India for supporting the research (RESPOND Grant No.10/4/432).