Image Registration: Preprocessing Operations

Ardy Goshtasby
Wright State University and Image Fusion Systems Research
Preprocessing Operations

All operations performed on images that improve the registration performance. These include:

– Image enhancement
– Image segmentation
– Edge detection
Image Enhancement

• This involves image denoising and image deblurring. Denoising is the process of reducing image noise, while deblurring is the process of reducing image blur.
• Denoising is achieved by image smoothing or filtering.
• Deblurring is achieved by image sharpening or inverse filtering.
**Image Smoothing**

Given image \( f(x,y) \) and smoothing filter \( h(i,j) \), image smoothing is defined by:

\[
\bar{f}(x, y) = \sum_{i=-k}^{k} \sum_{j=-l}^{l} f(x + i, y + j) h(i, j)
\]

The intensity of pixel \((x,y)\) in the output is obtained from a weighted sum of intensities of pixels at and surrounding \((x,y)\) in the input. \( h(i,j) \) for different \( i \) and \( j \) shows the weights, whose sum is 1.
Mean Filtering

- When the weights defined by $h(i,j)$ are all the same, the operation is known as mean filtering.
- Intensities of all pixels at and surrounding $(x,y)$ in input have the same effect on the intensity at $(x,y)$ in output.
- This operation is not rotationally invariant if $h$ is nonzero within a rectangular window. To obtain a rotationally invariant smoothing, $h$ should represent a circular window.
Gaussian Filtering

• If the weights used in image smoothing represent Gaussian coefficients, the process is known as Gaussian filtering.
• Gaussian and mean filtering are effective when image noise has a mean of zero.
• Gaussian filtering is rotationally invariant.
• Gaussians extend to infinity from all sides, so all pixels in input affect the value at a pixel in output.
• A 2-D Gaussian can be decomposed into 2-D 1-D Gaussians, and filtering can be carried in 1-D rather than in 2-D: $G(x,y) = G(x) \ast G(y)$
Computation of Mean and Gaussian Filtering

Although filtering is a convolution operation and can be computed using the FFT algorithm, since FFT considers an image a periodic 2-D signal, if left and right image borders, or top and bottom image borders have different properties, artifacts will be obtained near the image borders. To avoid this, the computations should be performed directly.

- Image containing Zero-mean noise
- Computed with FFT
- Computed directly

Gaussian filter of $\sigma = 2$ pixels
Median Filtering

- The value at \((x,y)\) in output is obtained from the median of values at and surrounding \((x,y)\) in input.
- To ensure median filtering is rotationally invariant, median of circular rather than rectangular windows should be used.
- Median filtering is effective when impulse noise is present in the image.

![Image containing 2% impulse noise](image1)
![Filter radius 2 pixels](image2)
![Filter radius 4 pixels](image3)

Median filtering
Image Sharpening

- This is the reverse of image smoothing. The objective is to enhance image details.
- It is achieved by the deblurring or inverse filtering operation.
- Given blurred image $g$ and blurring filter $h$, the sharp image (image before being blurred) is estimated from:

$$
\hat{f}(x, y) = \mathcal{F}^{-1}\left\{\frac{G(u, v)}{H(u, v)}\right\}
$$

- Find the Fourier transforms of the blurred image, divide by the Fourier transform of the filter point-by-point, and find the inverse Fourier transform of the result.
- Note that the Fourier transform of the filter should not contain any zeros.
Image Sharpening Example

(a) An outdoor scene image. (b) – (d) Sharpened image using different deblurring filters.
Image Segmentation

This is the process of partitioning an image into meaningful regions. There are two main approaches to image segmentation.

– Methods that use information within regions to form regions of homogeneous property.
– Methods that use information about property changes between regions to form region boundaries.
Intensity Thresholding

- Assuming an image contains objects of homogeneous property, and property values within an object have a Gaussian distribution, the image can be segmented using threshold values that represent the valley between the Gaussian modes.
- This method works well when an image contains homogeneous objects and background and the properties of the object and background are different.
- The threshold value may be computed by:
  - Detecting the valley between the modes in the histogram of the image.
  - Finding the property value that represents the average of properties of highest-gradient pixels.
  - Finding the property value at which a small change in the property value will minimally change the segmentation result.
Intensity Thresholding Example

A Landsat image

Thresholding at the valley between the first two peaks.

Histogram of the image

Threshold value

Thresholding at the average intensity of highest-gradient pixels.
Edge Detection

• Edge detection methods can be categorized into ones that search for locally maximum gradients and ones that search for zeros of the image second derivative.

• Methods that are search for maximum gradients find fewer false edges but they miss some good edges, disconnecting object boundaries.

• Methods that are based on image second derivative find closed boundary regions, but they also pick up some false boundaries.
Laplacian of Gaussian (LoG) Edge Detector

- In this method, the image is first smoothed to reduce noise. Then, the second derivative (Laplacian) of the smoothed images is computed. The obtained image will have positive and negative regions. The boundary between the positive and negative regions is taken as the image edges.

- This process is the same as convolving the image with the Laplacian of a 2-D Gaussian and finding its zero-crossings.
LoG Edge Detector

Determination of the LoG of an image involves computation of:

\[
\text{LoG}[f(x, y)] = \frac{\partial^2 [f(x, y) \ast G(x, y)]}{\partial x^2} + \frac{\partial^2 [f(x, y) \ast G(x, y)]}{\partial y^2} \\
= f(x, y) \ast \frac{\partial^2 G(x, y)}{\partial x^2} + f(x, y) \ast \frac{\partial^2 G(x, y)}{\partial y^2}
\]

or,

\[
\text{LoG}[f(x, y)] = \frac{\partial^2 G(x)}{\partial x^2} \ast G(y) \ast f(x, y) + G(x) \ast \frac{\partial^2 G(y)}{\partial y^2} \ast f(x, y)
\]
LoG Edge Detection Example

An X-ray angiogram

LoG zero-crossing edges
Edge Detection by Intensity Ratios

- Zero-crossing edges can be determined by convolving the negative and positive parts of the LoG with an image separately and subtracting the convolved images and locating the zero-crossings.
- If instead of subtracting corresponding values in the convolved images, we divide the values and locate the one-crossings, intensity-ratio edges will be obtained.
Intensity Ratio Edge Detection Example

(a) Intensity ratio edges. (b) 30% highest-gradient ratio edges. (c) Intensity difference edges. (d) 30% highest-gradient difference edges.

Intensity ratios detect more edges in dark areas while intensity differences detect more edges in bright areas.
Canny Edge Detector

- This method smoothes the image with a Gaussian to reduce noise and locates locally maximum gradients in the gradient direction and uses them as the edges.

An X-ray angiogram

Canny edges with $\sigma = 2.5$ pixels
Edge Detection in 3-D

• This is similar to edge detection in 2-D except that a 3-D Gaussian is used and LoG becomes:

\[
\text{LoG} \left[ f(x, y, z) \right] = \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) G(x, y, z) \ast f(x, y, z)
\]

\[
= \frac{\partial^2 G(x)}{\partial x^2} \ast G(y) \ast G(z) \ast f(x, y, z) \\
+ \frac{\partial^2 G(y)}{\partial y^2} \ast G(x) \ast G(z) \ast f(x, y, z) \\
+ \frac{\partial^2 G(z)}{\partial z^2} \ast G(x) \ast G(y) \ast f(x, y, z).
\]
3-D Edge Detection Example

A volumetric MR brain image

LoG edges
Edges in Color Images

• Edge detection in a color image can be considered edge detection in a 2-D vector field.
• If $u(x,y)$ and $v(x,y)$ represent color gradients in $x$ and $y$ directions, edges can be considered points where color gradients are locally maximum in the gradient direction.
• If $R(x,y)$, $G(x,y)$, and $B(x,y)$ represent red, green, and blue color components at $(x,y)$, respectively, color gradients are:

$$u(x,y) = \frac{\partial R(x,y)}{\partial x} \mathbf{r} + \frac{\partial G(x,y)}{\partial x} \mathbf{g} + \frac{\partial B(x,y)}{\partial x} \mathbf{b}$$

$$v(x,y) = \frac{\partial R(x,y)}{\partial y} \mathbf{r} + \frac{\partial G(x,y)}{\partial y} \mathbf{g} + \frac{\partial B(x,y)}{\partial y} \mathbf{b}$$

$r$, $g$, and $b$ are unit vectors along red, green, and blue axes, respectively, in the color space.
Color Edges

• Gradient direction at \((x,y)\) is the direction maximizing:

\[
F(x, y) = \left[ u(x, y) \cos \theta(x, y) + v(x, y) \sin \theta(x, y) \right]^2
\]

• The maximum \(\theta(x,y)\) is obtained by finding the derivative of \(F(x,y)\) with respect to \(\theta(x,y)\), setting it to zero and solving for \(\theta(x,y)\). Doing so, we obtain:

\[
\theta(x, y) = 0.5 \tan^{-1} \left( \frac{2u(x, y) \cdot v(x, y)}{u(x, y) \cdot u(x, y) - v(x, y) \cdot v(x, y)} \right)
\]
Color Edge Detection Example

A color image

Edges of the color image