

CS 331
Computer Vision

09 / 17 / 2024

Instructor: Michael Eckmann

Today's Topics

- Questions / Comments?
- Connected Components algorithm
- Some examples of opening, closing and border using Binary Image Morphology
- Another example of use of Binary Image Morphology

Connected component problem

- Problem:
 - Give a label to each connected component of an image (consider black/white images only for now where black is background, white is foreground). A connected component is a set of connected foreground pixels.

Connected components

- We can consider either neighbors as 4-adjacent or 8-adjacent
 - 4-adjacent means a pixel is connected if it is adjacent to another pixel in the region either up, down, left or right
 - 8-adjacent means a pixel is connected if it is adjacent to another pixel in the region either up, down, left, right or diagonally up/left, up/right, down/left, down/right

Connected components

- The following connected component labelling algorithm uses 4 adjacency.
- It uses a union-find data structure which has these operations
 - Union operation takes two labels and the union-find data structure and joins the two sets that have each of those labels into one set.
 - Find operation takes one label and the union-find data structure and finds the set containing that label and returns the smallest number label in that set.

Connected components

- One way this can be accomplished is by implementing this union-find data structure as an array that represents a forest of trees.
- Each tree in the forest represents an equivalence class (the labels in the nodes form a set that may be different labels for the SAME connected component.)
- The indices of the array are the labels. An element in the array is the parent label of the index label.
- Example on the board of what is stored in an array for a forest of trees.
- And pseudocode for union and find.

Connected components

- Also need an operation to just add a label as a tree of one node.
- Pseudocode for union and find.
 - Union
 - parameters: labelOne, labelTwo, parentArray
 - loop until find root of labelOne
 - loop until find root of labelTwo
 - If those two roots are not the same, make one the parent of the other.
 - Find
 - parameters: label, parentArray
 - loop until find root of label
 - return root

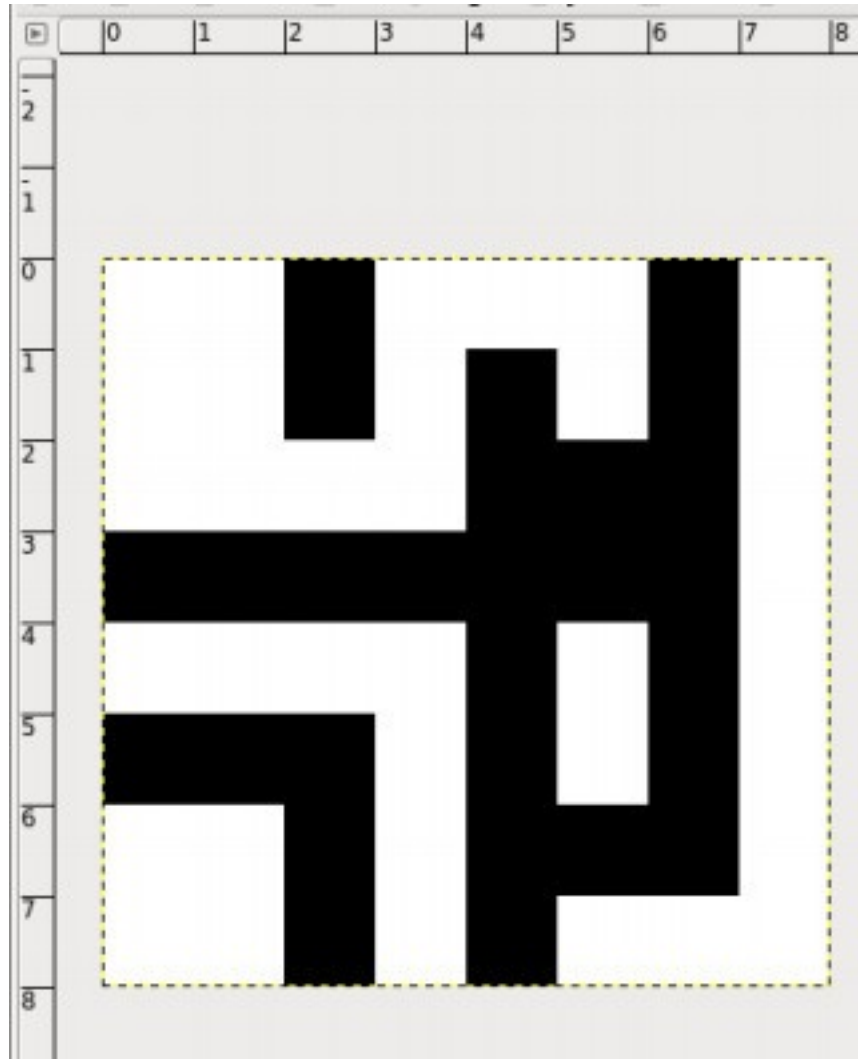
Connected components

- PASS 1: travel each row from left to right
 - if see a foreground pixel, examine label of pixel above and to left
 - if only one is labeled, then label the current pixel that label
 - if both are labeled, then label the current pixel the lesser of the two labels
 - if neither above nor left pixel has labels (i.e. they are background pixels), then pick a new label for current pixel
 - if above and left pixel have different labels, enter the equivalence class into the union-find data structure (perform a union of the two sets)
- After PASS 1, all foreground pixels have been labeled and we have a set of equivalence classes for all labels

Connected components

- PASS 2:
 - relabel the labels of the pixels with the root label of the equivalence class in the union-find data structure (using the find algorithm to determine this root label)
- Example run of this algorithm using image on next slide.

Connected components



Michael Eckmann - Skidmore
College - CS 331 - Fall 2024

Binary Morphology program

- Let me demonstrate binary morphology on a binary image of Abraham Lincoln.
- I have 3 versions of this image – one that is ideal, one that has juttred out pixels and one that is noisy.
- I have implemented dilate, erode, open, close and others
- Let's open the one that has juttred out pixels (with a 10x10 structuring element)
- Let's close the one that is noisy (with a 10x10 structuring element)

Binary Morphology program

- Recall that we know how to do dilate, erode, open, close, and, or, not, and minus operations. What combination of those do you think might produce a border?

Binary Morphology program

- Recall that we know how to do dilate, erode, open, close, and, or, not, and minus operations. What combination of those do you think might produce a border?
- I implemented it as erode followed by minus

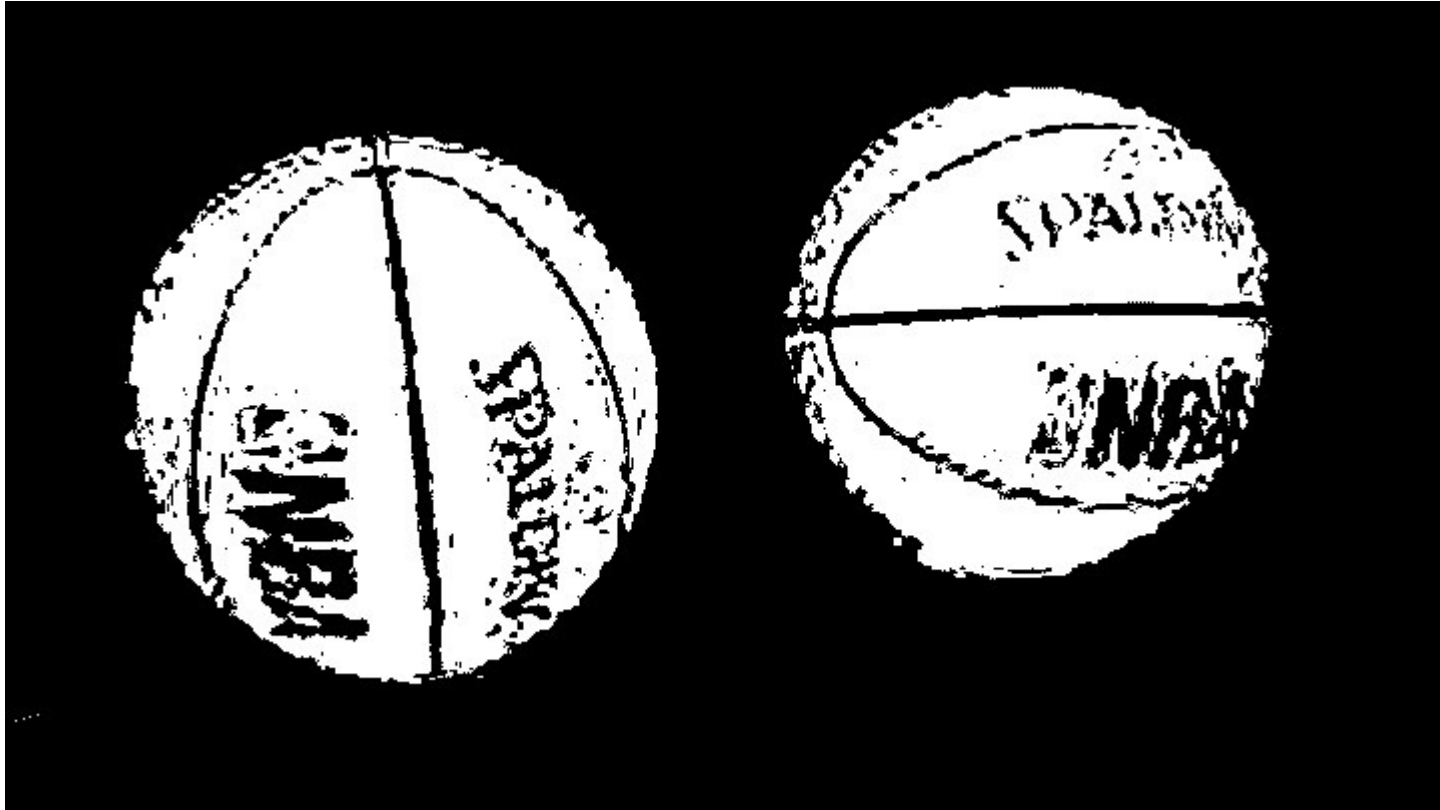
Binary Morphology program

- Let me show you that after opening once, future openings do not have an effect
- Same with closing
- Let's see if opening and then closing results in the same image as closing and then opening which was brought up last class.

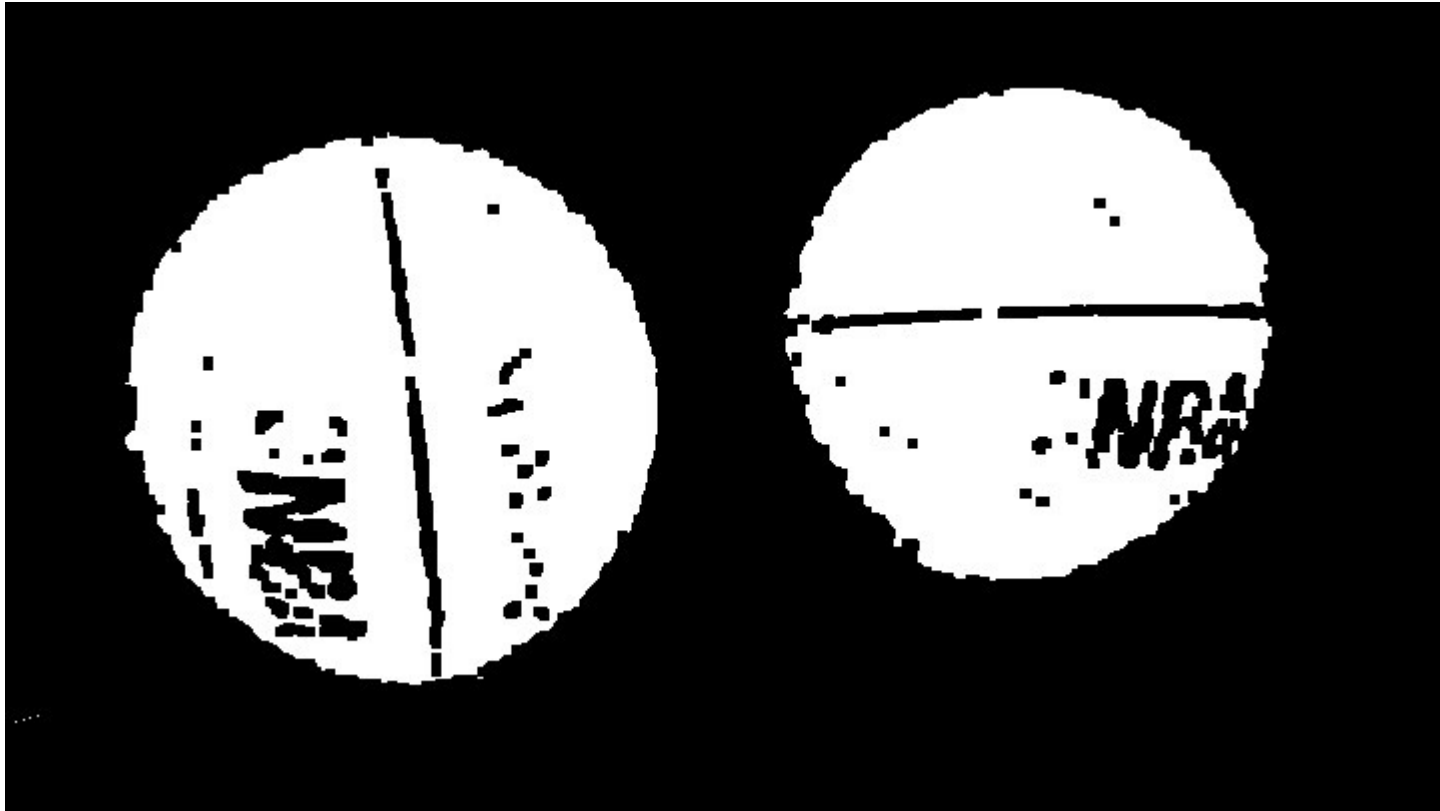
- An image of basketballs



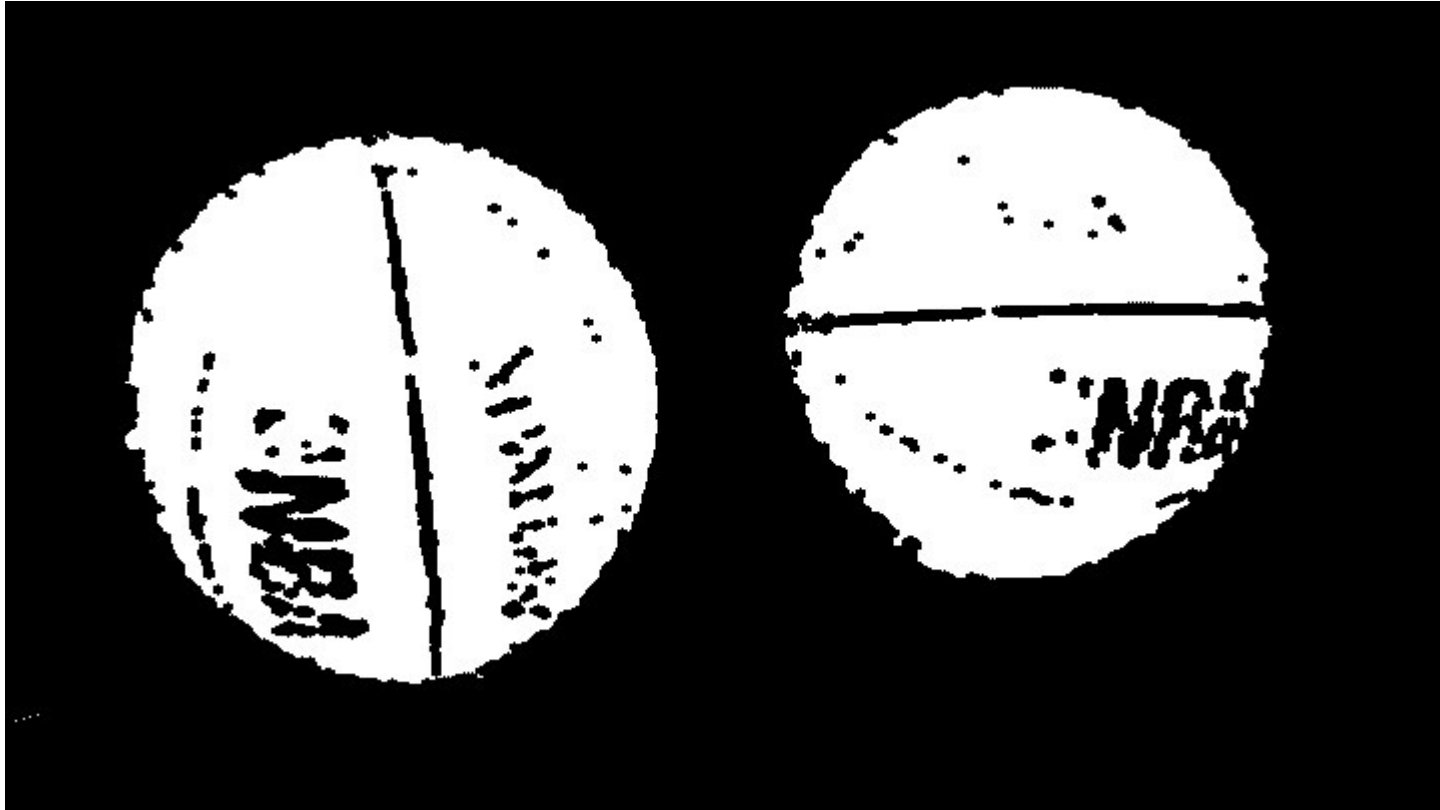
- A mask created by selecting all the basketball color hues (I converted to HSV and selected hue angles 0-20 (on the 0-359 scale))



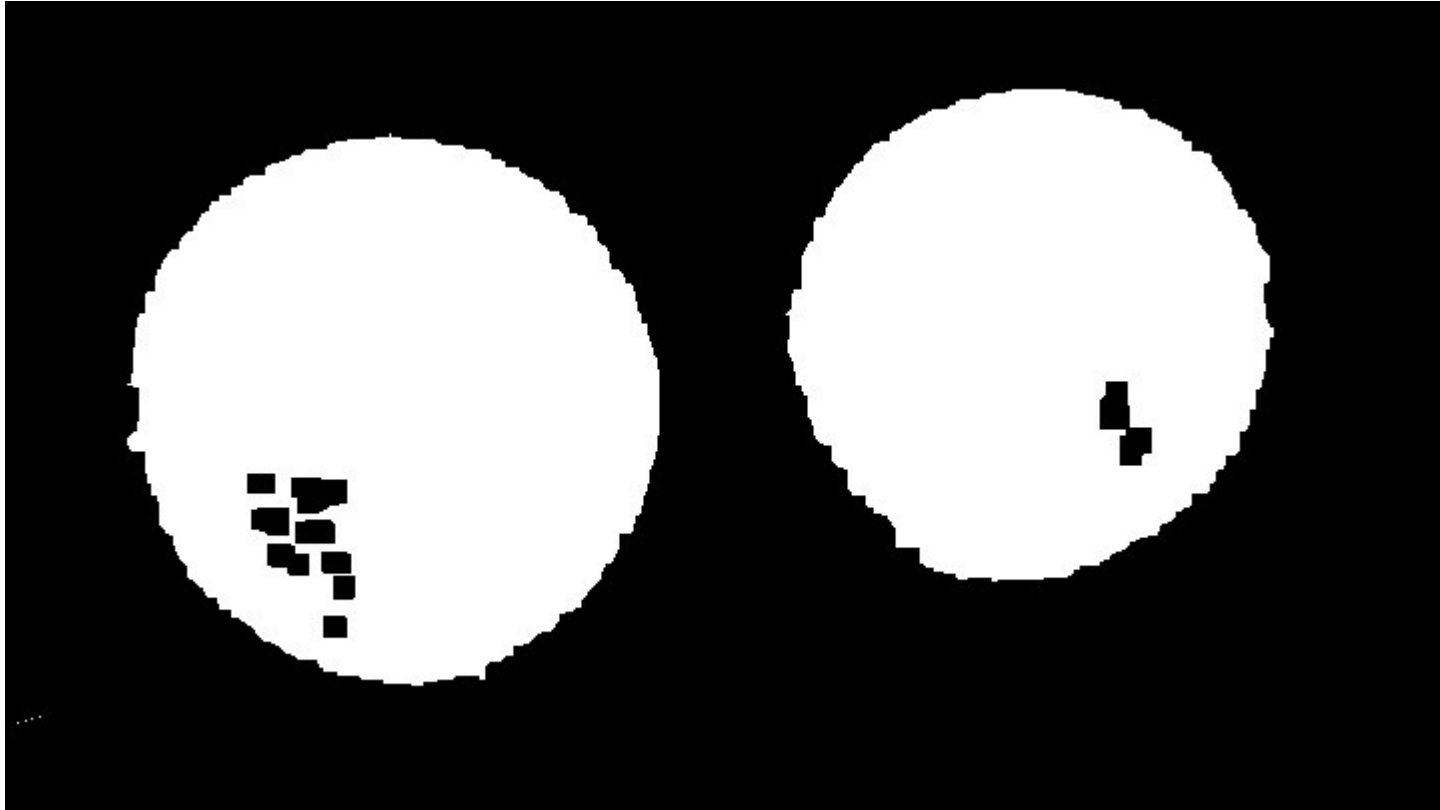
- The mask after closing with a 5x5 box structuring element



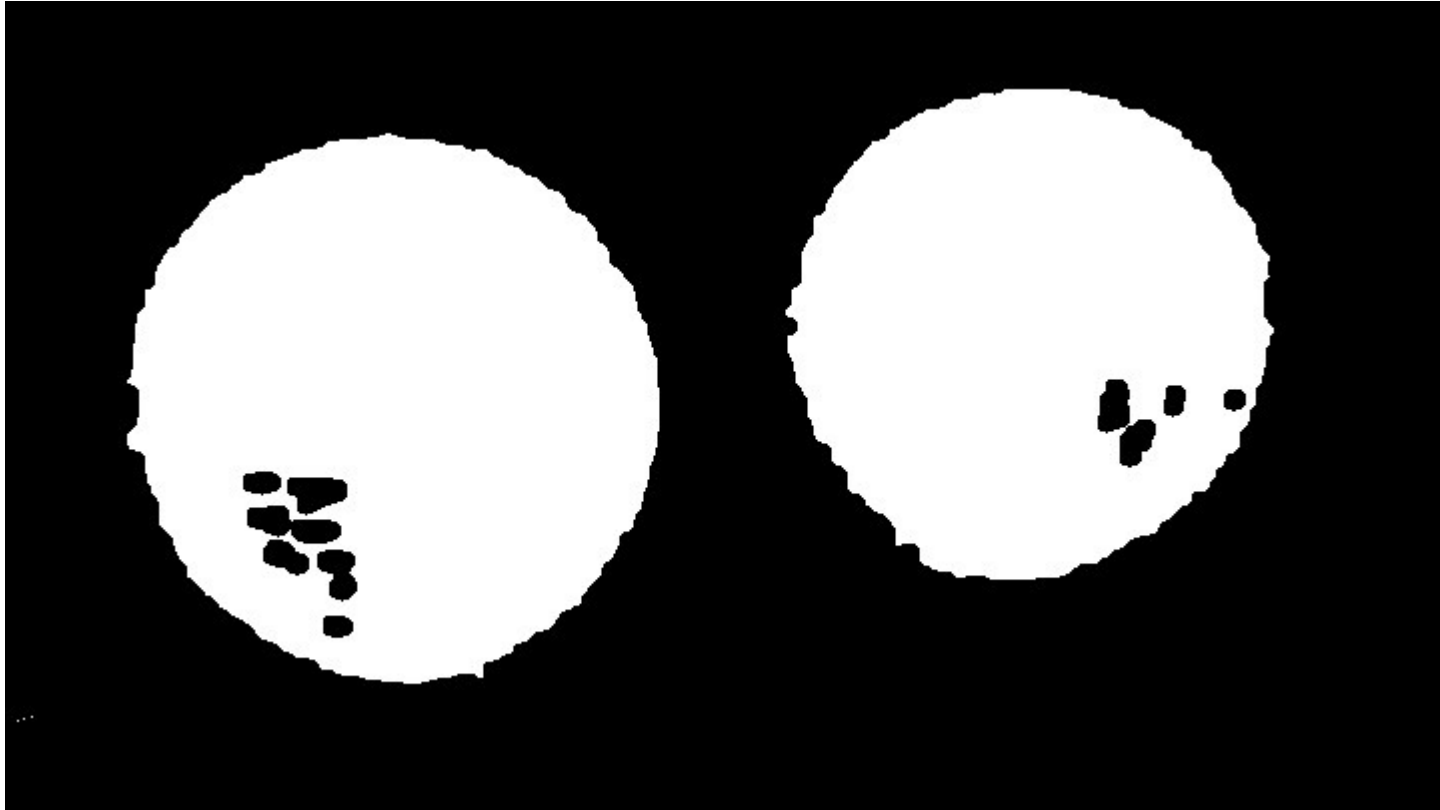
- The mask after closing with a 5x5 circle structuring element



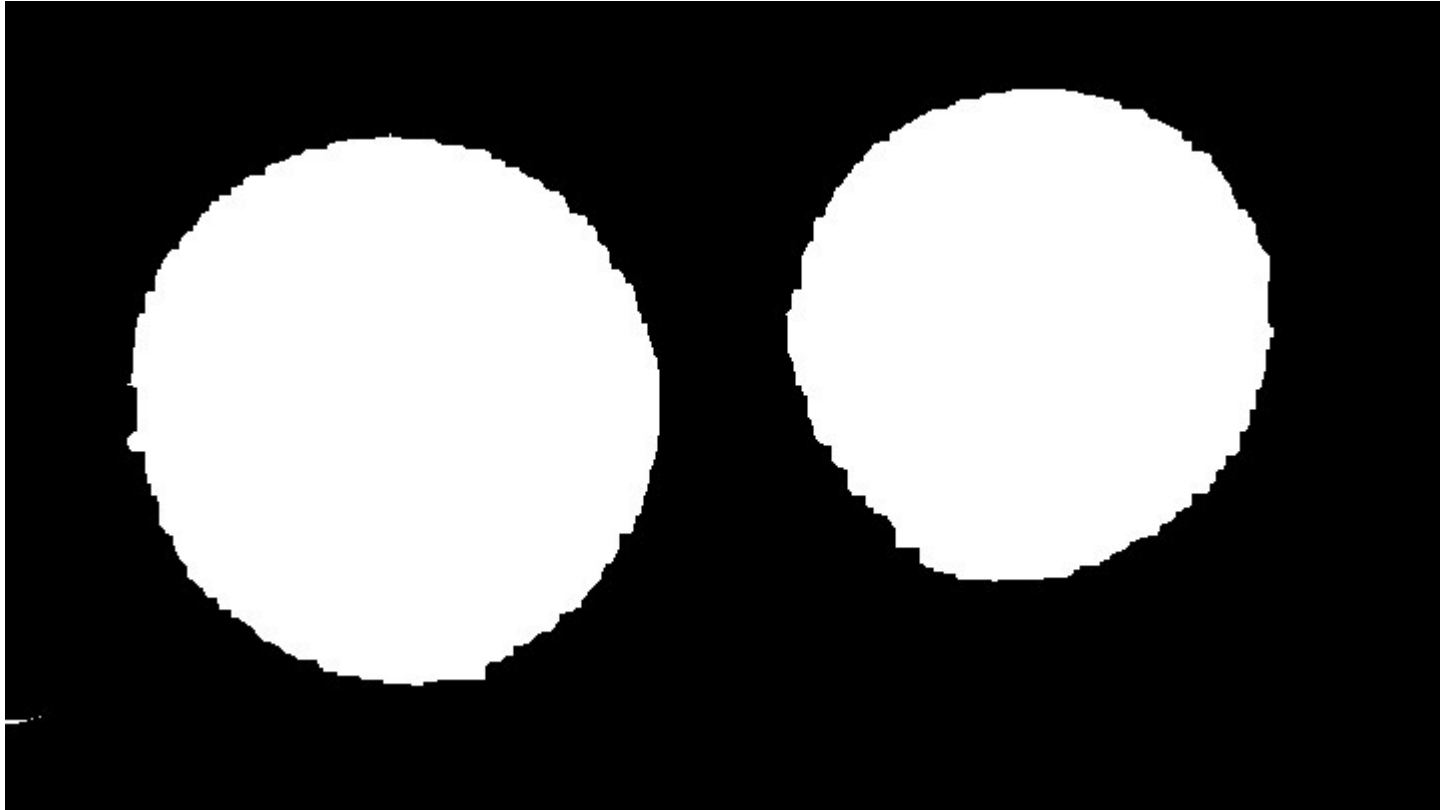
- The mask after closing with a 10x10 box structuring element



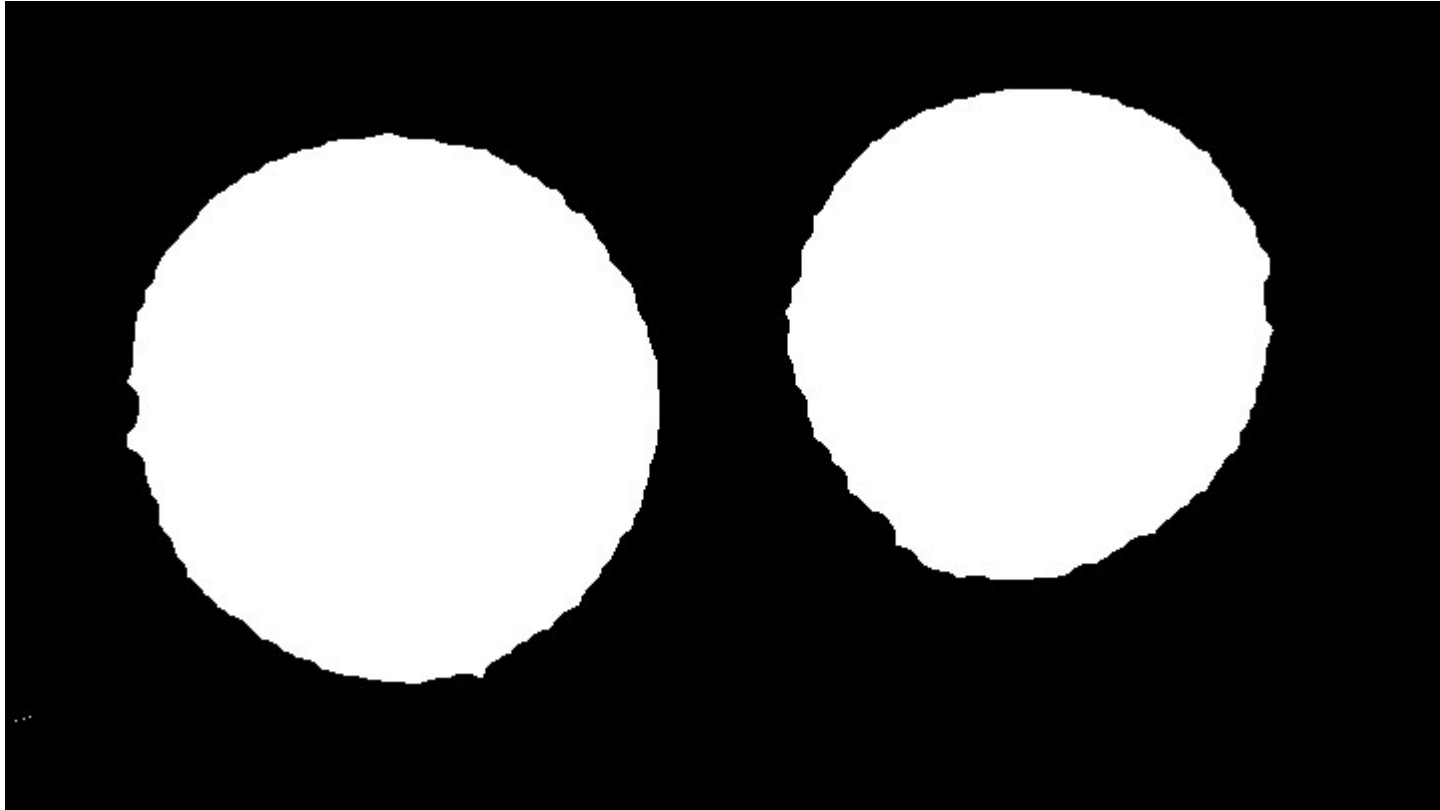
- The mask after closing with a 10x10 circle structuring element



- The mask after closing with a 20x20 box structuring element



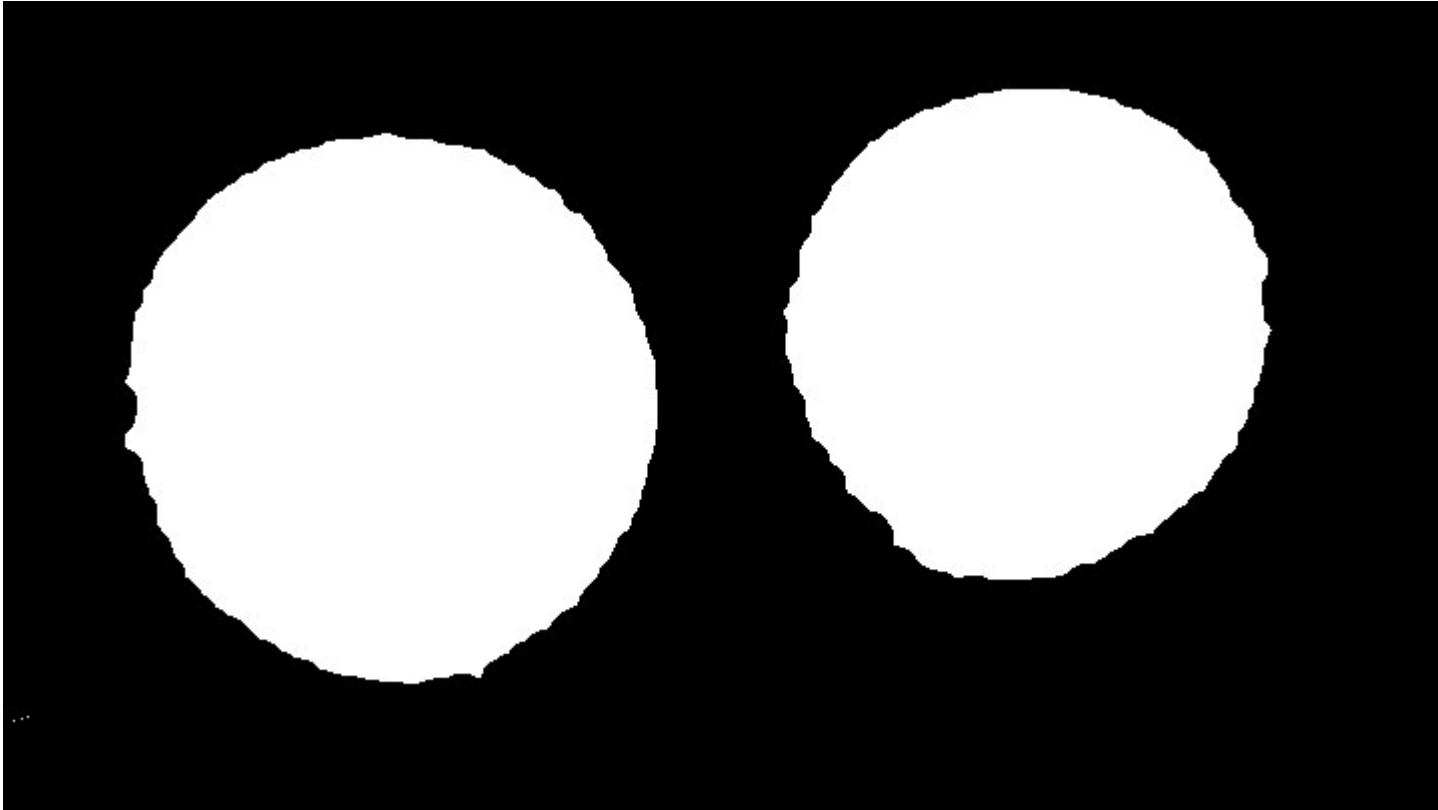
- The mask after closing with a 20x20 circle structuring element



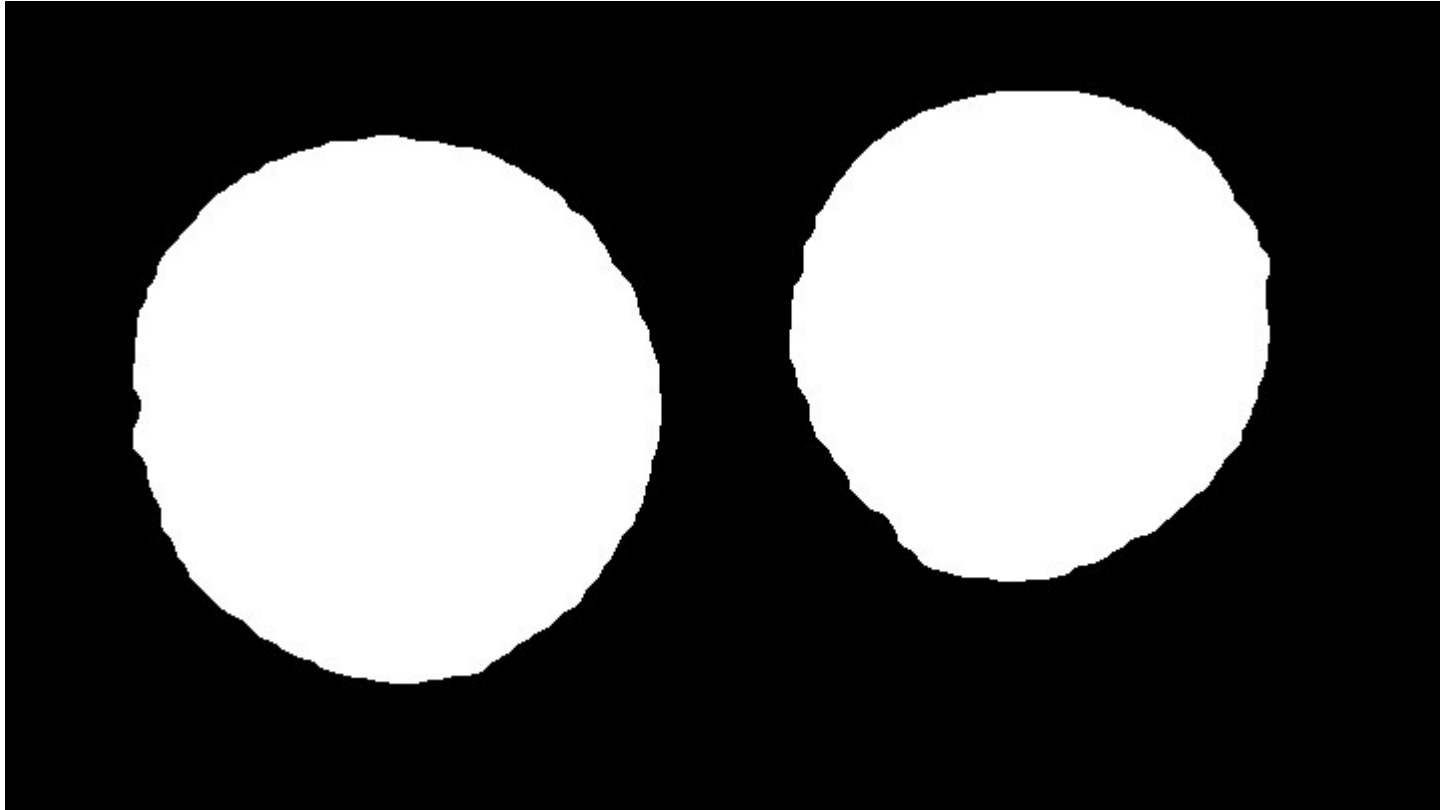
- The image after selecting only the pixels corresponding to the mask closed with a 20x20 circle structuring element



- Let me reshow the mask after closing with a 20x20 circle structuring element



The closed mask from last slide after opening with same (a 20x20 circle structuring element)



The image after selecting only the pixels corresponding to the mask closed and then opened with a 20x20 circle structuring element

