Image Enhancement

DD2423 Image Analysis and Computer Vision

Mårten Björkman

Computational Vision and Active Perception
School of Computer Science and Communication

November 15, 2013

Image enhancement by filtering

- Primary goal: noise removal
- Requirement: preserve relevant information
- It may be difficult to define “relevant information”, since it depends on the task, environment, etc.

Approaches:

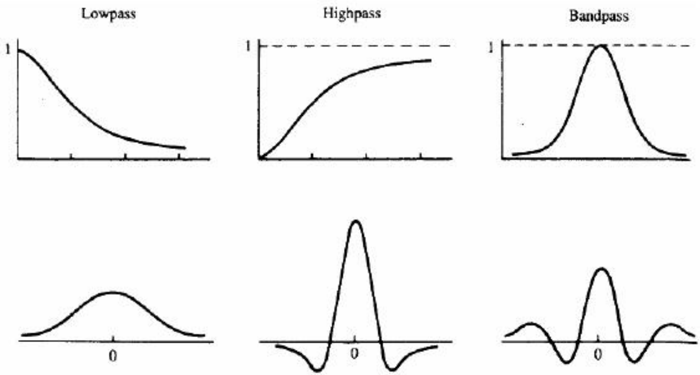
- Noise is identifiable
  - remove and interpolate
- Noise is not identifiable
  - image averaging, low-pass filtering
  - median filtering
  - min/max filtering
- Contrast augmentation
  - high-pass filtering

Spatial filtering

- Use of spatial masks for filtering is called spatial filtering.

  May be linear or nonlinear.
- Linear filters can be:
  - Lowpass: eliminate high frequency components such as characterized by egdes and sharp details in an image.
    $\Rightarrow$ Net effect is image blurring.
  - Highpass: eliminate low frequency components such as slowly varying characteristics (shadings).
    $\Rightarrow$ Net effect is sharpening of edges and other details (also noise).
  - Bandpass: eliminate outside a given frequency range.
    $\Rightarrow$ Combination of the above. Common in practice.

Spatial filtering (examples)



Some filters in frequency domain and corresponding spatial filter masks.

Exercise

- Assume you have a filter kernel $[1, 0, -1]$. How does this look like in the Fourier domain? Is it a lowpass, highpass or bandpass filter?

- Assume you have a filter kernel $[1, 0, -1]$. How does this look like in the Fourier domain? Is it a lowpass, highpass or bandpass filter?

  Answer: To see this we have to express the filter in continuous domain, which we can do with Dirac functions.

$$f(x) = \delta(x) - \delta(x-2)$$

  To get the Fourier Transform we exploit the sifting property of Dirac functions.

$$\hat{f}(u) = \int_x f(x)e^{-iux}dx = 1 - e^{-2iu} = e^{-iu}(e^{iu} - e^{-iu}) = 2ie^{-iu}\sin(u)$$

$$\|\hat{f}(u)\| = 2\|\sin(u)\|$$

  Since $\|\hat{f}(0)\| = \|\hat{f}(\pi)\| = 0$ and $\|\hat{f}(\pi/2)\| = 2$, it is a bandpass filter.

Different kinds of noise

- Noise is the result of errors in the image acquisition that result in pixel values that do not reflect the true intensities of the real scene (scanning devices, CCD detector, transmission)
- Signal independent additive noise (sampling noise)

$$g = f + \nu$$

- Signal dependent multiplicative noise (illumination variations)

$$g = f + \nu f = (1 + \nu) f$$

- Measurement noise (salt and pepper)

Linear smoothing - Averaging

Idea: Average over K data points and reduce variance of uncorrelated noise by factor of K.

Ensemble average:

$$F_1, \ldots, F_K \rightarrow \boxed{} \rightarrow G$$

$F_1, \ldots, F_k$ represents several almost identical images (several images of the same static scene).

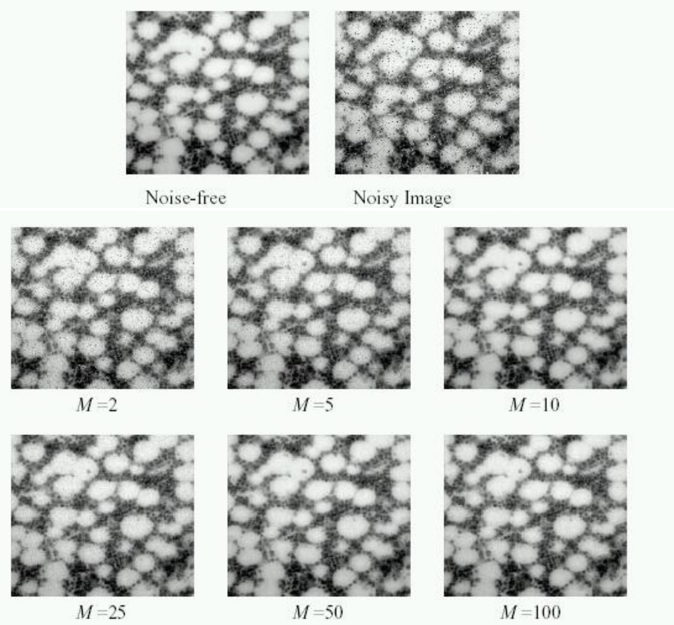
$$G(x) = \frac{1}{K} \sum_{k=1}^{K} F_k(x)$$

$$\begin{array}{rc} N(\mu, \sigma^2) & \{\text{one image}\} \\ N(K\mu, K\sigma^2) & \{\text{sum of } K \text{ images}\} \\ N(\mu, \sigma^2/K) & \{\text{average of } K \text{ images}\} \end{array}$$

\+ Excellent method (e.g. with poor cameras)

\- Requires time (and static scenes) - not always possible

Image averaging



Local spatial averaging / Mean filtering

- Let $N(x)$ represent neighborhood of a point x and

$$G(x) = \sum_{\eta \in N(x)} C_\eta F(x - \eta)$$

- Often $\sum C_\eta = 1$, Example: $N = N_8, C_\eta = \frac{1}{9}$ gives $\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$

Two main problems with mean filtering:

- A single pixel can significantly affect the mean value of all the pixels in its neighborhood (errors are spread).
- It blurs edges - a problem if we require sharp edges in the output.

Local spatial averaging (continue)

- Common requirements:
  - Coefficients should sum up to 1.
  - Symmetric up/down and left/right.
  - Center pixel has most influence on output.
  - Filter should be separable.
- These result in:

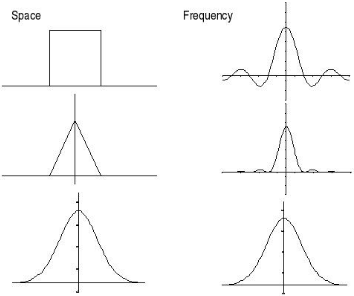
$$C_\eta = \begin{pmatrix} \frac{\Delta t}{2} \\ 1-\Delta t \\ \frac{\Delta t}{2} \end{pmatrix} \begin{pmatrix} \frac{\Delta t}{2} & 1-\Delta t & \frac{\Delta t}{2} \end{pmatrix} = \begin{pmatrix} \frac{\Delta t^2}{4} & \frac{\Delta t}{2}(1-\Delta t) & \frac{\Delta t^2}{4} \\ \frac{\Delta t}{2}(1-\Delta t) & (1-\Delta t)^2 & \frac{\Delta t}{2}(1-\Delta t) \\ \frac{\Delta t^2}{4} & \frac{\Delta t}{2}(1-\Delta t) & \frac{\Delta t^2}{4} \end{pmatrix}$$

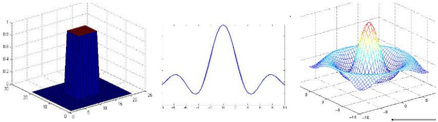
- Special case: $\Delta t = \frac{1}{2}$ gives $\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}$

Basic idea

- Most information in images is concentrated at low frequencies.
- Noise is uniformly distributed over all frequencies (white noise).

  $\Rightarrow$ Suppress high frequency.
- Different filters have different qualities in Fourier space.





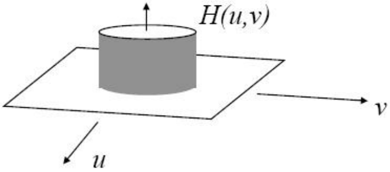
Ideal low pass filter

A transfer function for a 2-D ideal lowpass filter (ILPF) is given as

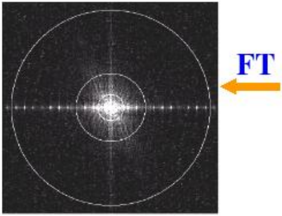
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

where D_0 is a stated nonnegative quantity (the cutoff frequency) and D(u,v) is the distance from the point (u,v) to the center of the frequency plane

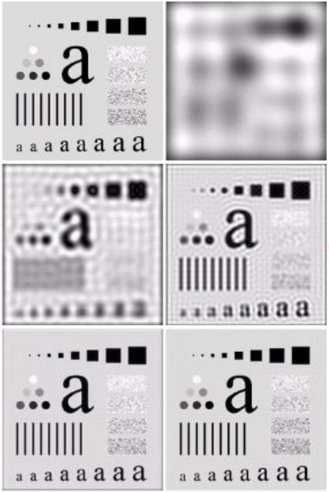
$$D(u,v) = \sqrt{u^2 + v^2}$$



Ideal low pass filter



Ideal in frequency domain means non-ideal in spatial domain, vice versa.



ringing and blurring

Butterworth low pass filter

For the Butterworth filter, the function defining the scaling of each point in the frequency spectrum is as follows:

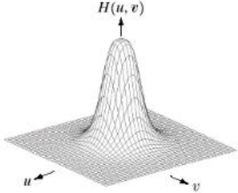
$$H(u,v) = \frac{1}{1 + \left(\frac{D(u,v)}{D_0} \right)^{2n}}$$

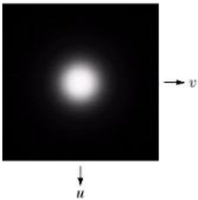
Here D(u,v) is the distance of point (u,v) from the center,
D_0 is the cutoff distance (radius) of the filter
n is the order of the filter

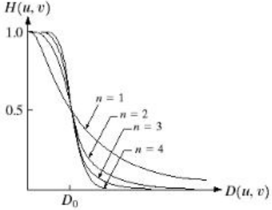
Butterworth low pass filter (continue)

- Smooth transfer function, no sharp discontinuity, no clear cutoff frequency.

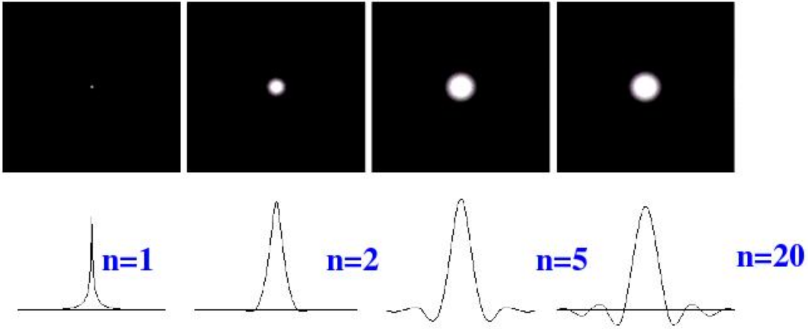
$$H(u,v)=\frac{1}{1+\left[\frac{D(u,v)}{D_0}\right]^{2n}}$$







Butterworth low pass filter (continue)



$$H(u,v)=\frac{1}{1+\left[\frac{D(u,v)}{D_0}\right]^{2n}}$$

Good compromise: Gaussian low-pass filter

$$g(x,y;\sigma^2) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$
$$\hat{g}(u,v;\sigma^2) = e^{-\sigma^2(u^2+v^2)/2}$$

where (u^2+v^2) = squared distance from the origin.

- The parameter measures spread of Gaussian curve. Smaller the value, the larger the cutoff frequency and milder the filtering. When $(x^2+y^2) = \sigma^2$, the filter is at 0.607 of its maximum value.

- Note: Gaussian in spatial domain and Gaussian in frequency.

Binomial kernels

- The filter $(\frac{\Delta t}{2}, 1 - \Delta t, \frac{\Delta t}{2})$ can for $\Delta t = \frac{1}{2}$ be written

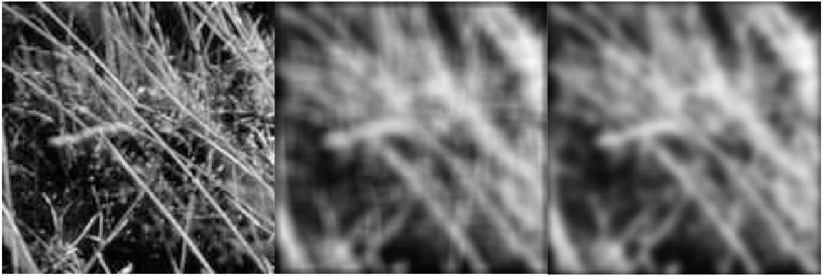
$$(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}) = \frac{1}{4}(1, 2, 1) = \frac{1}{2}(1, 1) * \frac{1}{2}(1, 1)$$

Repeated use of $(1, 1)$ kernels gives rise to Pascal's triangle.

coefficients	normalization factors
1	1
1 1	1/2
1 2 1	1/4
1 3 3 1	1/8
1 4 6 4 1	1/16
1 5 10 10 5 1	1/32
1 6 15 20 15 6 1	1/64
⋮	⋮

- Central limit theorem ⇒ kernels approach Gaussian kernels.

Image averaging: Average vs. Gaussian



Original Average Gaussian

Anisotropic smoothing

- Anisotropic smoothing: smooth differently in different directions.
- Idea: smooth pixels based on similarity $s(p,q)$ between pixels.

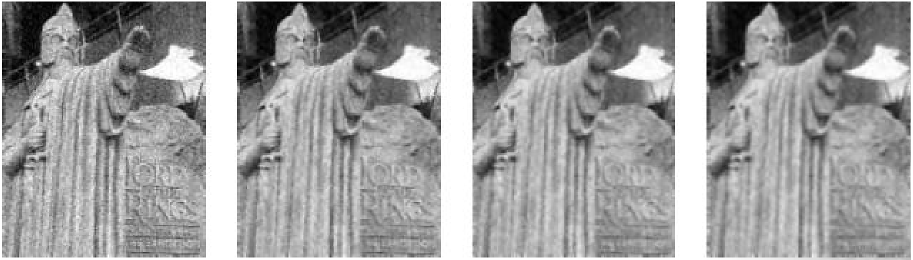
$$p' = \frac{\sum q \cdot s(p,q)}{\sum s(p,q)}$$

- Similarity $s(p,q)$ can be measured in colour, position, etc.
- Examples:

$$s(p,q) = e^{(\frac{p-q}{K})^2}, \;\; s(p,q) = \frac{1}{1-(\frac{p-q}{K})^2}$$

- Problems: Different kernels at different positions $\Rightarrow$ Impossible to analyse in frequency space.

Anisotropic smoothing



K = 5 | K = 25 | K = 50 | K = 100

Higher values of K gives greater smoothing.

Non-linear filtering

- Nonlinear spatial filters also operate on neighborhoods.
- Operations are based directly on pixel values in neighborhood.

  They do not explicitly use coefficient values as in filter masks.
- Purpose: Incorporate prior knowledge to avoid destructive behavior, typically at edges and corners.
- Basic methods:

  - median filtering

  - min/max filtering

  - selective averaging

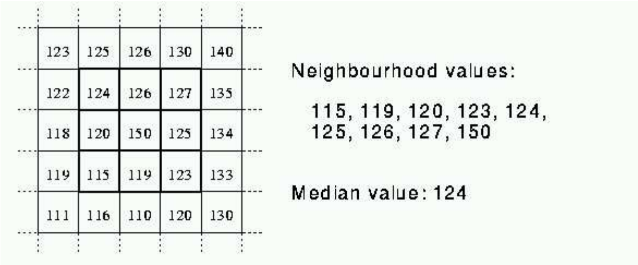
  - weighted averaging

Median filtering

$$G(m) = median_{k \in N(m)} \ F(k)$$

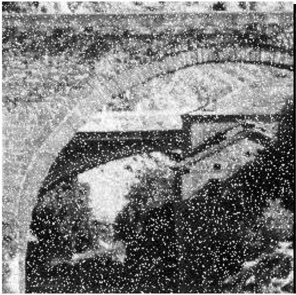
Properties:

+ Preserves the value in 1D monotonic structures (shading).
+ Preserves the position of 1D step edges.
+ Eliminates local extreme values (e.g. salt-and-pepper).
- Creates painting-like images.



Median filtering - example

Original



med5x5

mean5x5

Min/max filtering

- Suppress bright areas on dark background (or vice versa)

$$G(m) = \min_{k \in N_m} F(k)$$

- Properties:
  - bright areas decrease
  - isolated bright points disappear

Sharpening

- Purpose: Enhance local contrast, highlight fine details.
- Methods:

  - Unsharp masking

  - High-pass filtering (spectral)

  - Differentiation (first and second order derivatives)
- Common desirable property:

  - Isotropy (rotational invariance)
- Common problems:

  - Differentiation and high-pass filtering enhance noise
- Difference compared to grey-level transformations:

  - Spatial variations are taken into account

Easiest way: unsharp masking

- Idea: "subtract out the blur"
- Blur image → subtract from original → weight → add to original

$$g(x,y) = f(x,y) + \alpha(f(x,y) - \bar{f}(x,y))$$

Original (× A) Original Blur

$$\frac{1}{A}\left[\begin{bmatrix}0&0&0\\0&A&0\\0&0&0\end{bmatrix} + \left(\begin{bmatrix}0&0&0\\0&5&0\\0&0&0\end{bmatrix} - \begin{bmatrix}0&1&0\\1&1&1\\0&1&0\end{bmatrix}\right)\right]$$

$$= \frac{1}{A}\begin{bmatrix}0&-1&0\\-1&A+4&-1\\0&-1&0\end{bmatrix}$$

High-pass filters

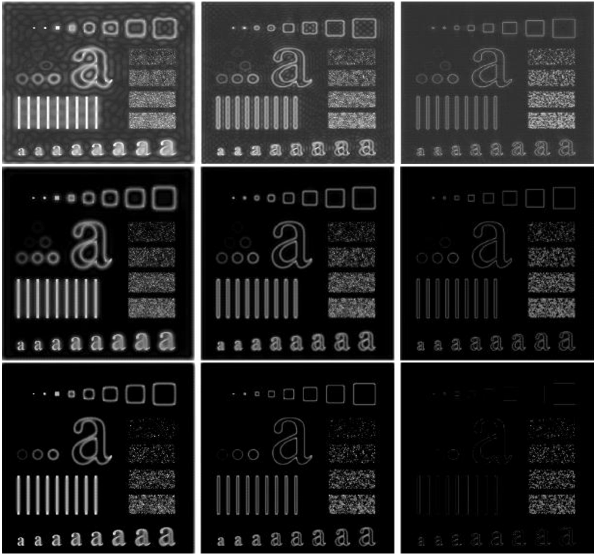
- Sharpening with a high-pass filter:

$$G(u,v) = F(u,v) + \alpha(H_{hp}(u,v)F(u,v))$$

- Quite similar to unsharp masking, but in Fourier domain.

  - $H_{hp}(u,v)=1-H_{lp}(u,v)$
  - Ideal: $H(u,v)=1 \quad D(u,v)>D_0$; $H(u,v)=0 \quad D(u,v)\leq D_0$
  - Butterworth: $$|H(u,v)|^2=\frac{1}{1+\left[\frac{D_0}{D(u,v)}\right]^{2n}}$$
  - Gaussian: $$H(u,v)=1-e^{-D^2(u,v)/2D_0^2}$$

High-pass filters



Results with ideal (top), Butterworth (middle) and Gaussian (bottom) filters.

Differentiation

- Requirements for a first order derivative operator:
  1. zero in flat areas
  2. non-zero along ramp signals of constant slope
  3. non-zero in the onset and end of a gray-level step or ramp
- Requirements for a second order derivative operator:
  1. zero in flat areas
  2. zero along ramp signals of constant slope
  3. non-zero at the onset and end of a gray-level step or ramp

First and Second order derivative

- Basic definition of a first order x-wise derivative operator:

$$f_x = f(x+1, y) - f(x, y)$$

Similarly, a first order y-size derivative f_y can be defined.

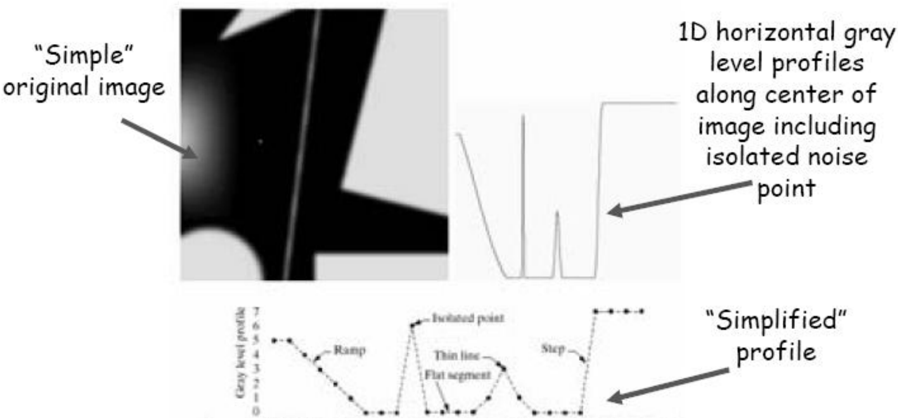
- More common in practice (derivative at x, not $x + 0.5$):

$$f_x = \frac{1}{2}(f(x+1, y) - f(x-1, y))$$

- Second order x-wise derivative operator:

$$f_{xx} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

Derivatives



f	5	5	4	3	2	1	0	0	0	6	0	...	0	0	7	7	7	7
f_x		-1	-1	-1	-1	-1	0	0	6	-6	0	...	0	7	0	0	0	
f_{xx}		-1	0	0	0	0	1	0	6	-12	6	...	0	7	-7	0	0	

Derivatives

- Along a ramp f_x is non-zero, while f_{xx} is zero.
- f_{xx} enhances final details than f_x (but also enhances noise).
- Magnitude of f_x can be used to detect edges.
- f_{xx} produces two values for every edge (positive and negative).
- Sign of f_{xx} tells whether a pixel near an edge is dark or white.

Examples of differentiation operators

- We are interested in filters whose response is independent of the direction of discontinuities in the image.

  Isotropic filters are rotationally invariant: rotating the image and then applying the filter is the same as applying the filter first and then rotating the image.

- Gradient: $\nabla f = (f_x, f_y)$

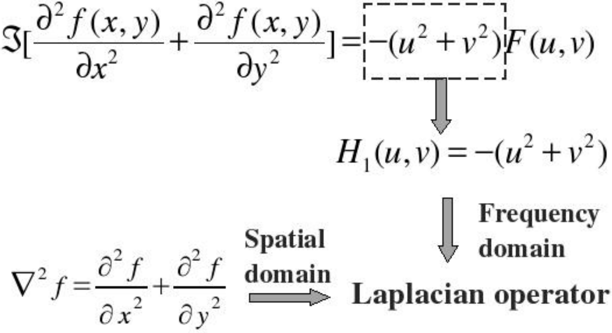
  First order, linear, non-isotropic

- Gradient magnitude: $\mid \nabla f \mid = \sqrt{f_x^2 + f_y^2}$

  First order, non-linear, isotropic

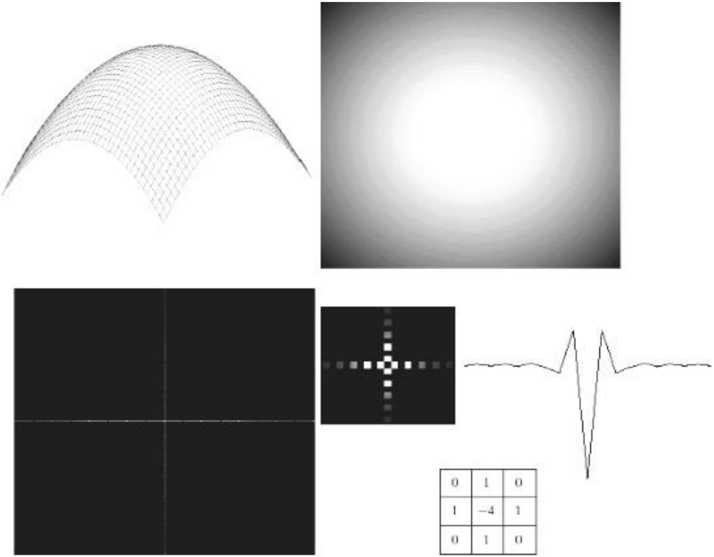
- Laplacian: $\nabla^2 f = f_{xx} + f_{yy}$

  2nd order, linear, isotropic



Isotropic: depends only on the distance from origin, not on the angle.

Laplacian operator



Laplacian in frequency (upper) and spatial (lower) domain.

Laplacian in action

enhanced image — Original image — Laplacian output

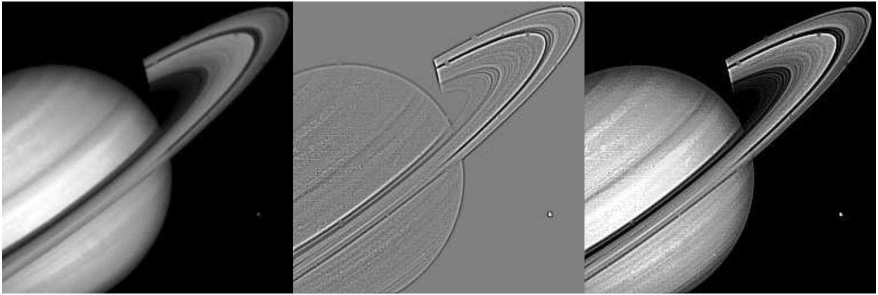
Spatial domain $g(x,y) = f(x,y) - \nabla^2 f(x,y)$

Frequency domain $G(u,v) = F(u,v) + (u^2 + v^2)F(u,v)$

Laplacian

new operator $H_2(u,v) = 1 + (u^2 + v^2) = 1 - H_1(u,v)$

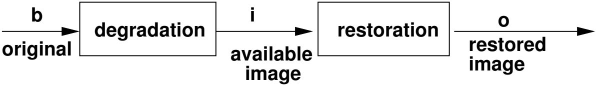
Application of the Laplacian operator



Original image (left), application of Laplacian operator (middle), and subtraction of the Laplacian from the original image (right).

Image restoration

- Image degradation model:



- Task: Use the image degradation model to restore the original image as well as possible.
- Common degradation mechanisms:
  - smoothing, imaging defects
  - defocusing, motion blur
  - noise (sensor noise, quantization)
- Model: linear shift invariant filter; uncorrelated and additive noise.
- Reality: non-linear shift dependent degradation; correlated and non-additive noise.

Inverse filtering

$$f \to \boxed{\quad *h \quad} \to g = h * f$$

$$\hat{f} \to \boxed{\quad \hat{h} \quad} \to \hat{g} = \hat{h}\,\hat{f}$$

- How to recreate f from g ?
- Formally simple. Let

$$\hat{h}' = \frac{1}{\hat{h}} \Rightarrow \hat{g}\hat{h}' = \frac{\hat{h}\hat{f}}{\hat{h}} = \hat{f} \quad \text{(inverse filtering)}$$

- Problems:
  - $\hat{h}'$ undefined when $\hat{h}(\omega) = 0$.
  - Inverse Fourier transform of $\hat{h}'$ not necessarily convergent.
  - Noise enhanced at frequencies where $|\,\hat{h}(\omega)\,|$ is small.

Summary of good questions

- What are the differences between lowpass, bandpass and highpass filters?
- What kind of noise can you have?
- Why does image averaging work?
- Why are ideal lowpass filter rarely used in practice?
- What characteristics does a Gaussian filter have?
- What is the difference between mean and median filters?
- How can you do sharpening?
- How can you approximate a first order derivative?
- What is a Laplacian?
- Why is inverse filtering hard?

Readings

- Gonzalez & Woods: Chapters 3.4 - 3.6, 4.7 - 4.10
- Szeliski Chapters 3.2 and 3.3.1