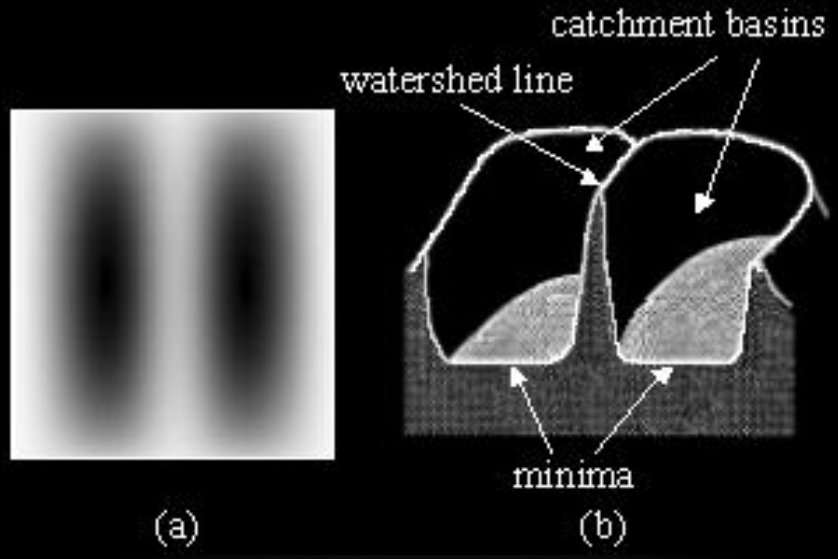
WATERSHED SEGMENTATION FOR BINARY IMAGES WITH DIFFERENT DISTANCE TRANSFORMS

Dragoș Papavă

GMRV

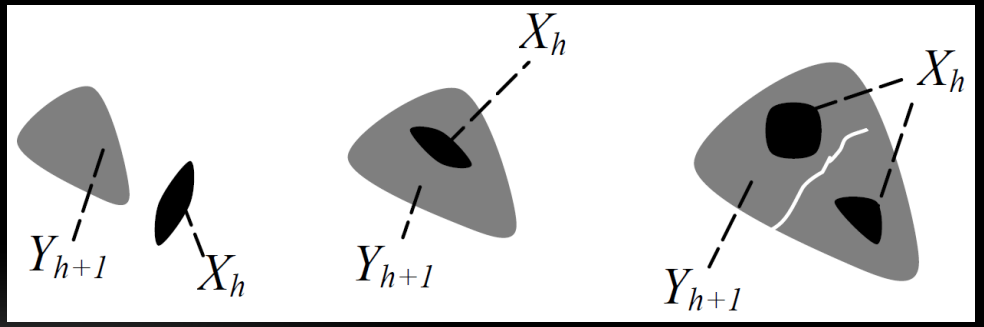
INTRODUCTION

- Image segmentation
  - Partitioning of an image into its constituent regions or objects
  - Extraction of component contours
- Watershed



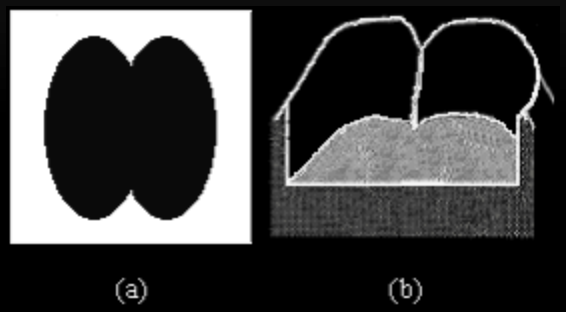
ALGORITHM

- Objective:
  - Finding all the watershed lines
  - The highest gray level
- Steps:
  - (1) Sorting step
    - Creating a list of all the pixels and their gray level values in increasing order
  - (2) Flooding step
    - X_h = pixels of gray level h
    - Y_{h+1} = pixels of gray level ≤ $h+1$
    - Three relations:
      - New minimum
      - Y_{h+1} belongs to basin of X_h
      - Y_{h+1} contains different minima



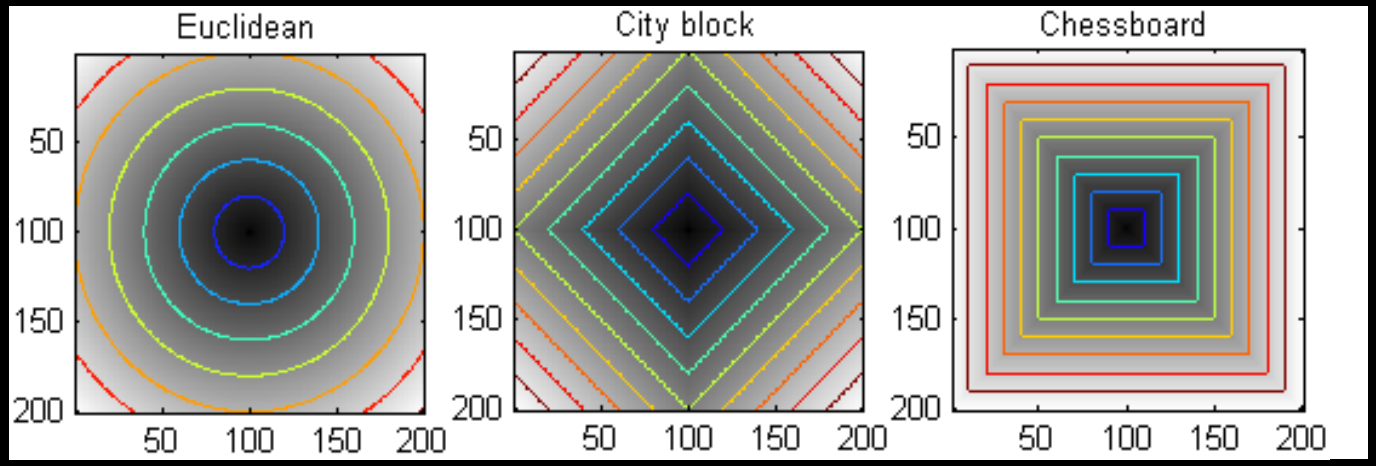
BINARY IMAGES

- The problem:

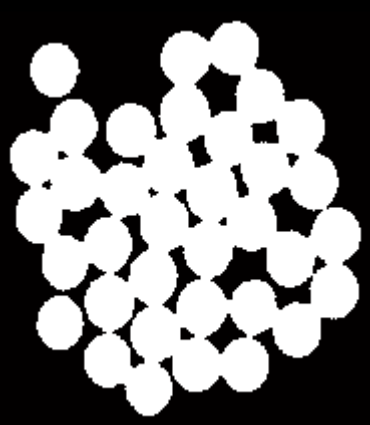


- Solution:
  - Filter the image (a)
  - $d_{Euclidean}([i_1, j_1], [i_2, j_2]) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}$
  - $d_{Cityblock}([i_1, j_1], [i_2, j_2]) = |i_1 - i_2| + |j_1 - j_2|$
  - $d_{Chessboard}([i_1, j_1], [i_2, j_2]) = \max(\big||i_1 - i_2|\big|, |j_1 - j_2|)$

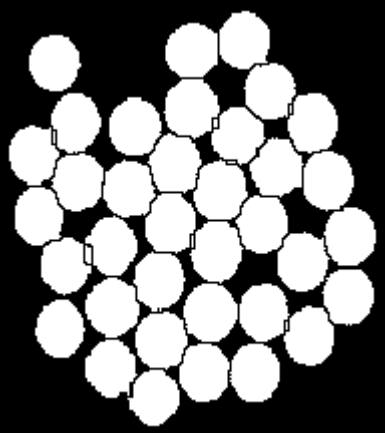
DISTANCE TRANSFORMS



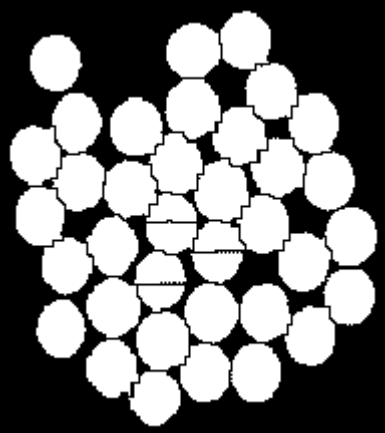
COMPARISON OF EFFECTS



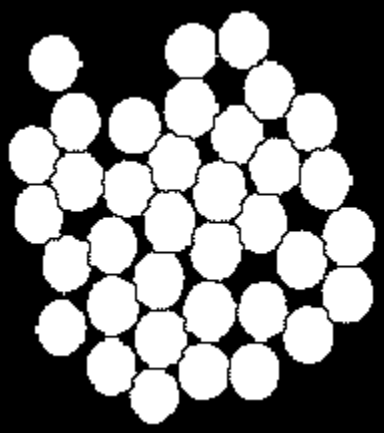
Rocks



Euclidean

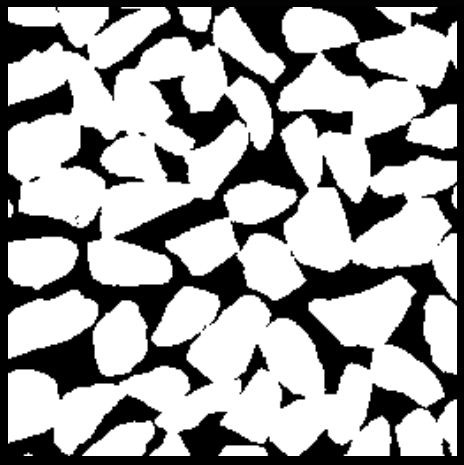


Cityblock

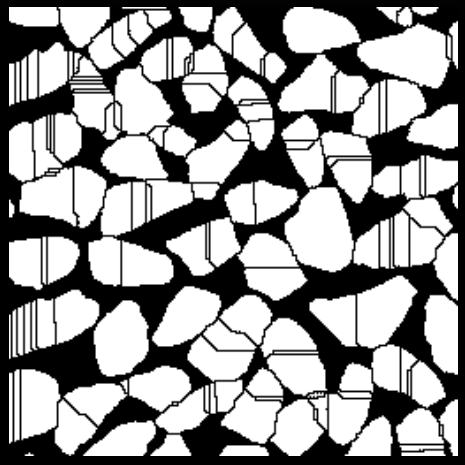


Chessboard

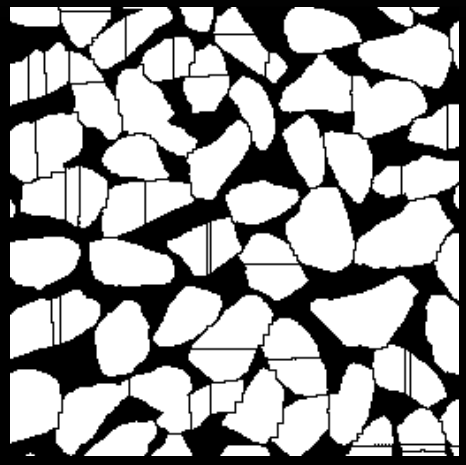
COMPARISON OF EFFECTS



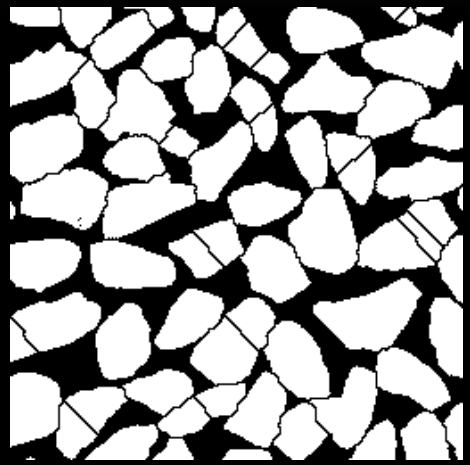
Rocks



Euclidean

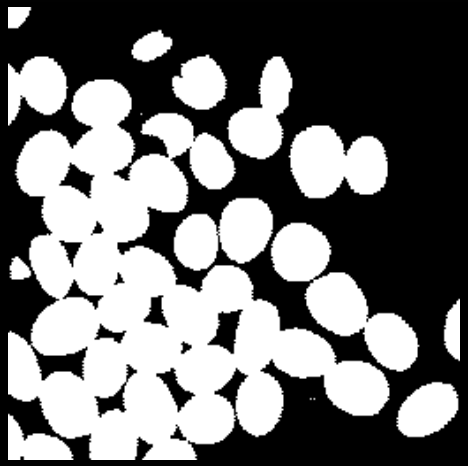


Cityblock

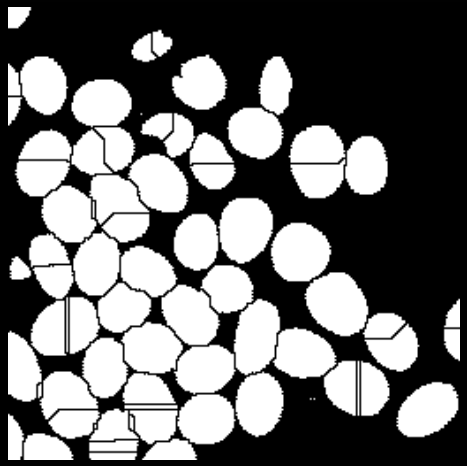


Chessboard

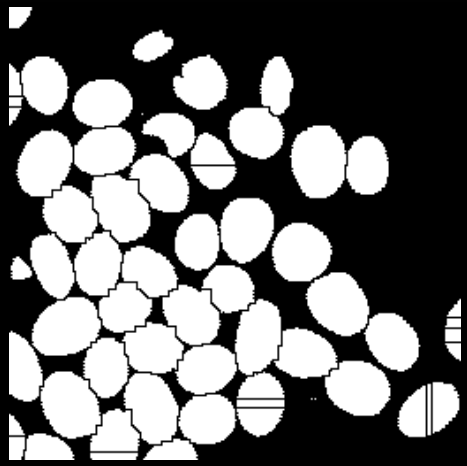
COMPARISON OF EFFECTS



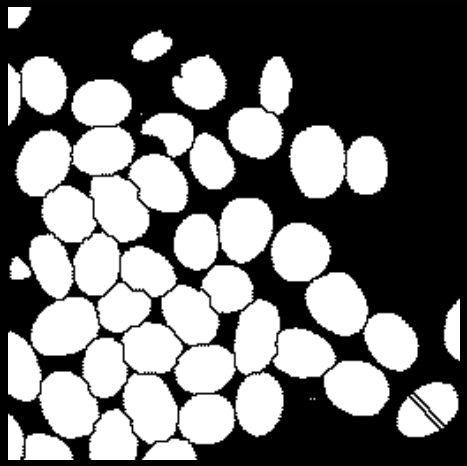
Rocks



Euclidean



Cityblock



Chessboard

RESULTS

Image	NOC* by manual count	NOC by euclidean DT	NOC by cityblock DT	NOC by chessboard DT
Chocolate beans	36	43 (+19% error)	39 (+8% error)	36 (+0% error)
Rocks	60	162 (+170% error)	88 (+47% error)	72 (+20% error)
Coffee beans	50	74 (+48% error)	62 (+24% error)	52 (+4% error)

*NOC – Number of components