

CS 376A
Digital Image Processing

02 / 06 / 2023

Instructor: Michael Eckmann

Today's Topics

- Questions / Comments?
- Binary Image Morphology
 - Structuring element
 - Hits, Fits
 - Erode, Dilate
 - Close, Open
 - And, Or, Not, Minus
 - Lossless image format (ppm)

Morphological operators

- Motivation for these operators - often in an image (possibly after processing in some way) we have small gaps that we would like to fill in or jagged edges that we would like to remove.
- We will consider strict black and white (binary) images first and talk later about expanding to other kinds of images.
- Note: for single bit per pixel images, white pixels have value 1, black have value 0.
- Examples of morphological processing.

structuring element

- a structuring element is a typically small binary image representing some shape. A structuring element has an origin.
 - e.g. a 3x3 square of all 1-pixels, with center as origin
 - a 5x5 binary image with a diamond shape of 1-pixels, others 0's
 - let's me draw some on the board
- a structuring element is applied to a binary image by hovering the origin pixel over each pixel in the image one at a time
 - only the 1-pixels in the structuring element matter (the 0-pixels are not compared)
 - **fits** means that *all* the 1-pixels in the structuring element correspond to 1-pixels in the image
 - **hits** means that *at least one* of the 1-pixels in the structuring element corresponds to a 1-pixel in the image

binary morphology

- binary image morphological operations
 - image is f , structuring element is s , resulting image is g
 - dilation – increases the size of regions
 - f dilated by s results in g where g contains a 1 where s hits f , 0 elsewhere
 - erosion – decreases the size of regions
 - f eroded by s results in g where g contains a 1 where s fits f , 0 elsewhere
 - Let's perform dilation and erosion on an image with a specific structuring element on the board.

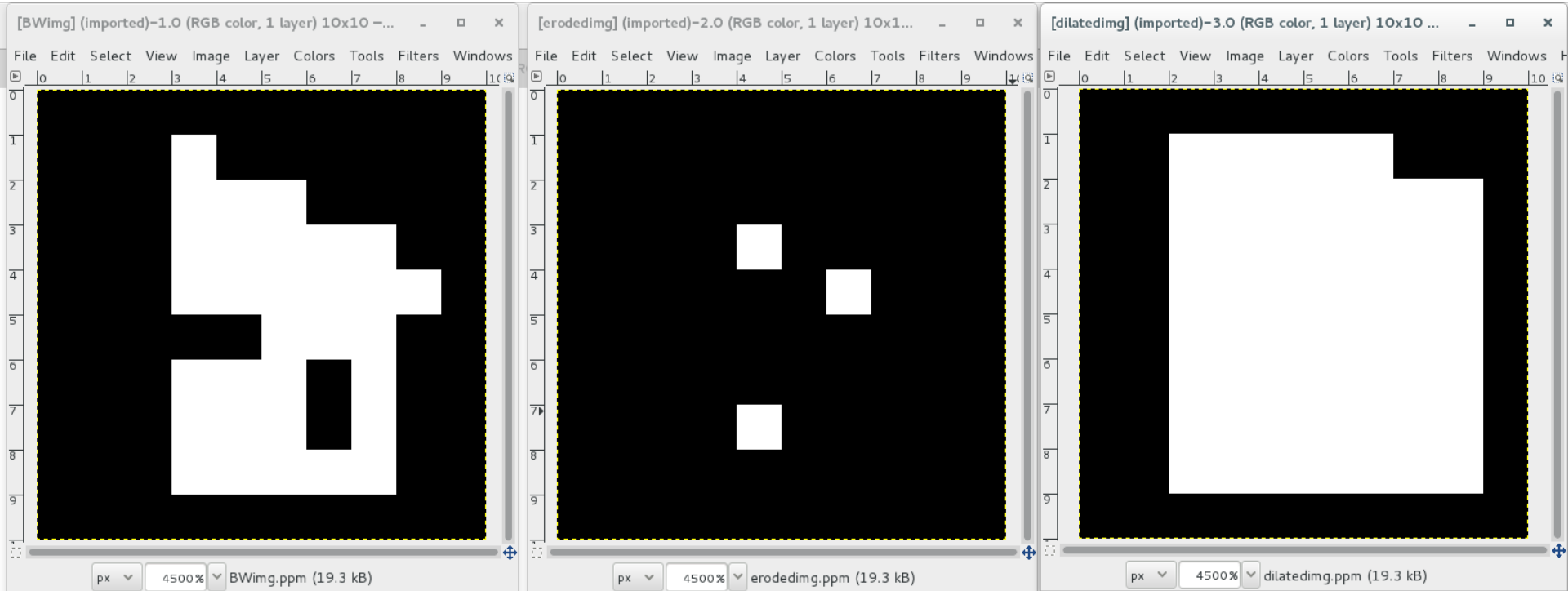
binary morphology

- binary image morphological operations
 - image is f , structuring element is s , resulting image is g
 - closing – closes up holes within regions
 - f closed by s results in g , which is the result of f dilated by s and then eroded by s
 - opening – get rid of jutting out portions of regions
 - f opened by s results in g , which is the result of f eroded by s and then dilated by s
- Both opening and closing are idempotent
 - this means after one closing of f by s , further closings by s do not change the result
 - same for opening

binary morphology

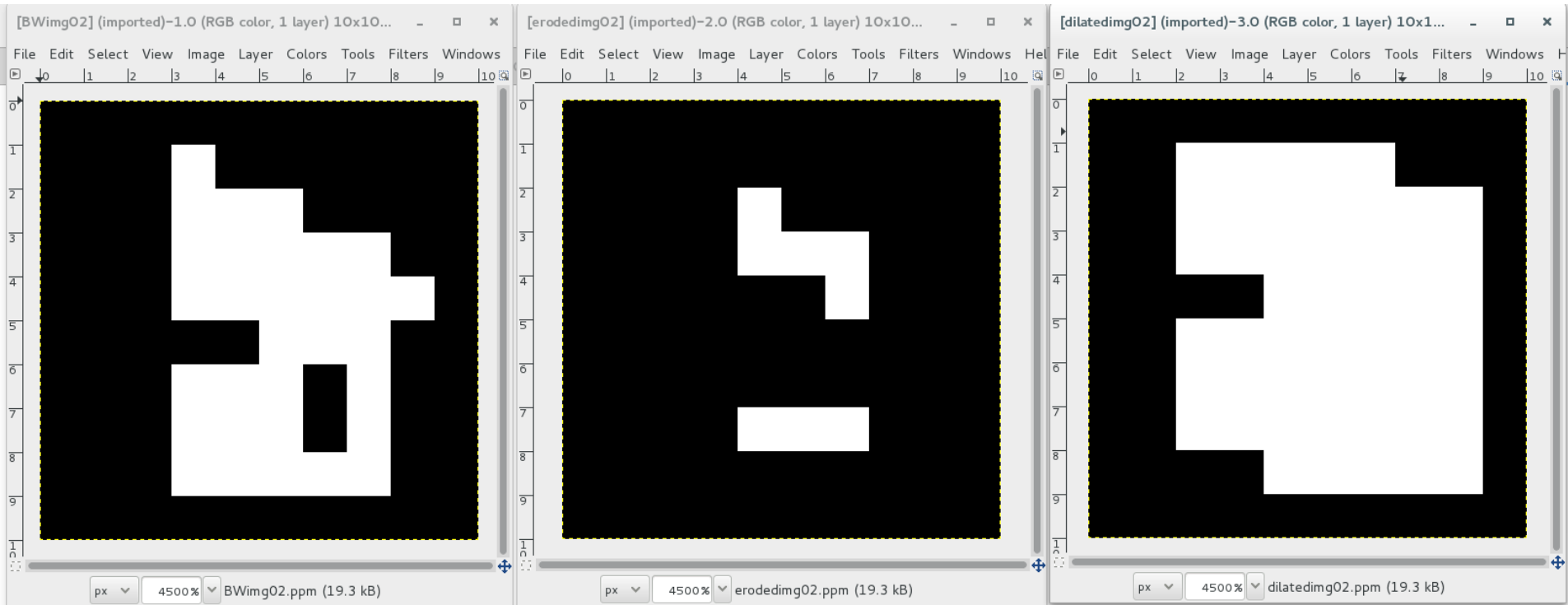
- Other simple operations that can be done to binary images
 - f and g are binary images of the same dimensions
 - AND: f AND g results in h where h contains a 1 iff f and g both contain a 1
 - OR: f OR g results in h where h contains a 1 if either f and/or g contain a 1
 - NOT: NOT f results in h where all 0's in f result in 1's in h and all 1's in f result in 0's in h
 - MINUS: f MINUS g results in h where
 - 1 minus 0 = 1
 - all others 0

examples



- Left image – original 10x10 w/ white starting in 2nd row, 4th column,
- Middle image – eroded w/ 3x3 Box,
- Right image – dilated w/ 3x3 Box.

examples



- Left image – original 10x10 w/ white starting in 2nd row, 4th column,
- Middle image – eroded w/ 3x3 L shape,
- Right image – dilated w/ 3x3 L shape.

binary morphology

- Jpeg is lossy so I chose to use a lossless easy to create with text files image format, which is very space inefficient --- ppm image file format.
- P3
- width height
- maxVal
- followed by width*height*3 numbers representing the RGB values of the pixels from top left to bottom right
- See: <http://netpbm.sourceforge.net/doc/ppm.html>

binary morphology

- I wrote code to
 - read ppm files into an RGBImage
 - write ppm files from an RGBImage
- For binary morphology I wrote code to
 - dilate
 - erode
- an RGBImage (that is assumed to only have 0,0,0 and 255,255,255 pixels) by a 2d structuring element (of booleans)
- Let's try a few structuring elements and images
- Also, let's write open and close and try those.