Image inpainting

Outline

- Why inpaint?
- Image filtering (convolution), simplest inpainting
- Heat flow, diffusion process, diffusion equation
- Anisotropic diffusion
- Further development

Partly taken from "Restoring profanity" by Carola Schönlieb

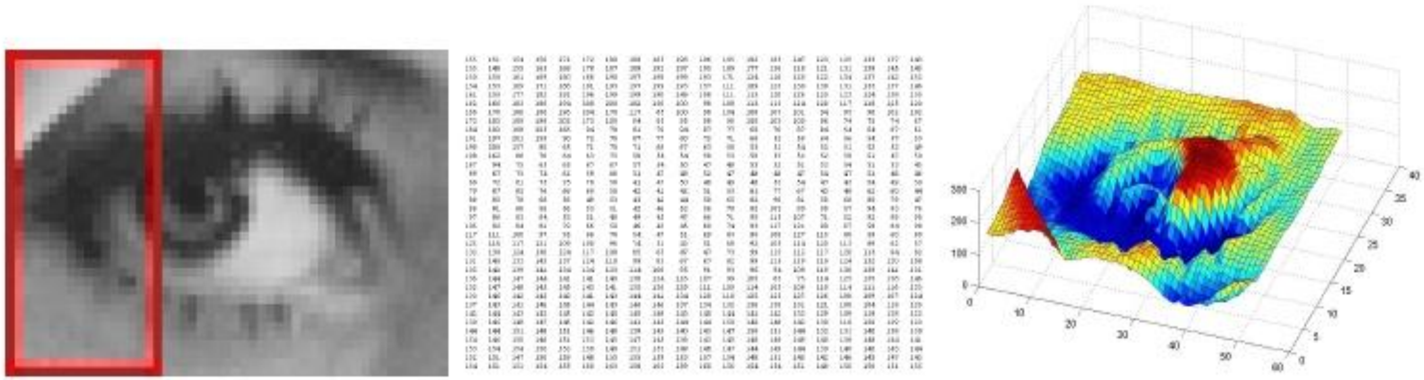
Inpainting applications

- Image restoration
- Object removal
- Text removal
- Special effects

Image courtesy Andrea Baczynski

Simplest inpainting

- Image is 2D discrete function



- Form an iterative algorithm
- Each update gets you closer to the desired result

$$I_{i,j}^{n+1} = I_{i,j}^{n} + \Delta t I_{t,i,j}^{n} \quad \forall (i,j) \in \Omega$$

Simplest inpainting

- Smooth image (using Gaussian filter)
- Replace missing pixels by smoothed version
- Repeat until the result stops changing

(MATLAB demo)

conv_intro.m

Convolution

Convolution of two functions f(x) and g(x)

$$h(x) = f(x) \otimes g(x) = \int_{-\infty}^{+\infty} f(r) g(x-r) dr$$

- Output filtered image → $h(x)$
- Image → $f(x)$
- convolution operator → $\otimes$
- Filter (mask/kernel) → $g(x)$
- Support region of filter where g(x-r) is nonzero → $x-r$

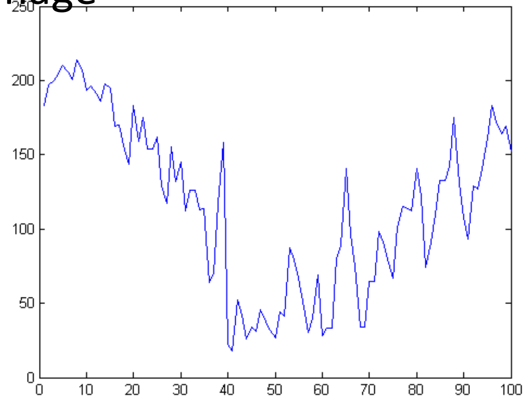
Discrete image processing 2D form

$$H(x,y) = \sum_{j=1}^{height} \sum_{i=1}^{width} I(i,j) M(x-i, y-j)$$

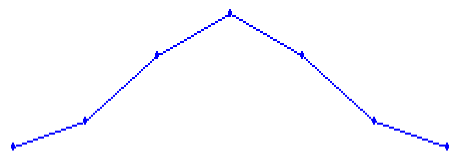
Compute the convolution where there are valid indices in the kernel

Convolution example in 1D

Horizontal slice from Mandrill image

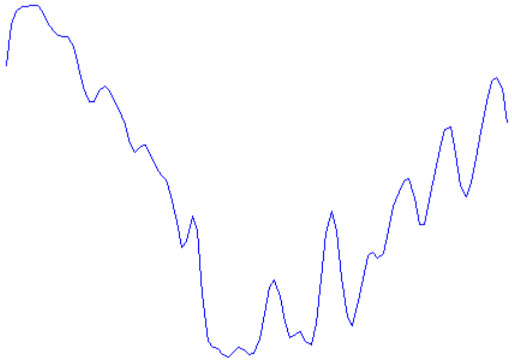


1D Gaussian filter



=

Filtered Signal



0.01	0.08	0.24	0.34	0.24	0.08	0.01

Common convolution kernels

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

Arithmetic mean filter (smoothing)
>>fspecial('average')

-0.17	-0.67	-0.17
-0.67	3.33	-0.67
-0.17	-0.67	-0.17

Laplacian (enhance edges)
>>fspecial('laplacian')

-0.17	-0.67	-0.17
-0.67	4.33	-0.67
-0.17	-0.67	-0.17

Sharpening filter
>>fspecial('unsharp')

0.01	0.08	0.01
0.08	0.62	0.08
0.01	0.08	0.01

Gaussian filter (smoothing)
>>fspecial('gaussian')

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

Sobel operators (edge detection in x and y directions)
>>fspecial('sobel')
>>fspecial('sobel')’

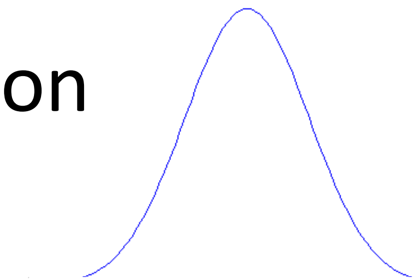
Investigate the listed kernels in Matlab by performing convolutions on the Mandrill and Lena images. Study the effects of different kernel sizes (3x3, 9x9, 25x25) on the output.

The median filter is used for noise reduction. It works by replacing a pixel value with the median of its neighbourhood pixel values (vs the mean filter which uses the mean of the neighbourhood pixel values). Apply Matlab’s median filter function medfilt2 on the Mandrill and Lena images. Remember to use different filter sizes (3x3, 9x9, 16x16).

Useful functions for convolution

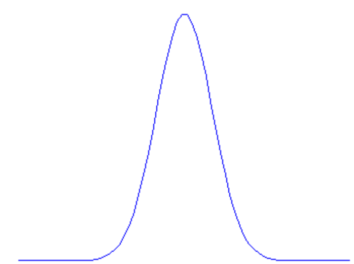
- Generate useful filters for convolution

>>fspecial('gaussian',[kernel_height kernel_width],sigma)



- 1D convolution

>>conv(signal,filter)



- 2D convolution

>>conv2(double(I(:,:,2)),fspecial('gaussian‘,[kernel_height kernel_width] ,sigma),'valid')

image

kernel

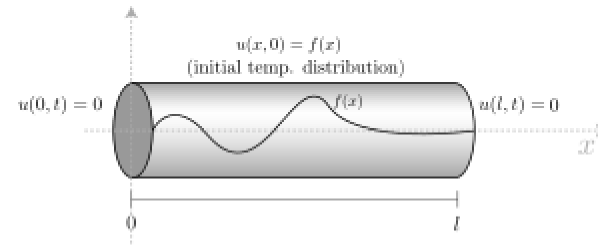
Border padding options

Perform the convolution of an image using Gaussian kernels with different sizes and standard deviations and display the output images.

Heat (diffusion) equation

- 1822 Joseph Fourier: heat propagation in rod

$$\frac{\partial u(x,t)}{\partial t} = a^2 \frac{\partial^2 u(x,t)}{\partial x^2}$$



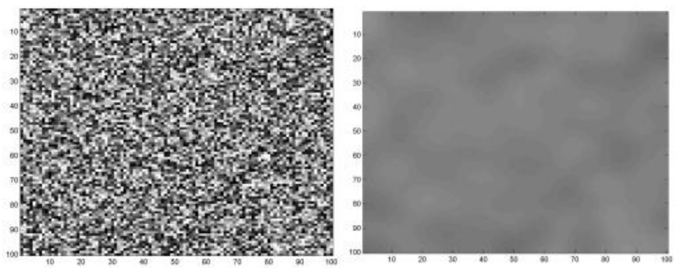
- Same in image (2D)

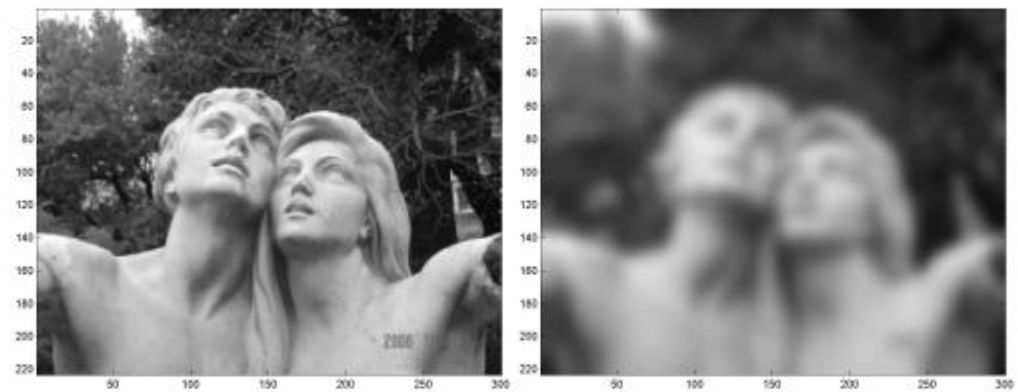
$$I_t(t,x,y) = a^2(I_{xx} + I_{yy}) = a^2 \Delta I$$

- Model intensity propagation as diffusion

Diffusion equation solution

- Left: initial distribution, right: t=10

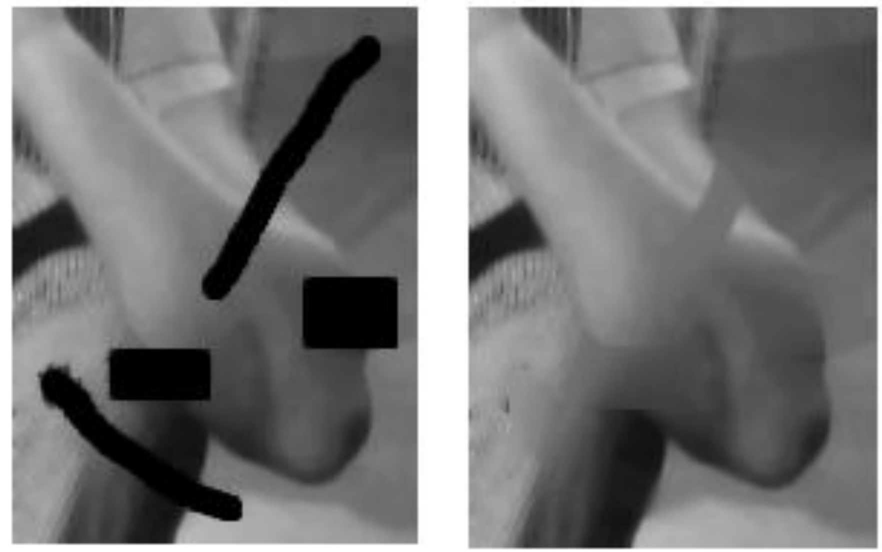




- Damaged image:

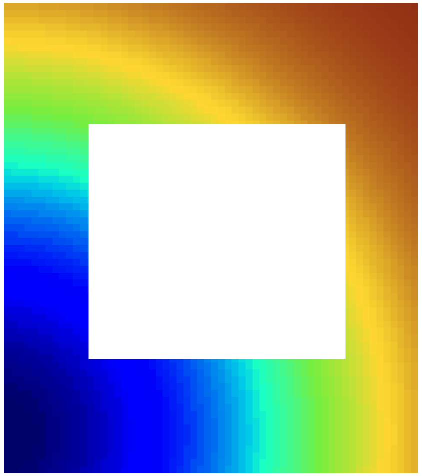
Anisotropic diffusion

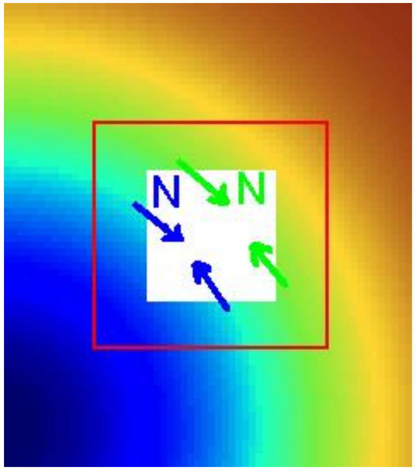
- Problem: edges are not preserved
- Do as professional restorator: continue along edges
- Solution: use anisotropic diffusion equation



Anisotropic diffusion (2)

- Fill the hole by isophote direction



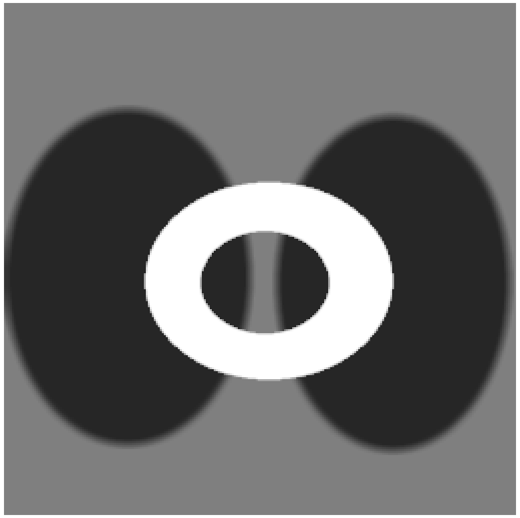


$$\vec{N} = \left[-I_y, I_x\right]$$

- Anisotropic equation

$$I_t = \nabla(\Delta I) \cdot \vec{N}$$

PDE inpainting examples



Example: photo restoration

Further development

- Higher order equations for curvature minimization (smooth contour joining)
- Texture synthesis for large inpainting areas
- Combined approaches