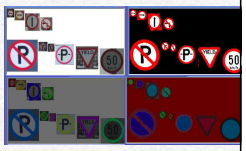
Binary

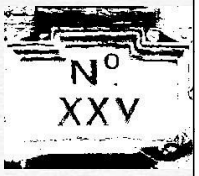
- Thresholding
- Threshold detection
- Variations
- Mathematical Morphology



1

Thresholding

- Binary thresholding
  - for all pixels
    - g(i,j) = 1 for f(i,j) ≥ T
    - = 0 for f(i,j) < T
  - Simple scenes?
  - LUT
    - for all grey levels
      - LUT(k) = 1 for k ≥ T
      - = 0 for k < T
    - for all pixels
      - g(i,j) = LUT(f(i,j))
  - Objects of interest vs. background

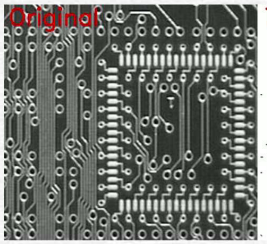



```
threshold(gray_image,binary_image,threshold,
                                    255,THRESH_BINARY);
```


2

Thresholding

- Distinct foreground & background needed

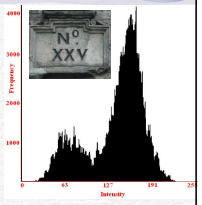


- How do we determine the best threshold?

3

Threshold Detection

- Manual Setting
  - Issue of changing lighting
- Need to determine automatically



- For the techniques which follow:
  - Image – $f(i,j)$
  - Histogram – $h(g)$
  - Probability Distribution – $p(g) = h(g) / \Sigma_g h(g)$

4

Threshold Detection – Otsu Thresholding

- Minimize the spread of the pixels…
  - Smallest within class variance

$$\sigma_W^2(T) = w_f(T)\sigma_f^2(T) + w_b(T)\sigma_b^2(T)$$

$$w_f(T) = \sum_{g=T}^{255} p(g) \qquad \sigma_f^2(T) = \sum_{g=T}^{255} p(g).(g-\mu_f(T))^2 \Big/ w_f(T)$$

$$w_b(T) = \sum_{a=0}^{T-1} p(g) \qquad \sigma_b^2(T) = \sum_{g=0}^{T-1} p(g).(g-\mu_b(T))^2 \Big/ w_b(T)$$

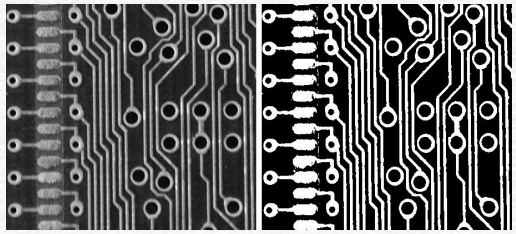
$$\mu_f(T) = \sum_{g=T}^{255} p(g).g \Big/ w_f(T) \qquad \mu_b(T) = \sum_{g=0}^{T-1} p(g).g \Big/ w_b(T)$$

  - Largest between class variance

$$\sigma_B^2(T) = w_f(T)w_b(T)\left(\mu_f(T) - \mu_b(T)\right)^2$$

5

Threshold Detection – Otsu Thresholding



```
threshold( gray_image, binary_image, threshold,
           255, THRESH_BINARY | THRESH_OTSU );
```


6

Variations – Adaptive Thresholding

- The adaptive thresholding algorithm is
  - Divide the image into sub-images,
  - Compute thresholds for all sub-images,
  - Interpolate thresholds for every point using bilinear interpolation.

7

Variations – Adaptive Thresholding

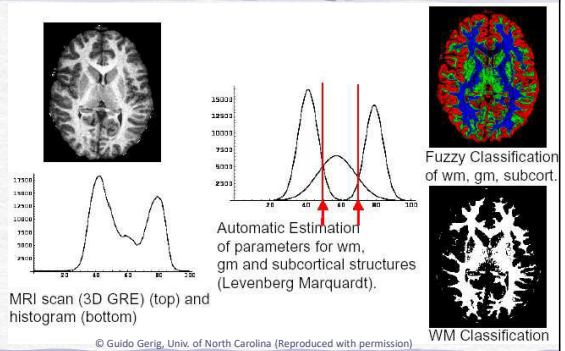
- *OpenCV version:*

if $((f(i,j) - (\Sigma_{a=-m..m,\ b=-m..m} f(i+a,j+b) / (2m+1)^2)) > offset)$
g(i,j) = 255
else g(i,j) = 0


```
adaptiveThreshold( gray_image,binary_image,output_value,
   ADAPTIVE_THRESH_MEAN_C,THRESH_BINARY,
   block_size,offset );
```


8

Variations – Multi-Spectral Thresholding

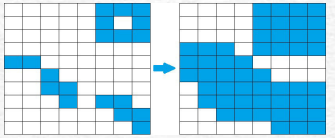


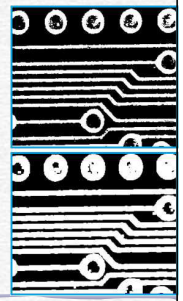
Cleaning Binary Images

- Cannot use normal smoothing operations
- Need to remove noisy points and smooth binary region boundaries.
- Mostly we use operations originally defined as part of "mathematical morphology" (which treats images as sets).
- Most common operations:
  - Erosion
  - Dilation
  - Opening
  - Closing

Mathematical Morphology – Dilation

- Minkowski set addition:
  - $X \oplus B = \{ p \in \varepsilon^2; p = x+b, x \in X \text{ and } b \in B \}$
  - B is the *structuring element*
  - B is typically isotropic
- Adds pixels around borders

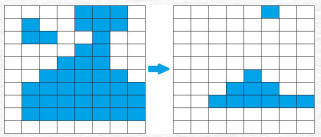


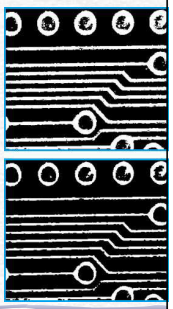


- Effects:
  - Makes 'regions' bigger
  - Fills small holes
  - Joins close 'regions'

Mathematical Morphology – Erosion

- Minkowski set subtraction:
  - $X \ominus B = \{ p \in \varepsilon^2; p+b \in X \text{ for every } b \in B \}$
  - Again B is typically isotropic
- Removes border pixels

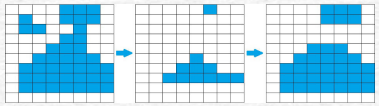




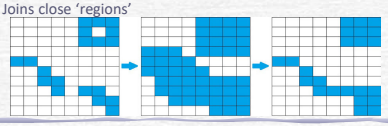
- Effects:
  - Makes 'regions' smaller
  - Removes noise
  - Removes narrow bridges

Mathematical Morphology –

- Opening: $X \circ D = (X \ominus D) \oplus D$
  - Removes noise
  - Removes narrow bridges
  - Roughly maintains 'region' size
  - Smooths shape



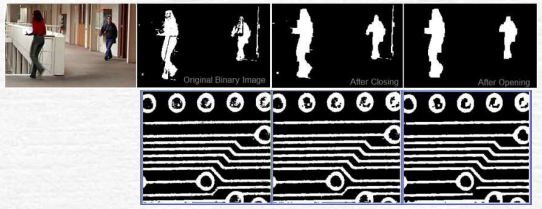
- Closing: $X \bullet D = (X \oplus D) \ominus D$
  - Fills small holes
  - Joins close 'regions'
  - Roughly maintains 'region' size



13

Mathematical Morphology –

- Opening:
  - $X \circ D = (X \ominus D) \oplus D$
- Closing:
  - $X \bullet D = (X \oplus D) \ominus D$
- Isotropic structuring element:
  - Eliminates small image details
- Properties
  - $X \circ D = (X \circ D) \circ D$ and $X \bullet D = (X \bullet D) \bullet D$



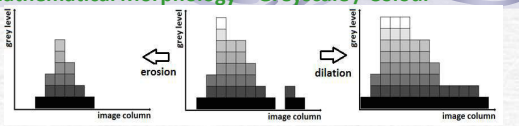
14

Mathematical Morphology – OpenCV Code

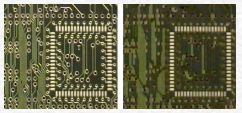
```
dilate( binary_image, dilated_image, Mat());;

Mat structuring_element( 5, 5, CV_8U, Scalar(1) );
dilate( binary_image, dilated_image, structuring_element);

erode( binary_image, eroded_image, Mat());

Mat structuring_element( 5, 5, CV_8U, Scalar(1) );
erode( binary_image, eroded_image, structuring_element);

Mat five_by_five_element( 5, 5, CV_8U, Scalar(1) );
morphologyEx( binary_image, opened_image,
                        MORPH_OPEN,  five_by_five_element );
 morphologyEx( binary_image, closed_image,
                        MORPH_CLOSE, five_by_five_element );
```

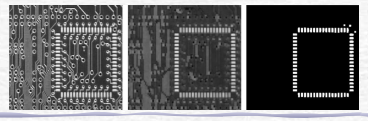

15

Mathematical Morphology – Greyscale / Colour



- One set per grey level (g)
  - All points >= g...





16

Mathematical Morphology – Local maxima

- Can be used to locate local maxima and minima

```
Mat dilated, thresholded_input, local_maxima, thresholded_8bit;
dilate( input, dilated, Mat());
compare( input, dilated, local_maxima, CMP_EQ );
threshold( input, thresholded_input, threshold, 255,
                                                THRESH_BINARY );
thresholded_input.convertTo( thresholded_8bit, CV_8U );
bitwise_and( local_maxima, thresholded_8bit, local_maxima );
```