Application for Morphological Image Processing

Author:
Dan Campbell
UW-Madison
Computer Engineering/Computer Science

Course:
ECE533

Date Submitted:
12/21/2006
I. Introduction

It is well known that there are many different learning styles. Some people learn better by reading books, others through a verbal explanation, while others learn most effectively through application. The goal of this project is to add another tool to the learning style, one focused on a visual learning style. By developing an application to demonstrate some tools of morphological image processing, the goal is to add another tool to the learning processes.

II. Background

Morphological image processing relies on the ordering of pixels in an image and many times is applied to binary and grayscale images. Through processes such as erosion, dilation, opening and closing, binary images can be modified to the user’s specifications.

Dilation/Erosion

First, define A as the reference image and B is the structure image used to process A. Dilation is defined by the equation:

$$ A ⊕ B = \{ z \mid [(\hat{B})_z \cap A] \subseteq A \} $$

Where $\hat{B}$ is B rotated about the origin. Dilation has many uses but a major one is bridging gaps in an image due to the fact that B is expanding the features of A.

Dilation on the other hand can be considered a narrowing of features on an image. Again defining A as the reference image and B as the structure image:

$$ A \Theta B = \{ z \mid (B)_z \subseteq A \} $$

Many times dilation can be used for removing irrelevant data from an image.

![Figure 1. Erosion(b) and dilation(c) of image (a) Source: [2]](image)
**Opening/Closing**

By utilizing the processes of erosion and dilation, opening and closing is simply and extension of these applications. The process of “opening” an image will likely smooth the edges, break narrow block connectors and remove small protrusions from a reference image. “Closing” an image will also smooth edges but will fuse narrow blocks and fill in holes.

Opening:

\[ A \ast B = (A \ominus B) \oplus B \]

Closing:

\[ A \ast B = (A \oplus B) \Theta B \]

By these definitions, the opening of A is the erosion of A by B and then that image dilated by B. The closing of A is the dilation of A by B and then eroded by B.

![Figure 2. Opening(d) and Closing(e) of image (a) Source: [2]](image)

By knowing that dilation and erosion are duals of each other:

\[ (A \Theta B)^c = A^c \oplus \hat{B} \]

we can conclude that with respect to set complementation and reflection, that opening and closing are complements of each other:

\[ (A \ast B)^c = A^c \circ \hat{B}. \]
III. Application

The application developed allows the user to perform four main operations to an image: dilation, erosion, opening and closing. Listed below are a few of the functionalities of the program:

**Visual inspection of image processing** allows the user to see how the structure image affects the original image.

**Variable playback speeds** allows the user to control the speed at which the structure image is processed through the image so a user can see how it affects the final image.

**User defined structure image** lets the user control what the 3x3 structure image looks like and allows users the ability to see how different structure images affect different images.

**User defined images** lets the user define an image up to 16x16. By clicking on the different cells, a user can setup an image to their specifications before processing.

**Rewind functionality** enables a user to revert back to the original image if multiple passes were made during image processing (such as during opening and closing).

**Application Screens**

*Dilation/Erosion*

![Figure 3. Screen seen by user when application is launched.](image-url)
Figure 4. Image and structure images have been defined and are ready for dilation processing.

Figure 5. Image partially processed by dilation algorithm.
Figure 6. Image fully processed by dilation algorithm.

Opening/Closing

Figure 7. Image and structure images have been defined and are ready for opening processing.
Image 8. Image after erosion and ready for dilation during opening algorithm.

IV. Known Bugs

Major Bugs

(1) Due to the fact that this program is single threaded and relies on a timer ticker to control the playback of the moving structure matrix, it is impossible to visually process the first pass during opening and closing. VB.Net does not have a built in pause until even function so the only way to accomplish this would be to multithread the application or place a semaphore around blocks of code. Due to an inexperience in multithreaded applications, there wasn’t enough time to accomplish this.

Minor Bugs

(1) If the ‘Play’ button is pressed while the image is being processed, the final image clears from the point that it has already been processed.
(2) Top corner 2x2 matrix sometimes disappears on the reference image until ‘Rewind’ or ‘Fast Forward’ is pushed.
V. Future Work

Due to the fact that this is a teaching tool, it was the goal for this project to make it easy to add to. Here are the items that could be added for the future quite easily:

(1) Image upload and binarization using segmentation, then allowing morphological image processing on the uploaded image
(2) Boundary extraction by subtracting the original image from the eroded image
(3) Hit or miss transformation could be implemented easily with the already included erosion processing
(4) With hit or miss, thinning and thickening could easily be added to the application

VI. References