

Histogram Matching and Image Processing



Histogram equalization matching, Erosion, Dilation, Opening, Closing, Averaging filter, Median filter, Thresholding, global thresholding, Local thresholding, Do histogram matching example numerical



Sure! Let's go through these image processing techniques one by one:

1. **Histogram Equalization Matching:** Adjusting the intensity distribution of an image to match the histogram of another image.
2. **Erosion:** Reduces the boundaries of the foreground object.
3. **Dilation:** Increases the boundaries of the foreground object.
4. **Opening:** Erosion followed by dilation, useful for removing small objects.
5. **Closing:** Dilation followed by erosion, useful for closing small holes.
6. **Averaging Filter:** Smooths images by reducing noise, computes the average of the pixel values in the kernel area.
7. **Median Filter:** Smooths images by reducing noise, computes the median of the pixel values in the kernel area.
8. **Thresholding:** Converts a grayscale image to a binary image.
 - **Global Thresholding:** Uses a single threshold value for the entire image.
 - **Local Thresholding:** Uses different threshold values for different regions.

Let's look at an example of histogram matching numerically.

Histogram Matching Example

Suppose we have two images:

- Image A: 50, 80, 100, 150, 200
- Image B: 0, 50, 100, 150, 200

Steps:

1. **Compute the histograms:**
 - Histogram of Image A: {50: 1, 80: 1, 100: 1, 150: 1, 200: 1}
 - Histogram of Image B: {0: 1, 50: 1, 100: 1, 150: 1, 200: 1}
2. **Compute the cumulative distribution functions (CDF):**
 - CDF of Image A: $1/5, 2/5, 3/5, 4/5, 5/5$
 - CDF of Image B: $1/5, 2/5, 3/5, 4/5, 5/5$
3. **Match the CDFs:**
 - Map each pixel value in Image A to the corresponding pixel value in Image B based on their CDF values.
4. **Transformation:**

- 50 in A matches 0 in B (1/5 in A matches 1/5 in B)
- 80 in A matches 50 in B (2/5 in A matches 2/5 in B)
- 100 in A matches 100 in B (3/5 in A matches 3/5 in B)
- 150 in A matches 150 in B (4/5 in A matches 4/5 in B)
- 200 in A matches 200 in B (5/5 in A matches 5/5 in B)

Thus, the new values for Image A after histogram matching to Image B are:

- Transformed Image A: 0, 50, 100, 150, 200

Applying Filters

For the sake of brevity, I'll provide concise definitions and examples for the other techniques.

Erosion and Dilation

- **Erosion:** Reduces object boundaries by eroding away pixels at the edges. Example using a 3x3 kernel:

mathematica

Original:

```
1 1 1
1 1 1
1 1 1
```

After Erosion:

```
0 0 0
0 1 0
0 0 0
```

- **Dilation:** Expands object boundaries by adding pixels to the edges. Example using a 3x3 kernel:

mathematica

Original:

```
0 0 0
0 1 0
0 0 0
```

After Dilation:

```
1 1 1
1 1 1
1 1 1
```

Opening and Closing

- **Opening:** Erosion followed by dilation. Removes small objects/noise.
- **Closing:** Dilation followed by erosion. Closes small holes.

Averaging and Median Filters

- **Averaging Filter:** Smooths images by averaging pixel values in a kernel.

yaml

```
Original:
10 10 10
10 20 10
10 10 10

After Averaging:
10 11 10
11 11 11
10 11 10
```

- **Median Filter:** Smooths images by taking the median of pixel values in a kernel.

mathematica

```
Original:
10 10 10
10 50 10
10 10 10

After Median:
10 10 10
10 10 10
10 10 10
```

Thresholding

- **Global Thresholding:** Converts to binary using a single threshold.

makefile

```
Threshold = 100
Original: 50, 150, 200
Binary: 0, 1, 1
```

- **Local Thresholding:** Converts to binary using different thresholds for different regions.

Would you like to see a specific implementation or further details on any of these techniques?

ChatGPT can make mistakes. Check important info.