Spatial Data and Analysis

Discussion 8

Felipe González

UC Berkeley

October 30th

Outline

1. Image Processing

2. Networks & Markets in Space

Image Processing

- The Image Processing Toolbox in Matlab is a set of functions and apps designed to analyze and visualize images:

  1. Analysis
     - Black & white
     - Histograms
     - Contrast
     - Crop
  2. Object identification
     - Techniques
     - Tools

How to Open and Display Images

- Open Google Earth, go to your favorite location, take a screenshot, and save it in your computer as `NAME.jpg`
- To open an image and save it in Matlab Workspace use the following command:

```
        HOUSE = imread('providencia.jpg');
```

- Your image will be saved as a 3D array ($M \times N \times 3$ array) if it has colors and as a 2D array ($M \times N$) if it is a grayscale image
- You can also open images directly from the Internet replacing 'NAME.jpg' by the corresponding URL of the image

Figure: Satellite image of where I used to live

Tweaking Images #1: Black & White

- We can transform this $M \times N \times 3$ array into a 2D array, i.e., a black and white image, using the following command:

```
HOUSE_BW = rgb2gray(HOUSE);
```

- This command simply uses a weighted average of the RGB colors specified in the third dimension of the image:

$$0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

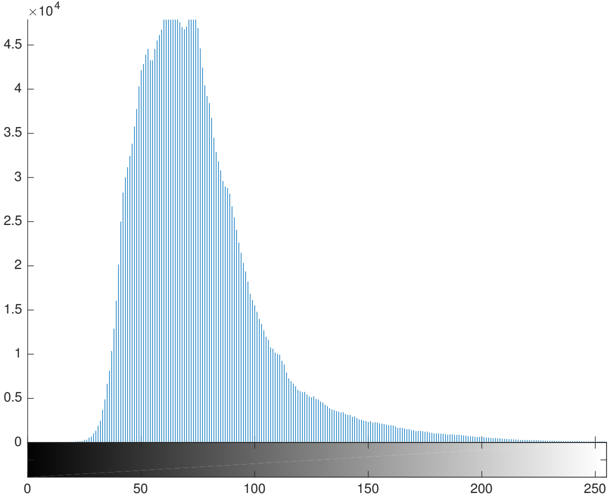
Figure: Black & white

Analyzing Images with Histograms

- Histograms are a useful way to analyze the distribution of colors in an image
- To see the distribution of colors in a black & white picture use the following code:

```
figure
imhist(HOUSE_BW)
```

Figure: Histogram of black & white pixels in the image

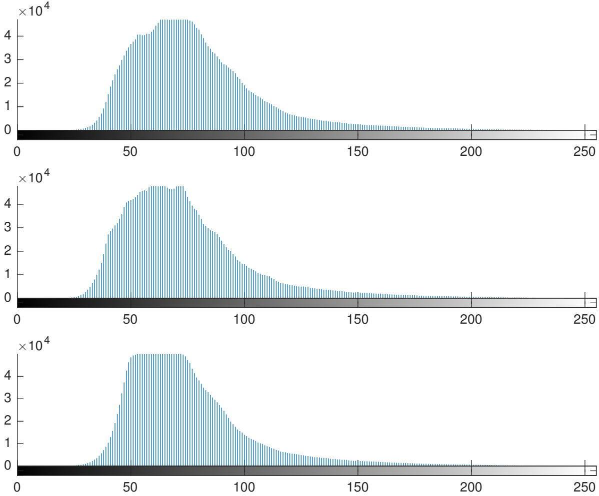


Histograms for RGB Images

- We cannot directly use `imhist` with RGB images because these images are 3D arrays.
- So, we need to separate the image in three parts, one for each layer in the third dimension:

```
figure
for i = 1:3
    subplot(3,1,i) ; imhist(HOUSE(:,:,i))
    box off
end
```

Figure: Histogram of RGB colors in the image



Tweaking Images #2: Contrast

- Another useful command let us enhance the contrast in the image to facilitate the observation of certain objects:

```
HOUSE_BW_C = histeq(HOUSE_BW);
```

- This command transforms the values of the image to replicate a specific histogram

Figure: Enhanced

Tweaking Images #3: Crop

- We can also crop an image by specifying the width and height from specific a specific starting point (x_0, y_0):

```
HOUSE_BW_R = imcrop(HOUSE_BW_C,[500 600 700 800]);
```

- In this case we are cropping our enhanced image starting from point $(500, 600)$ with width 700 and height 800

Figure: Cropped

How to Identify Objects in Images

- Many different techniques to identify objects in an image:
  - Thresholding
  - Hough transformation
  - etc.
- Useful tools:
  - Thresholds
  - Removing small objects
  - Clear border objects

Identify Objects #1: Thresholding

- A simple way of partitioning an image into a foreground and background:

```
        thresh = 200;
        t1 = im2bw(HOUSE_BW_C, thresh/255);
```

- This tool is most useful in images with high levels of contrast

Figure: Threshold := 200

Figure: Threshold := 100

Figure: Threshold := 10

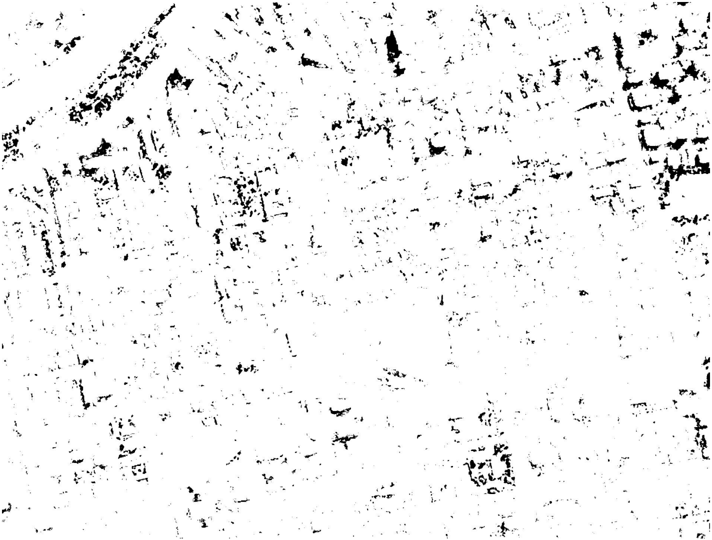
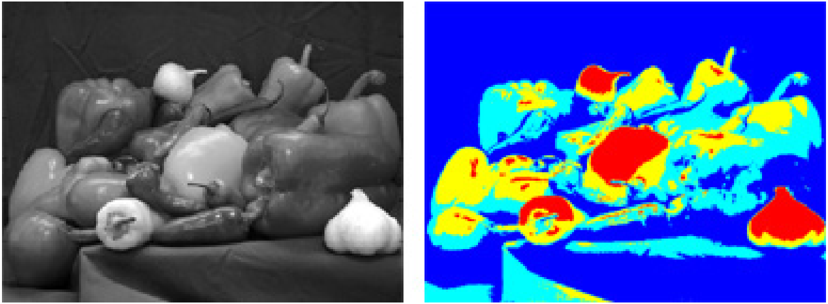


Figure: Multi-level thresholding (`multithres(A)`)



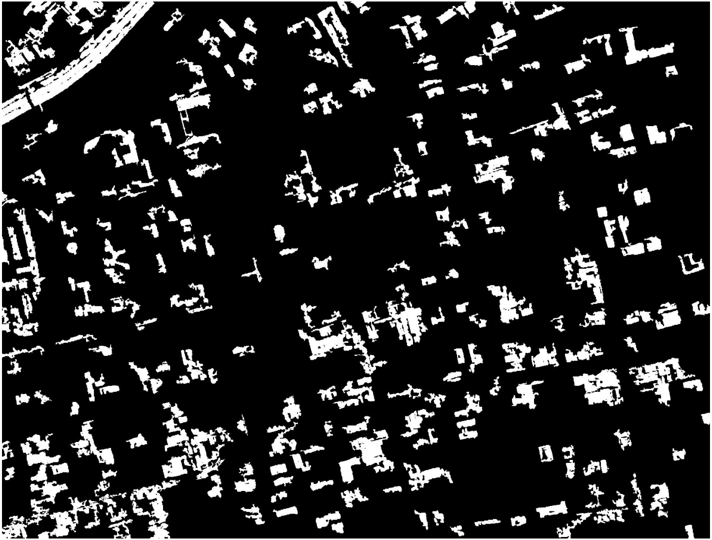
Identify Objects #2: Remove Objects

- Works on binary images (e.g., black and white images)
- Removes small objects (connected components) from image:

```
i1 = bwareaopen(t1, 100);
```

- The second argument specifies the size of connected components

Figure: Removing objects close to my old house



Identify Objects #2: Clear Borders

- We can also clear the borders of an image, i.e., delete structures that are lighter than their surroundings

```
i2 = imclearborder(t1);
```

Figure: Removing objects close to my old house

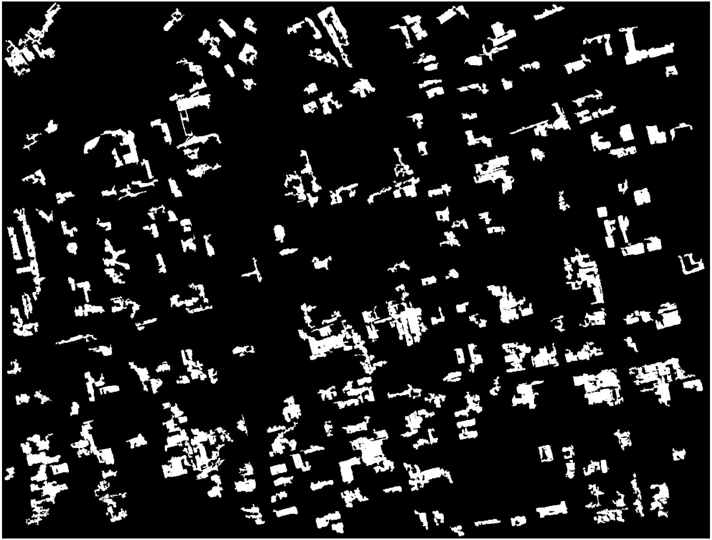
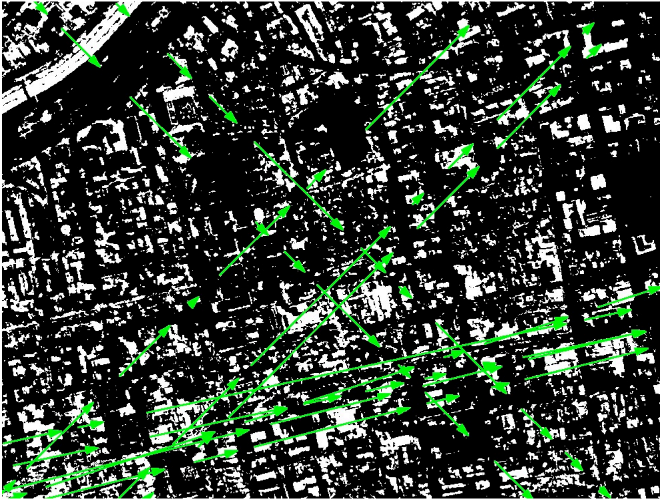


Figure: Identification of roads using Hough transformation



Outline

1. Image Processing

2. Networks & Markets in Space

Markets in Space

- Let there be a geographic space where firms $i = 1, \ldots, N$ operate at a fixed location $\ell_i = (x_i, y_i)$

- Let there be M consumers ($j = 1, \ldots, M$), each one living at a fixed location $\ell_j = (x_j, y_j)$

- Assume you observed consumer choices in space:

$$\begin{aligned} \text{Distances traveled} \quad &: \quad \Theta \;\equiv\; \{\theta_i \;:\; \forall i\} \\ \text{Locations} \quad &: \quad \Gamma \;\equiv\; \{\ell_i, \ell_j \;:\; \forall i, j\} \end{aligned}$$

Figure: N Firms in a Geographic Space

Market Identification

Objective

Create an algorithm that identifies markets from locations and revealed choices (Γ, Θ)

- Two key insights to construct the algorithm:
    - Maximum distance traveled by consumers bounds competition
    - Direct and indirect competition among firms

Figure: Consumers' point of view

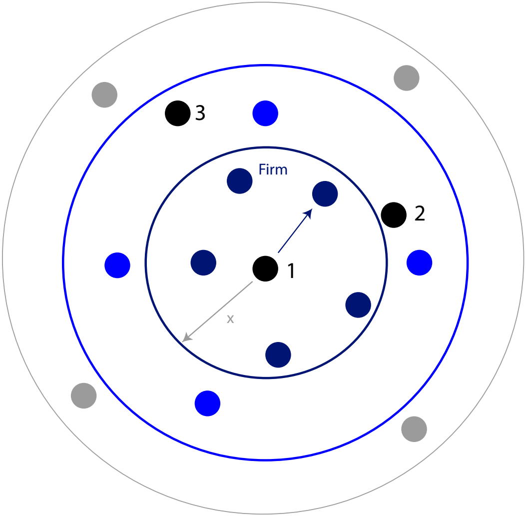


Figure: Competition from firms' perspective

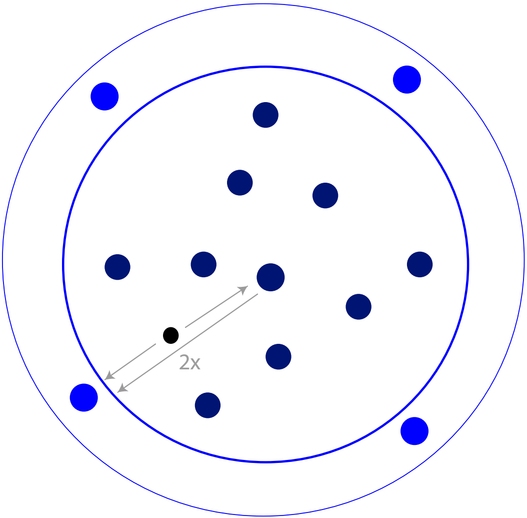


Figure: Second degree competitors

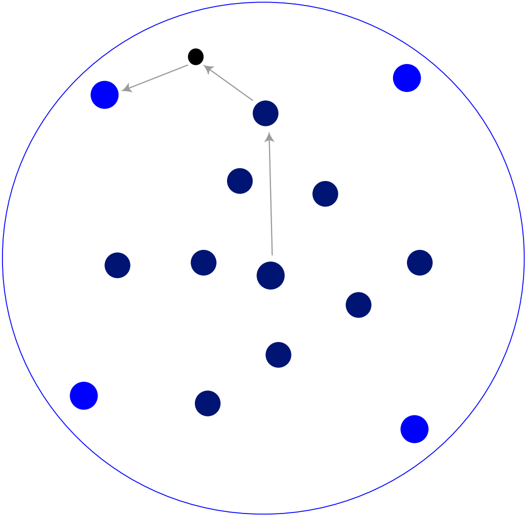
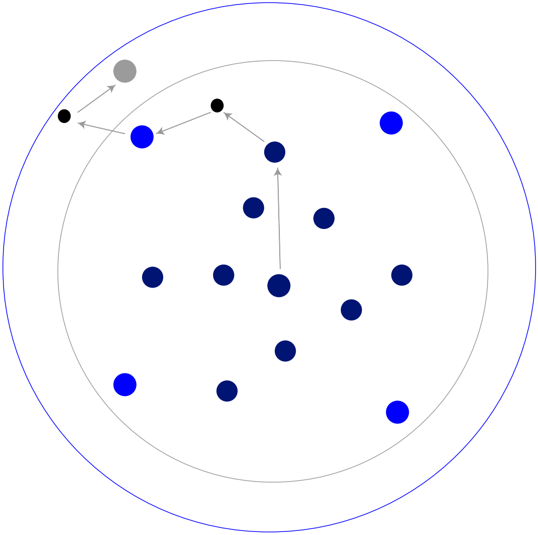


Figure: Third degree competitors



Algorithm

1. Define $2x$ and the maximum theoretical number of possible degrees of competition

2. Calculate adjacency matrix Υ for firms using $2x$ as a threshold for the existence of a link between firms

3. Construct market codes using Υ
    - Focus on firm i
    - Assign market m to firm i and firms connected to i
    - Assign market m to firms with second degree connections to i
    - Repeat for all degrees
    - Repeat for all firms

Figure: Market Identification (x_1)

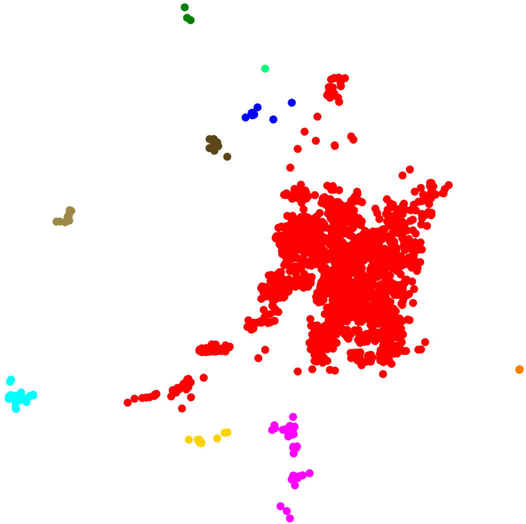


Figure: Market Identification ($x_2 < x_1$)

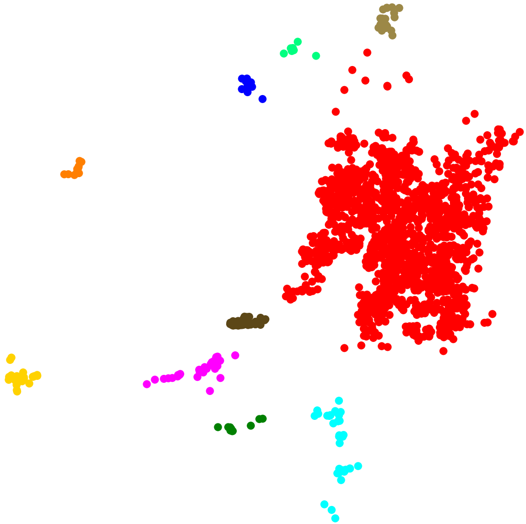


Figure: Market Identification ($x_3 < x_2 < x_1$)