Spatial Filtering

Srikumar Ramalingam
School of Computing
University of Utah

[Slides borrowed from Ross Whitaker]

Overview

- Correlation and convolution
- Linear filtering
  - Smoothing, kernels, models
  - Detection
  - Derivatives
- Nonlinear filtering
  - Median filtering
  - Bilateral filtering
  - Neighborhood statistics and nonlocal filtering

Cross Correlation

- Operation on image neighborhood and small …
  - "mask", "filter", "stencil", "kernel"
- Linear operations within a moving window

Input image

100	130	104	99	…
87	95	103	150	…
50	36	150	104	…
20	47	205	77	…
.	.	.	.	
.	.	.	.	

Filter

0.0	0.1	0.0
0.1	0.6	0.1
0.0	0.1	0.0

Output image

.	.	.	.
.	.	.	.
.	55.8	134.8	.
.	.	.	.

```
  0.0*87 + 0.1*95 + 0.0*103
+ 0.1*50 + 0.6*36 + 0.1*150
+ 0.0*20 + 0.1*47 + 0.0*205 = 55.8 134.8
```

Cross Correlation

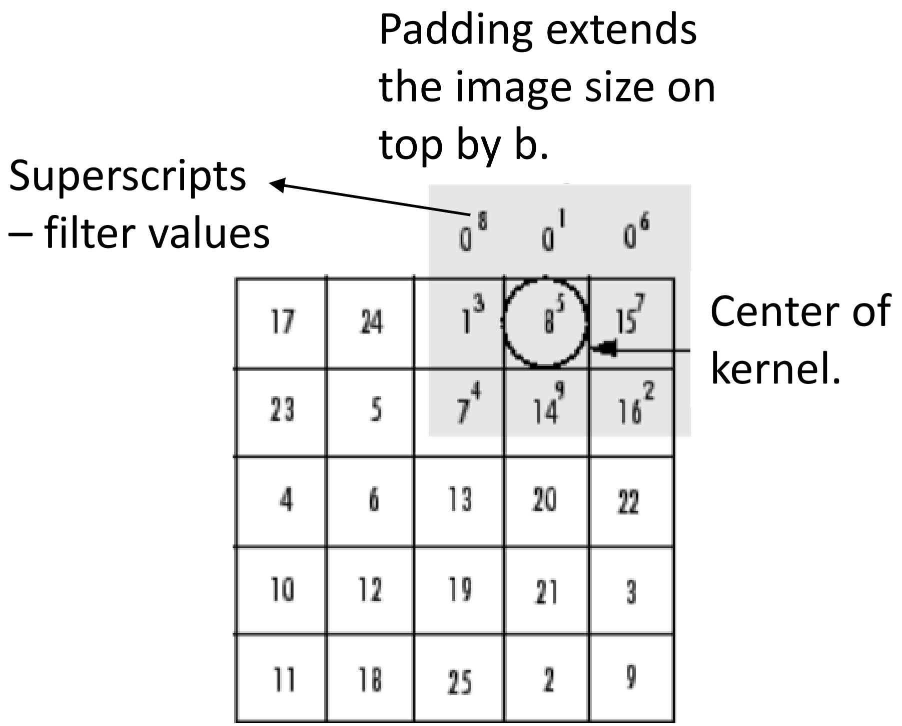
- 1D $$g(x) = \sum_{s=-a}^{a} w(s) f(x+s)$$

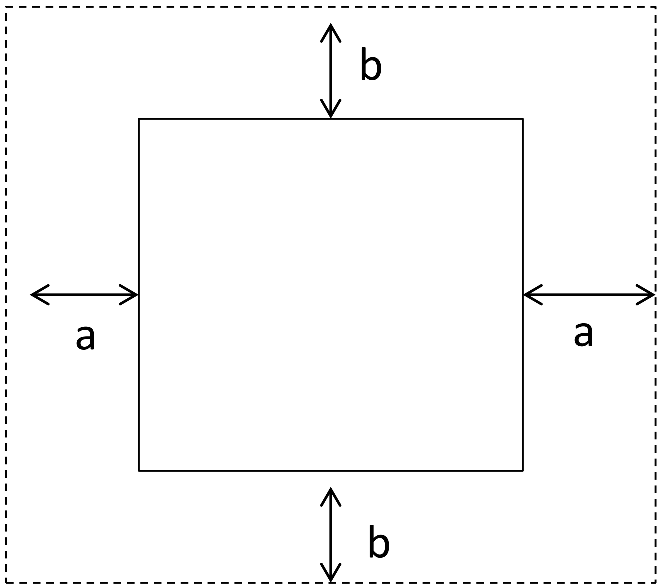
- 2D $$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$w(s,t) = \begin{matrix} w(-a,-b) & \cdots & & \cdots & w(a,-b) \\ \vdots & & & & \vdots \\ & \cdots & w(0,0) & \cdots & \\ \vdots & & & & \vdots \\ w(-a,b) & \cdots & & \cdots & w(a,b) \end{matrix}$$

Correlation: Technical Details

– Boundary conditions – Pad image with amount (a,b) for a filter of size (2a+1,2b+1).



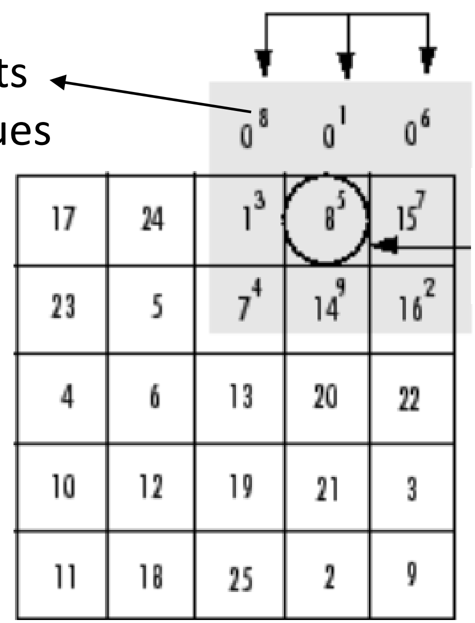


Correlation: Technical Details

- Boundary conditions – zero padding, replication of boundary pixels

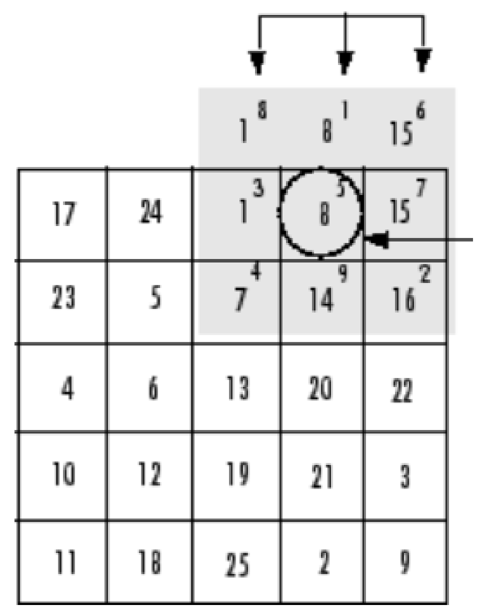
Outside pixels are assumed to be 0's.

Superscripts – filter values



Center of kernel.

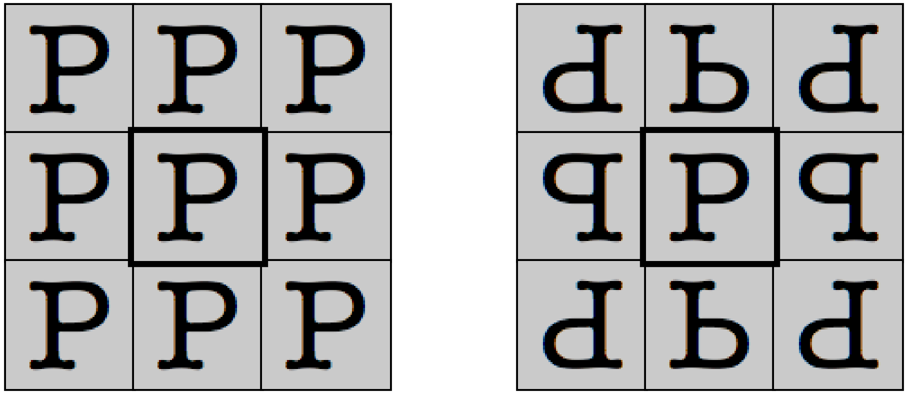
The pixel values are replicated from boundary pixels.



Center of kernel.

Correlation: Technical Details

- Boundary conditions
  - Pad image with amount (a,b)
    - Constant value or repeat edge values
  - Cyclical boundary conditions
    - Wrap or mirroring



Correlation: Technical Details

- Boundaries
  - Can also modify kernel – no long correlation
- For analysis
  - Image domains infinite
  - Data compact (goes to zero far away from origin)

$$g(x,y) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w(s,t) f(x+s, y+t)$$

Correlation: Properties

- Shift invariant – discrete equivalent of time invariant system - If $x(t)$ gives $y(t)$, then $x(t+t_0)$ gives y$(t+t_0)$.

$$g = w \circ f$$ New notation for correlation

$$w \circ f(x-x_0, y-y_0) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w(s,t) f(x-x_0+s, y-y_0+t) = g(x-x_0, y-y_0)$$

- Linear $w \circ (\alpha e + \beta f) = \alpha w \circ e + \beta w \circ f$

$$C_{wf} = w \circ f$$ Compact notation

Cross Correlation Continuous Case

- f, w must be "integrable"
  - Must die off fast enough so that integral is finite

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(s,t)f(x+s,y+t)dsdt$$

- Same properties as discrete case
  - Linear
  - Shift invariant

Filters: Considerations

- Normalize
  - Sums to one
  - Sums to zero (some cases, later)
- Symmetry
  - Left, right, up, down
  - Rotational
- Special case: auto correlation

$$C_{ff} = f \circ f$$

Examples 1

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$

$$1/9 * \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Examples 2

$$\frac{1}{9} * \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$\frac{1}{25} * \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix}$$

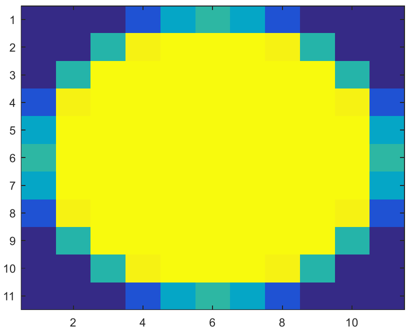
Smoothing and Noise

Noisy image

5x5 box filter

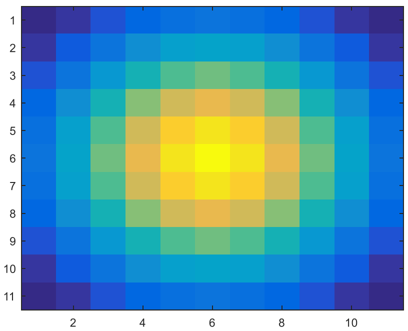
Other Filters

- Disk
  - Circularly symmetric, jagged in discrete case



- Gaussians
  - Circularly symmetric, smooth for large enough std-dev.
  - Must normalize in order to sum to one

$$\frac{1}{2\pi\sigma}\exp -\frac{\left(x^2+y^2\right)}{2\sigma^2}$$



- Derivatives – discrete/finite differences
  - Operators

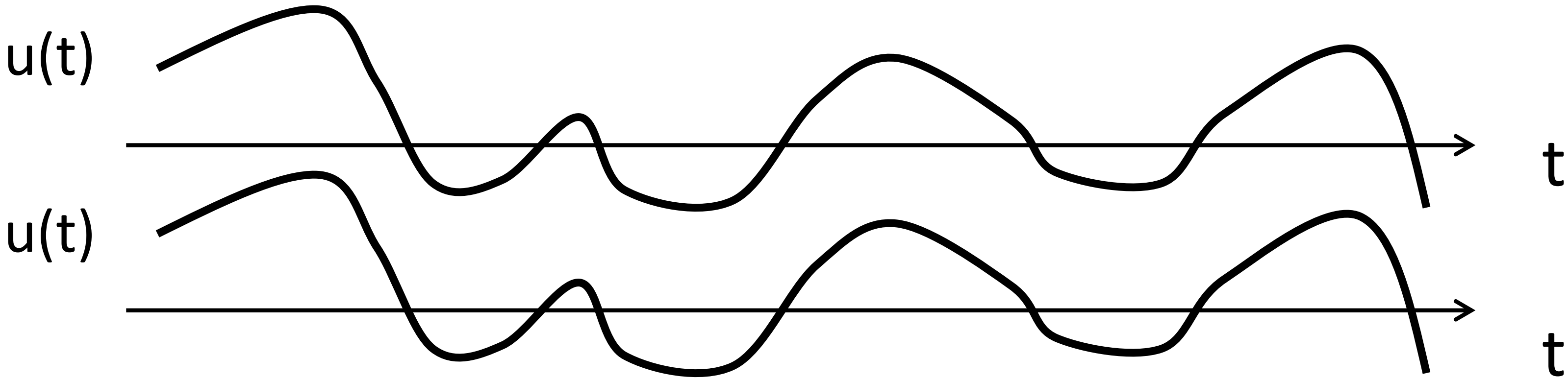
Pattern Matching/Detection

- The optimal (highest) response from a filter is the autocorrelation evaluated at position zero

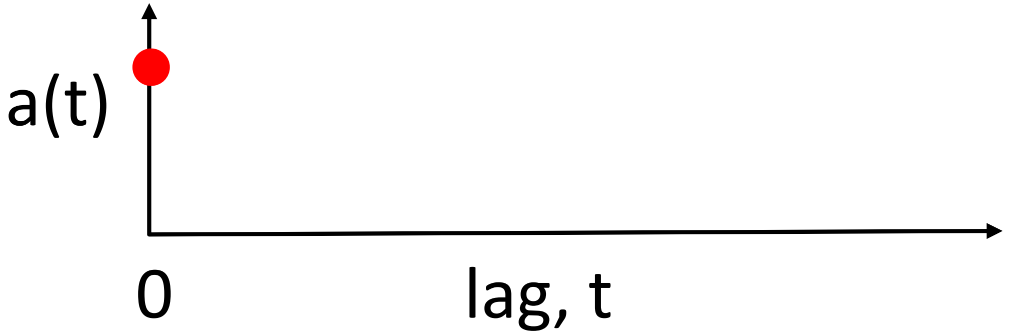
$$\max_{\bar{x}} C_{ff}(\bar{x}) = C_{ff}(0) = \int f(\bar{s}) f(\bar{s}) d\bar{s}$$

- A filter responds best when it matches a pattern that looks like itself
- Strategy
  - Detect objects in images by correlation with "matched" filter

Measure of correlation in time series at different lags

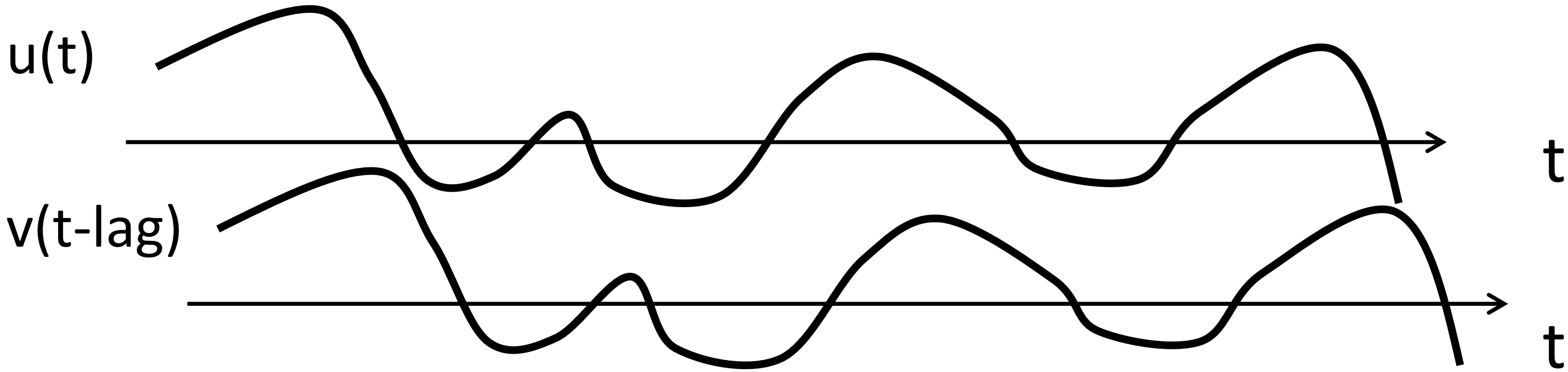


No lag, multiply and sum area

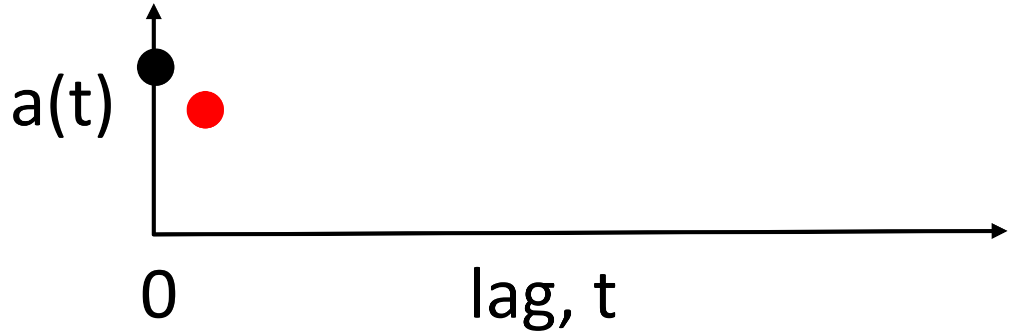


Source: http://www.ldeo.columbia.edu/users/menke/edawm/index.htm

Measure of correlation in time series at different lags

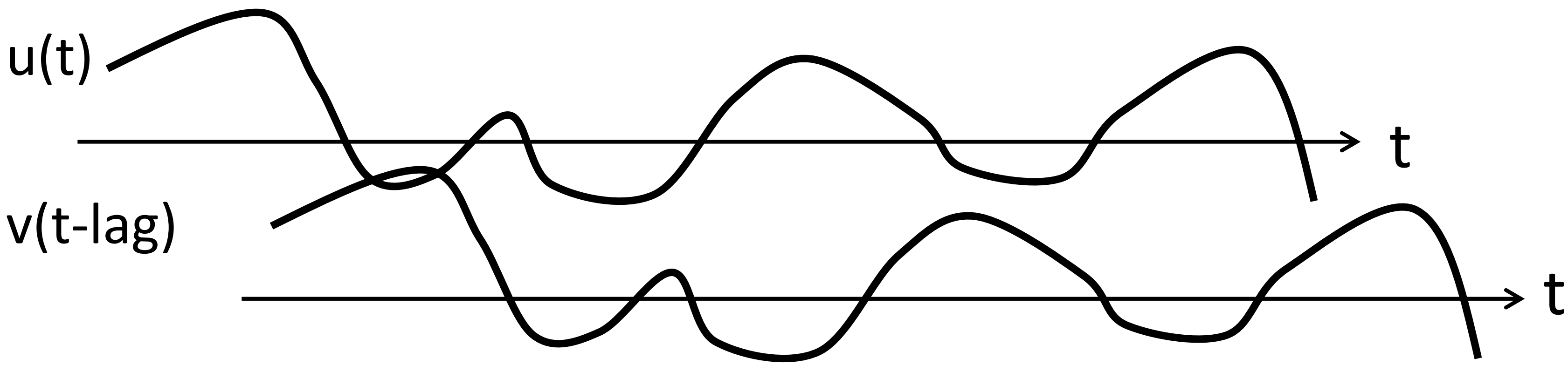


Small lag, multiply and sum area

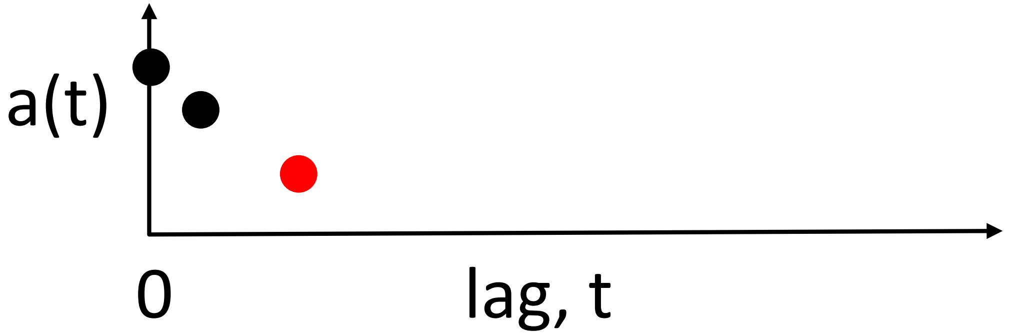


Source: http://www.ldeo.columbia.edu/users/menke/edawm/index.htm

Measure of correlation in time series at different lags

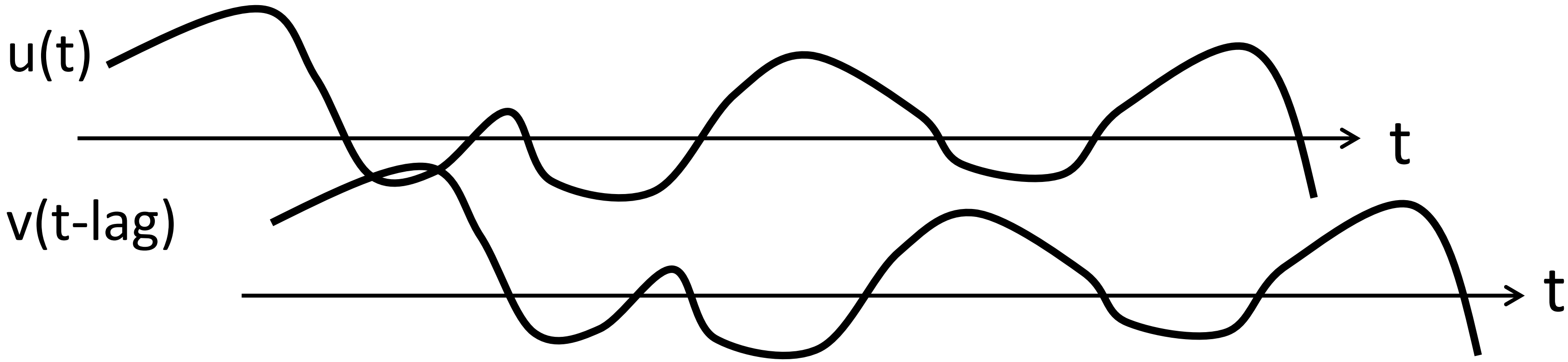


Large lag, multiply and sum area

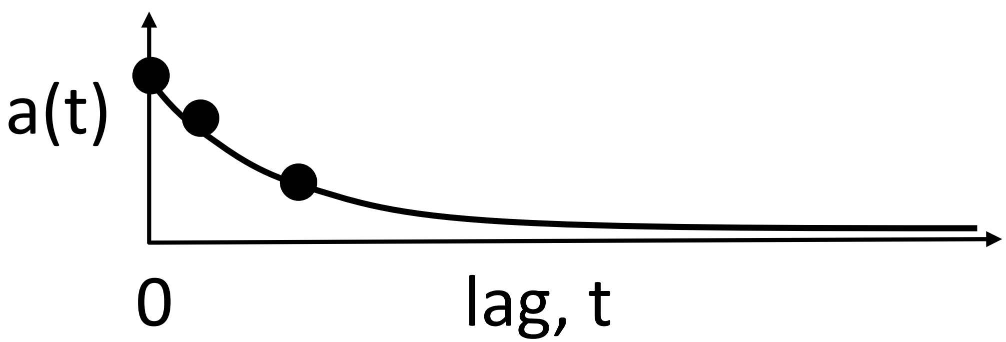


Source: http://www.ldeo.columbia.edu/users/menke/edawm/index.htm

Measure of correlation in time series at different lags



lag, multiply and sum area



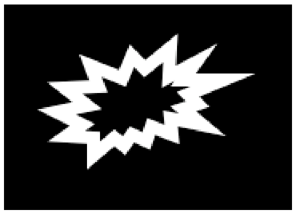
Source: http://www.ldeo.columbia.edu/users/menke/edawm/index.htm

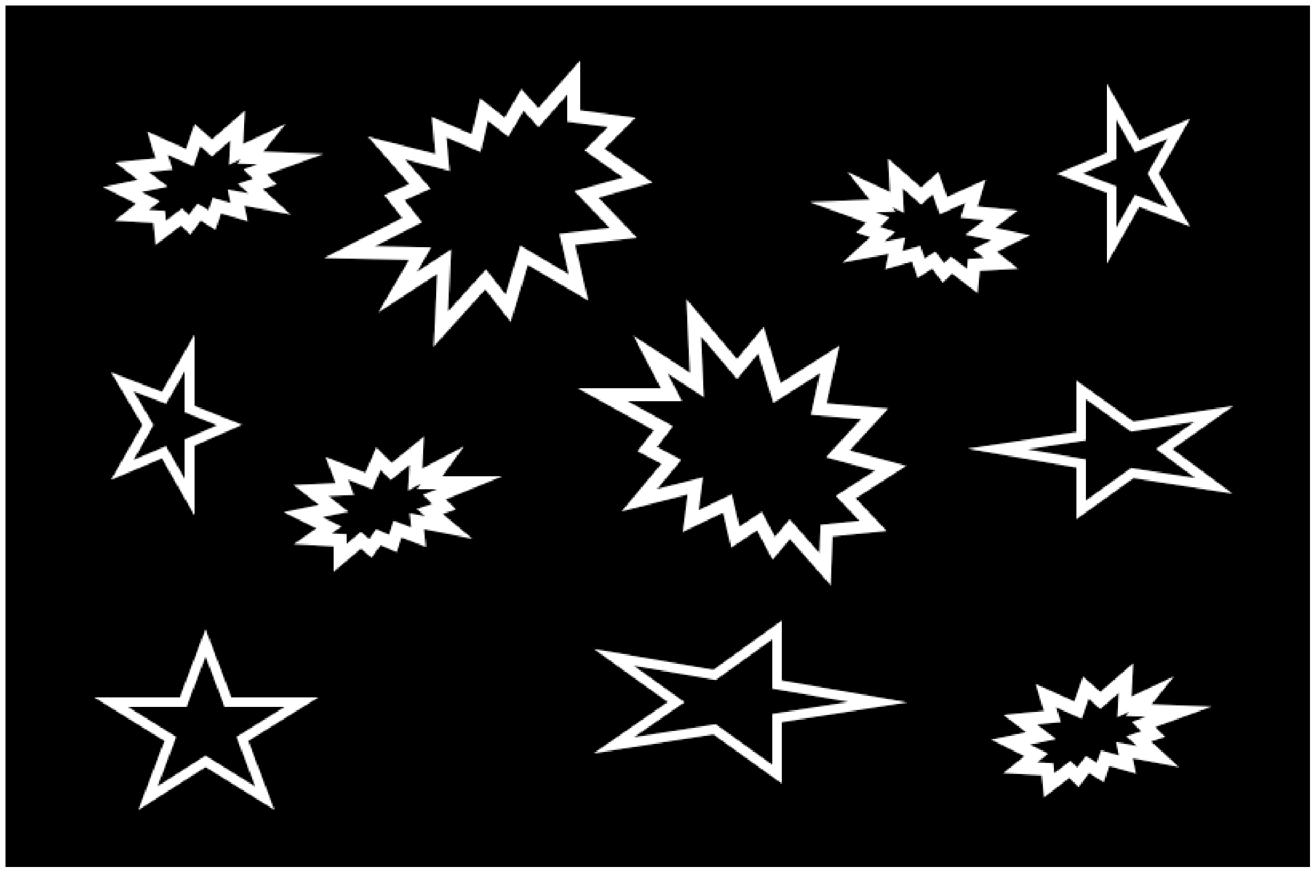
Reasoning

- Dot product of two normalized vectors a and b is maximum when both the vectors are equal.

$$a.b = |a||b|\cos(\theta)$$

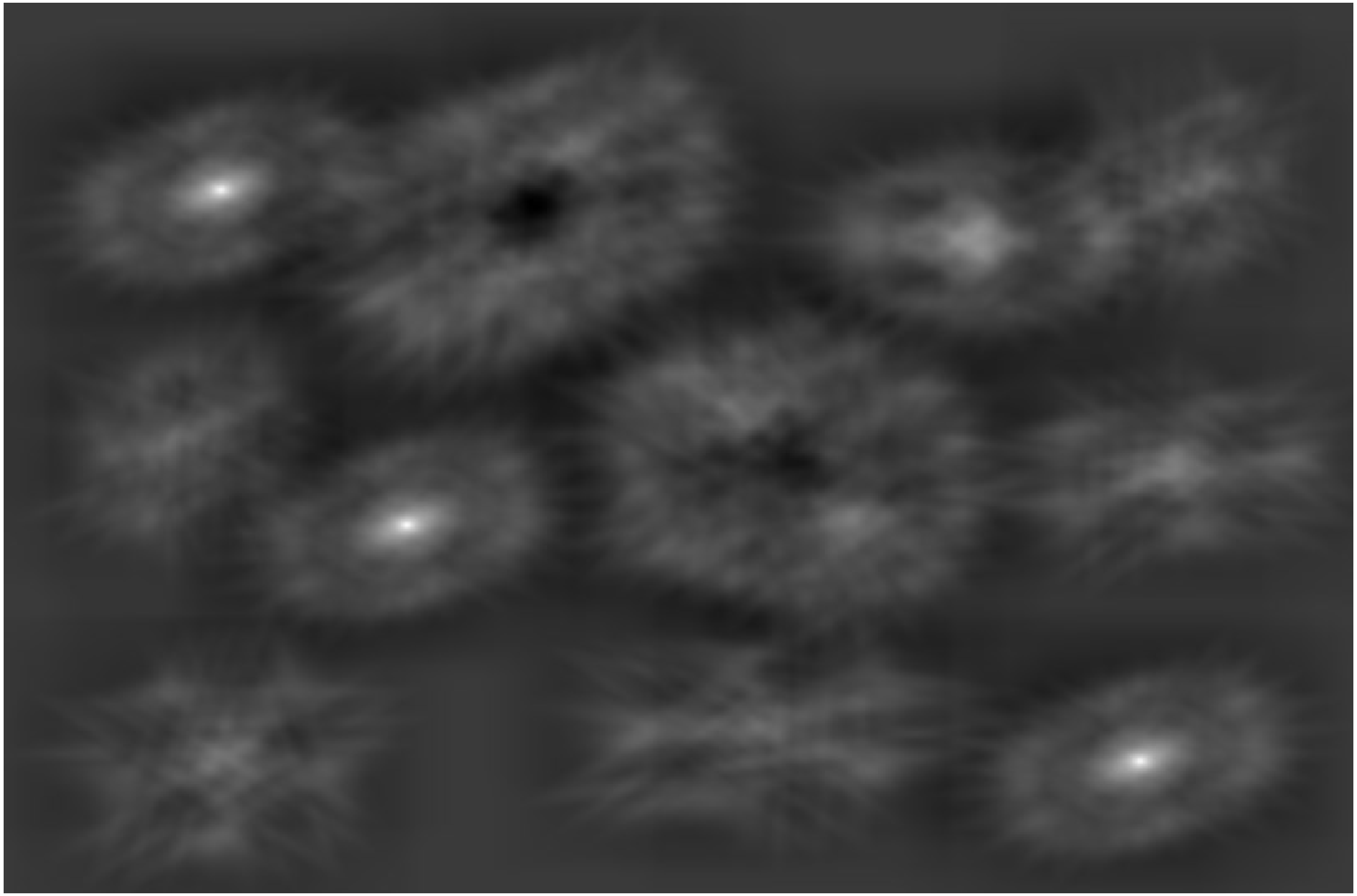
Match Filter Example





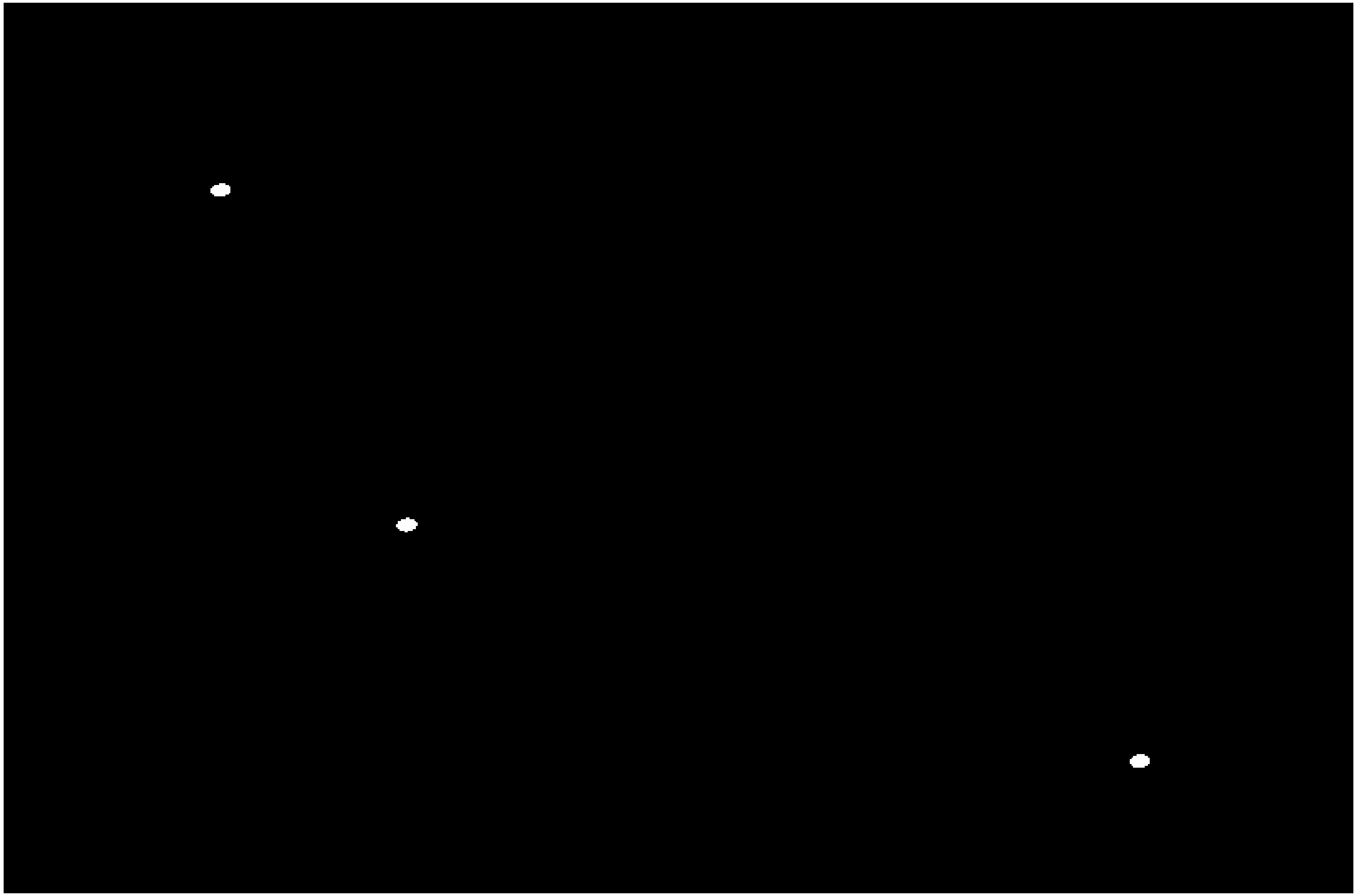
Trick: make sure kernel sums to zero

Match Filter Example



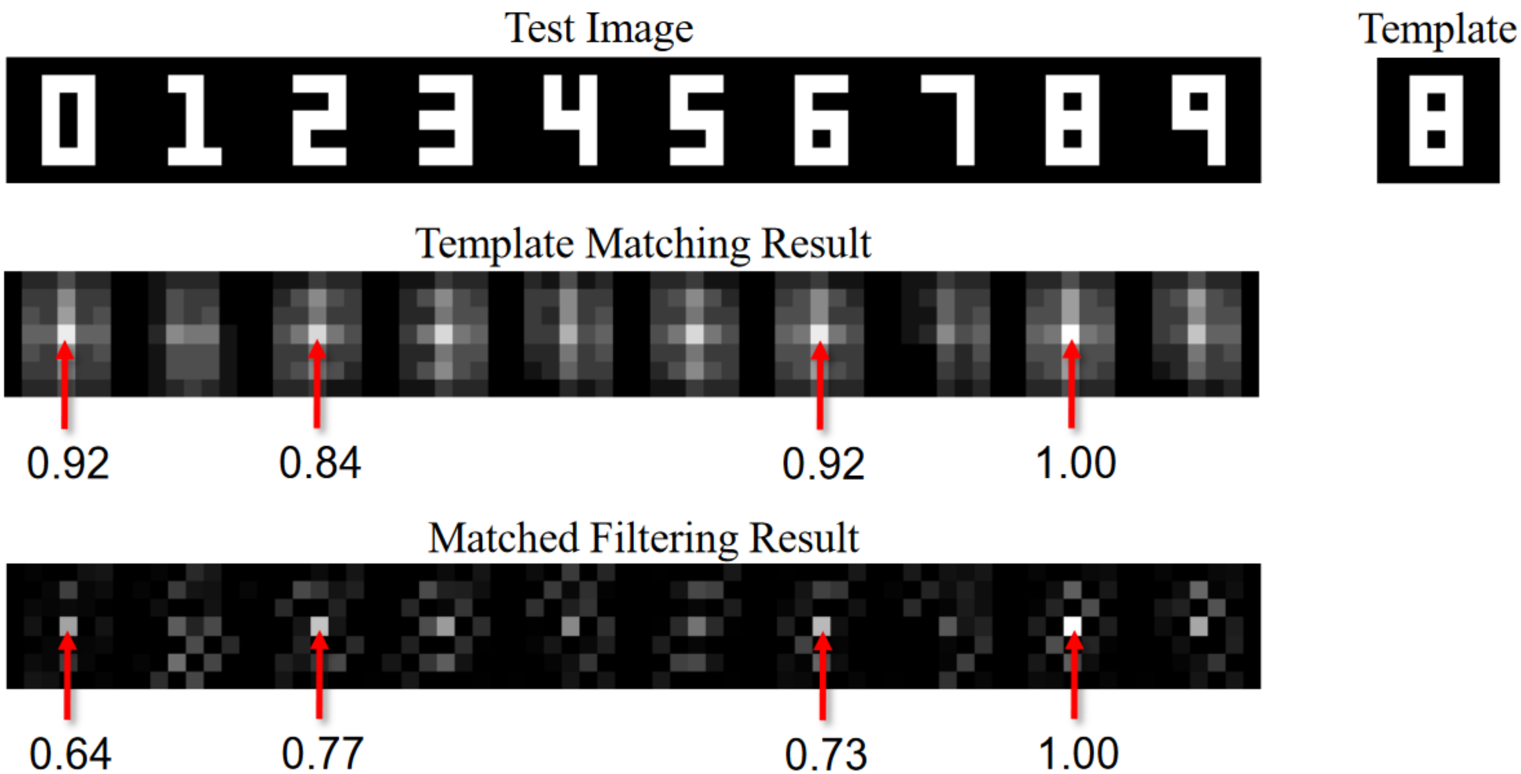
[Responses from match filtering]

Match Filter Example



[Peaks from match filtering]

Match Filter Example



[Source: Bernd Girod]

Derivatives: Finite Differences

$$\frac{\partial f}{\partial x} \approx \frac{f(x+1,y) - f(x-1,y)}{2}$$

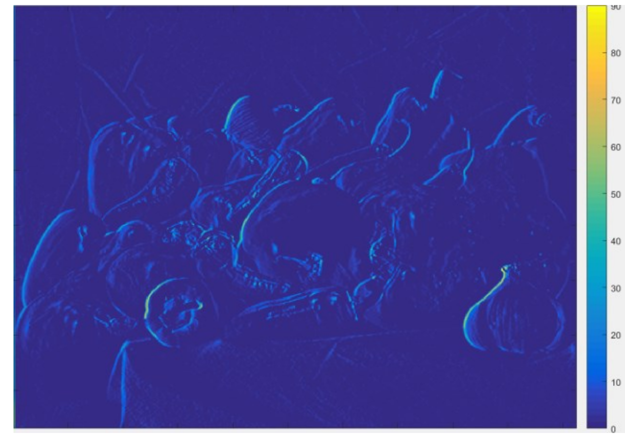
$$\frac{\partial f}{\partial x} \approx \frac{f(x,y+1) - f(x,y-1)}{2}$$

$$\frac{\partial f}{\partial x} \approx w_{dx} \circ f \qquad w_{dx} = \begin{array}{|c|c|c|}\hline -\frac{1}{2} & 0 & \frac{1}{2} \\ \hline \end{array}$$

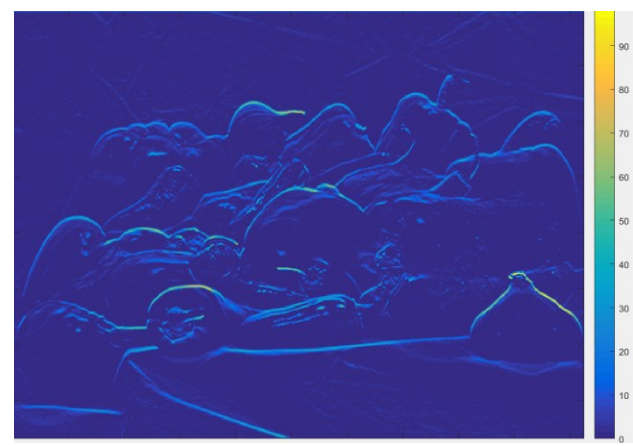
$$\frac{\partial f}{\partial y} \approx w_{dy} \circ f \qquad w_{dy} = \begin{array}{|c|}\hline -\frac{1}{2} \\ \hline 0 \\ \hline \frac{1}{2} \\ \hline \end{array}$$

Derivative Example

$$\begin{matrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{matrix}$$



$$\begin{matrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}$$



Other filters

- Prewitt

-1	0	+1
-1	0	+1
-1	0	+1

-1	-1	-1
0	0	0
+1	+1	+1

- Sobel

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

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

Convolution

- Discrete

$$g(x,y) = w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

- Continuous

$$g(x,y) = w(x,y) * f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(s,t) f(x-s, y-t) ds dt$$

- Same as cross correlation with kernel transposed around each axis

Convolution = Correlation

- The two operations (correlation and convolution) are the same if the kernel is symmetric about axes

$$g = w \circ f = w^* * f$$

w^* reflection of w

Convolution: Properties

- Shift invariant, linear
- Commutative

$$f * g = g * f$$

- Associative

$$f * (g * h) = (f * g) * h$$

- Others (discussed later):
  - Derivatives, convolution theorem, spectrum…

Associativity

Example: Correlation

[-1 3 -3 1]

[1 -2 1]

[-1 1] [1 -1] [-1 1]

[1 -2 1]

[1 -3 3 -1]

Associativity

Example: Convolution

[1 -3 3 -1]

[-1 2 -1]

[-1 1] [1 -1] [-1 1]

[-1 2 -1]

[1 -3 3 -1]

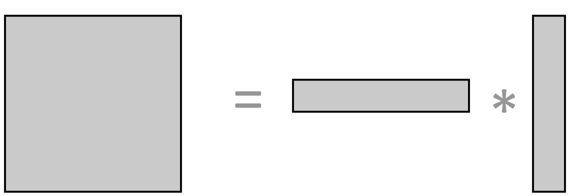
Computing Convolution

- Compute time
  - MxM mask
  - NxN image

  $\Bigg\}$ $O(M^2N^2)$ "for" loops are nested 4 deep

- Special case: separable

$$w = \overbrace{w_x * w_y}^{\text{Two 1D kernels}}$$

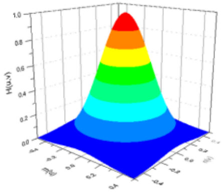


$$w * f = \underbrace{(w_x * w_y) * f}_{O(M^2N^2)} = \underbrace{w_x * (w_y * f)}_{O(MN^2)}$$

Separable Kernels

- Examples
  - Box/rectangle
  - Bilinear interpolation
  - Combinations of partial derivatives
    - d2f/dxdy
  - Gaussian
    - Only filter that is both circularly symmetric and separable

- Counter examples
  - Disk
  - Cone



  - Pyramid

Examples of Separable Kernels

Smoothing Filter

$$\frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix} * \frac{1}{3}\begin{bmatrix}1 & 1 & 1\end{bmatrix} = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

Sobel Filter

$$\mathbf{G_x} = \begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix} * A = \begin{bmatrix}1\\2\\1\end{bmatrix} * \begin{bmatrix}+1 & 0 & -1\end{bmatrix} * A$$

Examples of Separable filter

Prewitt Filter

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} = ?$$

$$\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} = ?$$

2nd order derivatives

Second derivatives:

$$\frac{\partial f}{\partial x} = [-1\ 1] * f = f(x) - f(x-1)$$

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial x} = [-1\ 1] * [-1\ 1] * f = [1\ \ -2\ \ \ 1] * f$$

$$= f(x+1) - 2f(x) + f(x-1)$$

$$\frac{\partial f}{\partial y} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} * f = f(y) - f(y-1)$$

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial x} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 \\ 1 \end{bmatrix} * f = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} * f$$

$$= f(x+1) - 2f(x) + f(x-1)$$

Laplacian

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$= f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$+ f(x, y+1) + f(x, y-1) - 2f(x, y)$$

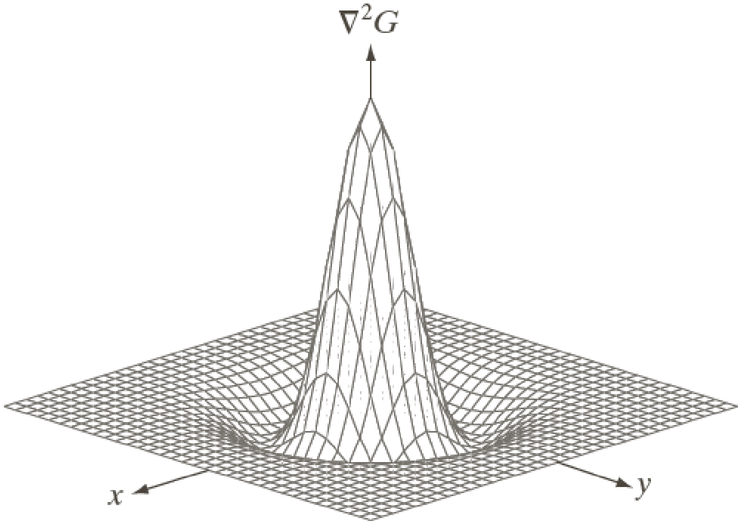
$$= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

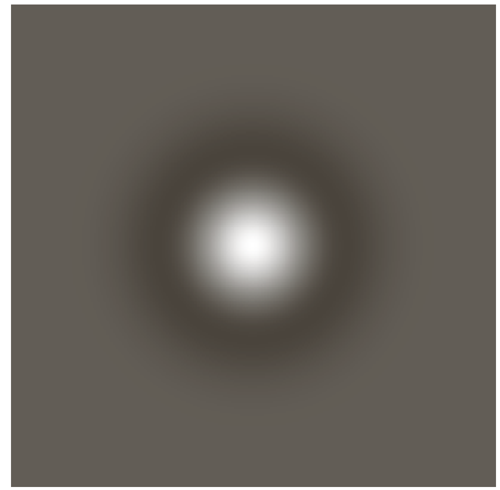
0	**+1**	**0**
+1	**-4**	**+1**
0	**+1**	**0**

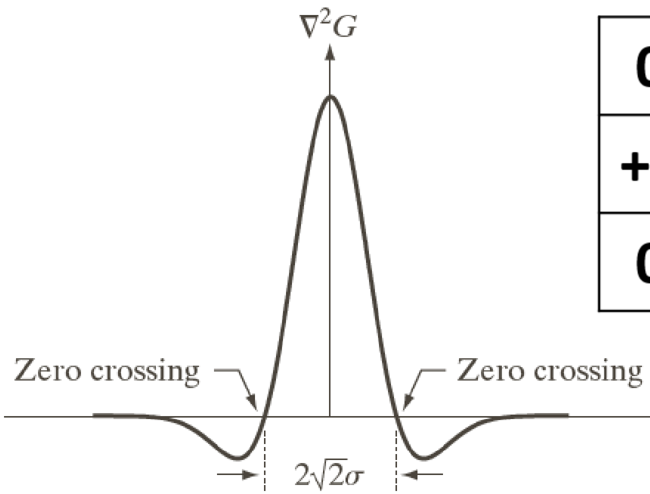
Laplacian of a Gaussian

- We take an image and blur it a little using Gaussian
- function.
- Calculate the second order derivatives or the Laplacian.
- locates edges and corners that are good for detecting keypoints.
- Computation of the second order derivative is also extremely sensitive to noise, and the blurring helps.

Laplacian of Gaussian







0	**+1**	**0**
+1	**-4**	**+1**
0	**+1**	**0**

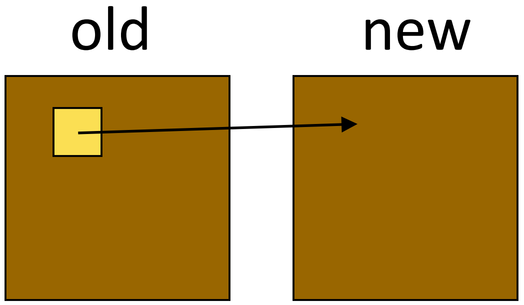
$$* \quad \frac{1}{2\pi\sigma}\exp\left[-\frac{\left(x^2+y^2\right)}{2\sigma^2}\right]$$

Nonlinear Methods For Filtering

- Median filtering
- Bilateral filtering
- Neighborhood statistics and nonlocal filtering

Median Filtering

- For each neighborhood in image
  - Sliding window
  - Usually odd size (symmetric) 5x5, 7x7,…
- Sort the greyscale values
- Set the center pixel to the median
- Important:
  - Separate input and output buffers
  - All statistics on the original image

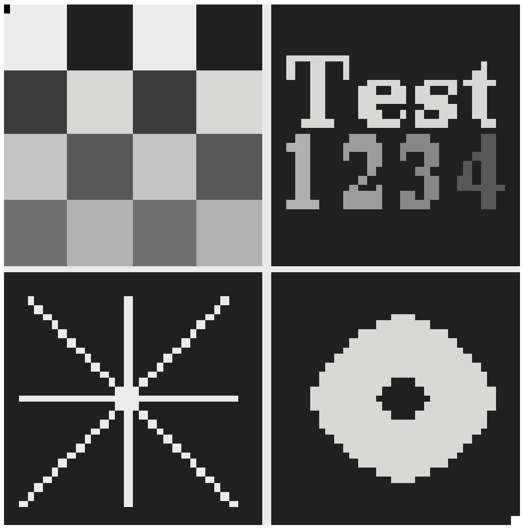


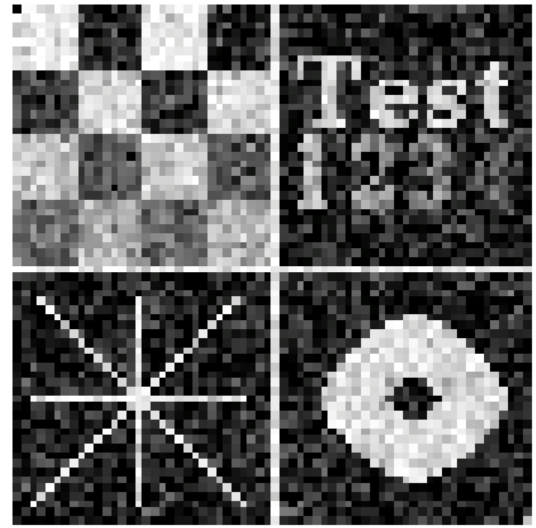
Median Filter

- Issues
  - Boundaries
    - Compute on pixels that fall within window
  - Computational efficiency
    - What is the best algorithm?
- Properties
  - Removes outliers (replacement noise – salt and pepper)
  - Window size controls size of structures
  - Preserves straight edges, but rounds corners and features

Median vs Gaussian

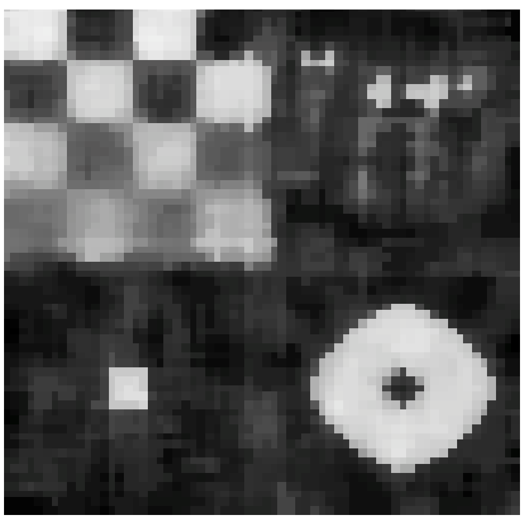
Original

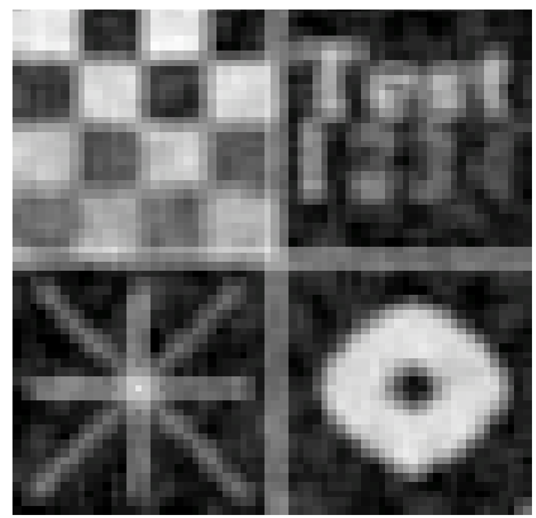




+ Gaussian Noise

3x3 Median

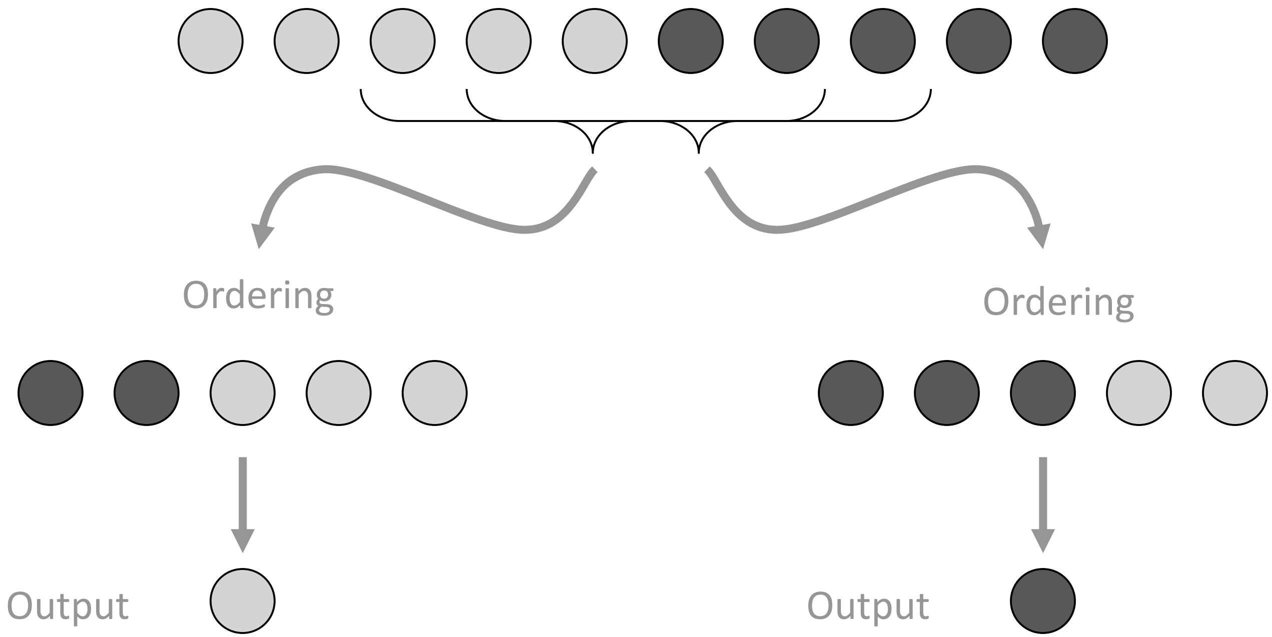




3x3 Box

Median Filtering

- Image model: piecewise constant (flat)



Order Statistics

- Median is special case of order-statistics filters
- Instead of weights based on neighborhoods, weights are based on ordering of data

Neighborhood

$$X_1, X_2, \ldots, X_N$$

Ordering

$$X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(N)}$$

Filter

$$F(X_1, X_2, \ldots, X_N) = \alpha_1 X_{(1)} + \alpha_2 X_{(2)} + \ldots + \alpha_N X_{(N)}$$

Neighborhood average (box)

$$\alpha_i = 1/N$$

Median filter

$$\alpha_i = \begin{cases} 1 & i = (N+1)/2 \\ 0 & \text{otherwise} \end{cases}$$

Trimmed average (outlier removal)

$$\alpha_i = \begin{cases} 1/M & (N-M+1)/2 \leq i \leq (N+M+1)/2 \\ 0 & \text{otherwise} \end{cases}$$

Piecewise Flat Image Models

- Image piecewise flat -> average only within similar regions
- Problem: don’t know region boundaries

Piecewise-Flat Image Models

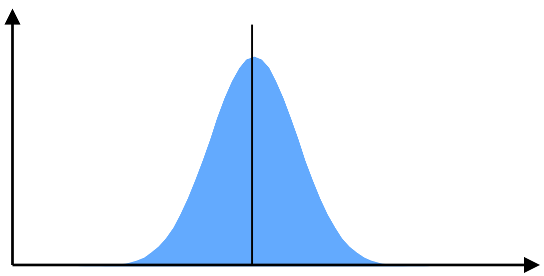
- Assign probabilities to other pixels in the image belonging to the same region
- Two considerations
  - Distance: far away pixels are less likely to be same region
  - Intensity: pixels with different intensities are less likely to be same region

Piecewise-Flat Images and Pixel Averaging

Distance (kernel/pdf)

$$G(\mathbf{x}_i - \mathbf{x}_j)$$

Prob pixel belongs to same region as i

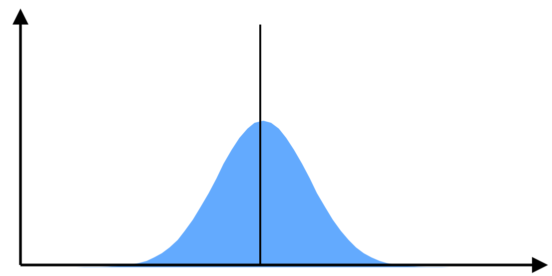


position

Distance (pdf)

$$H(f_i - f_j)$$

Prob pixel belongs to same region as i



intensity

Bilateral Filter

- Neighborhood – sliding window
- Weight contribution of neighbors according to:

$$f_i \leftarrow k_i^{-1} \sum_{j \in N} f_j G(\mathbf{x}_i - \mathbf{x}_j) H(f_i - f_j)$$

$$k_i = \sum_{j \in N} G(\mathbf{x}_i - \mathbf{x}_j) H(f_i - f_j)$$

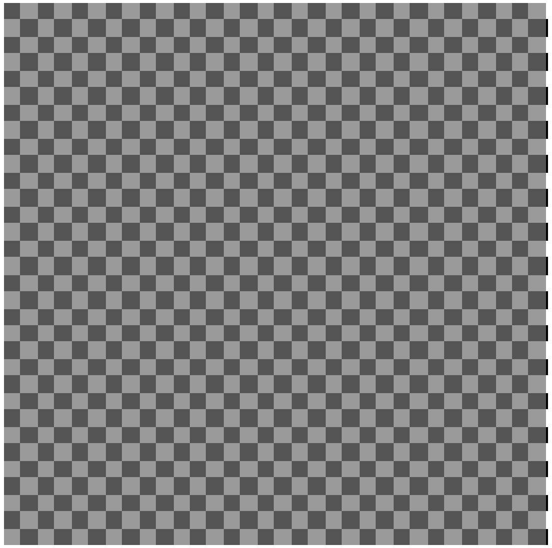
- G is a Gaussian (or lowpass), as is H, N is neighborhood,
  - Often use G(rij) where rij is distance between pixels
  - Update must be normalized for the samples used in this (particular) summation
- Spatial Gaussian with extra weighting for intensity
  - Weighted average in neighborhood with downgrading of intensity outliers

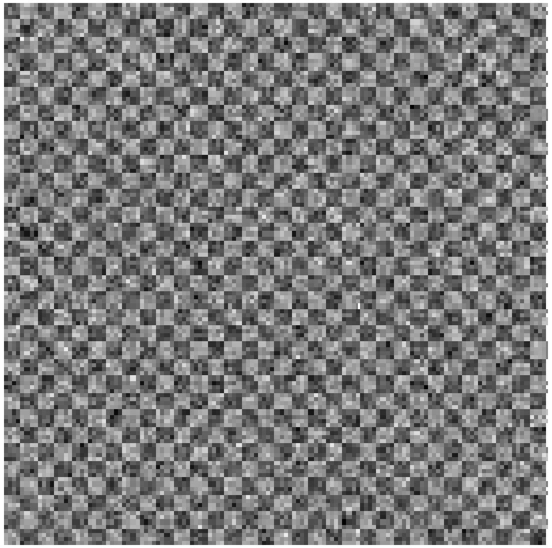
Bilateral Filtering

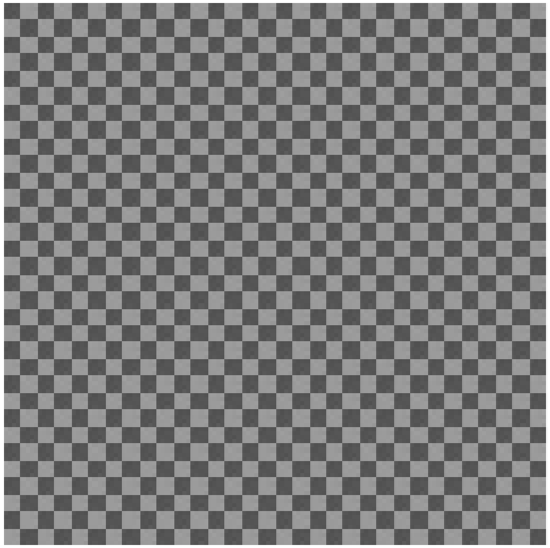
Gaussian Blurring

Bilateral

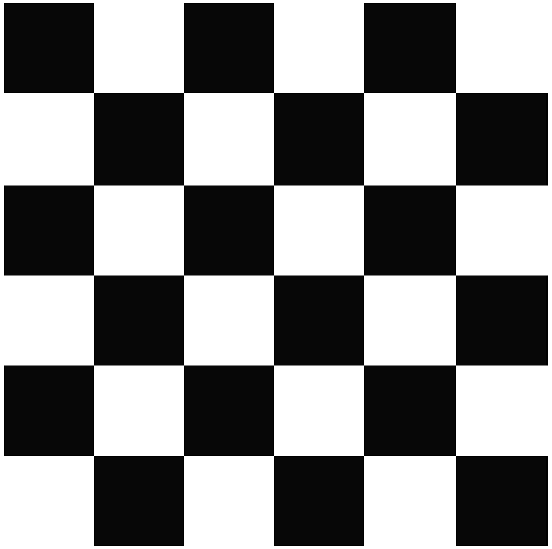
Results


Original

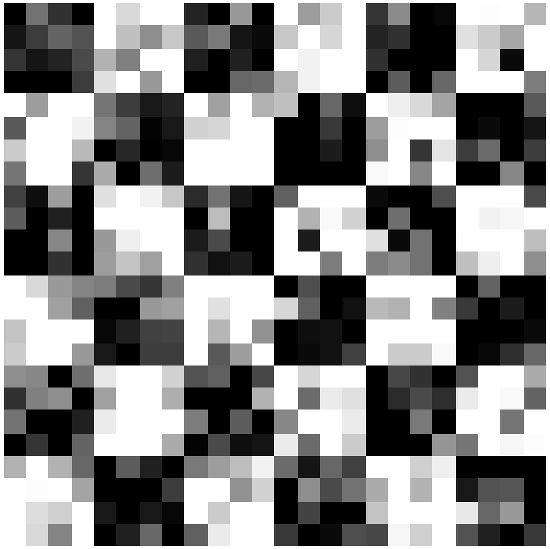

Noisy


Filtered

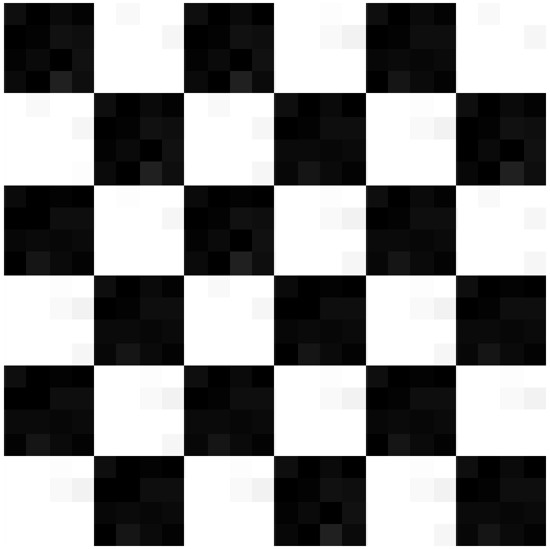
Checkerboard With Noise



Original

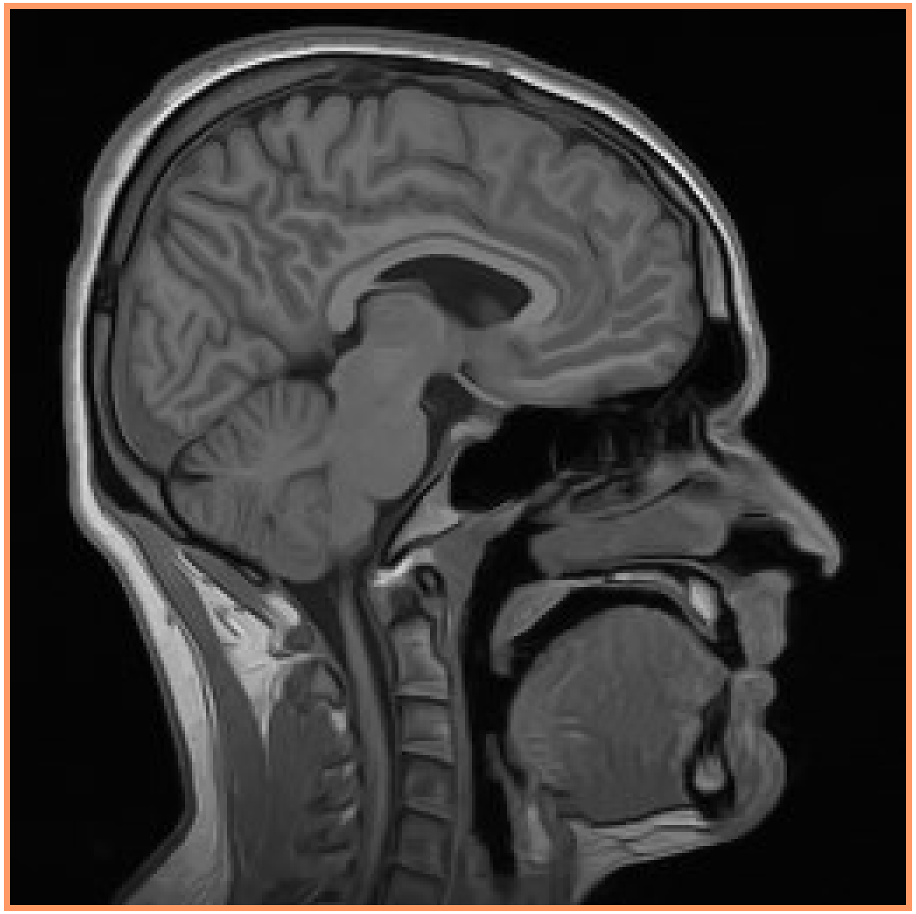


Noisy

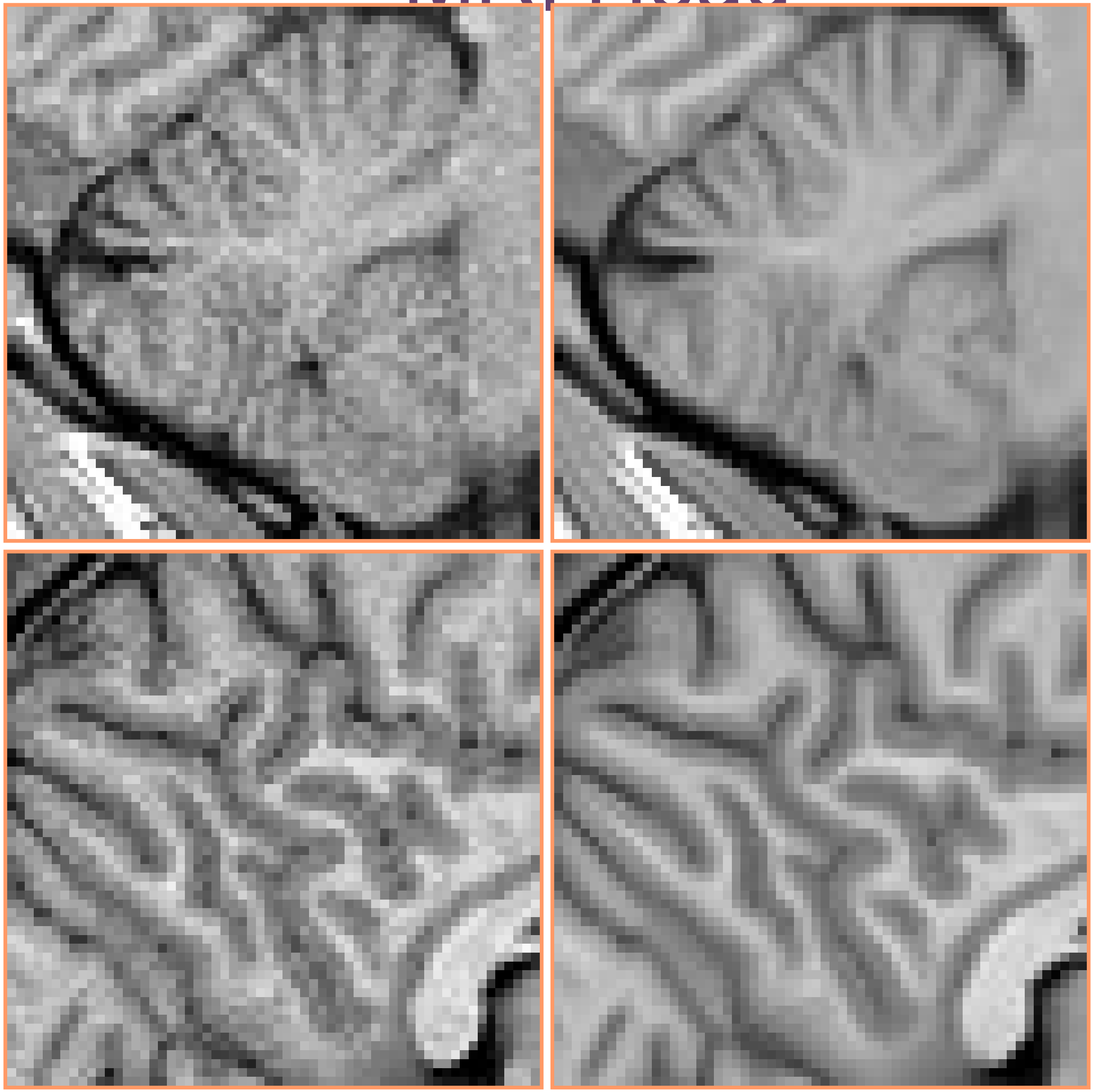


Filtered

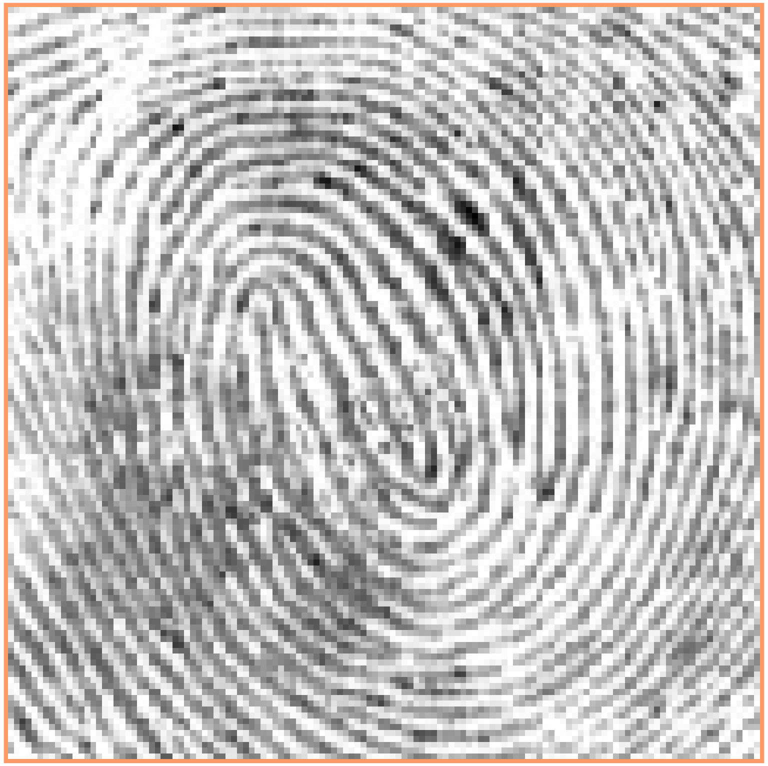
MRI Head



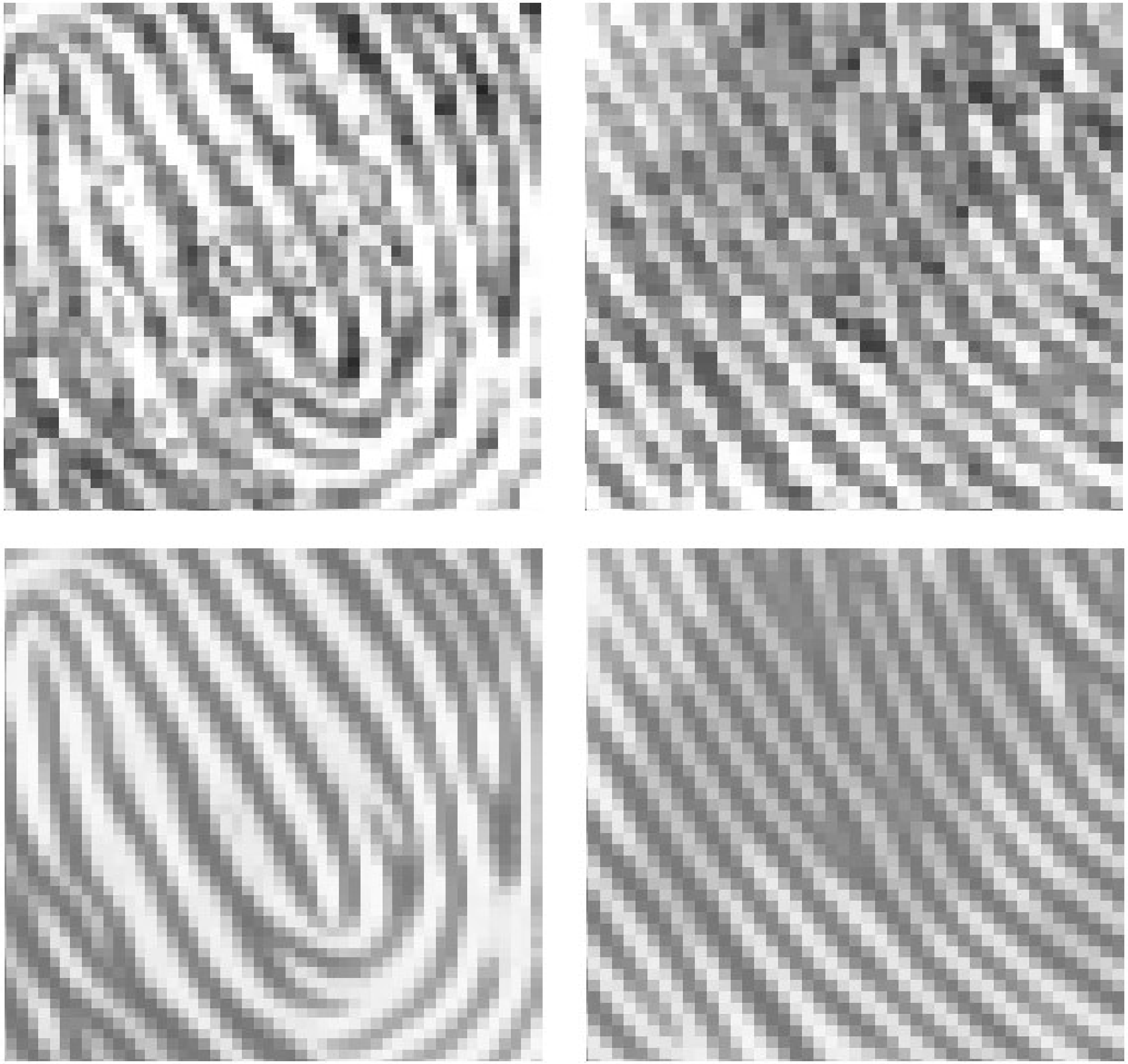
MRI Head



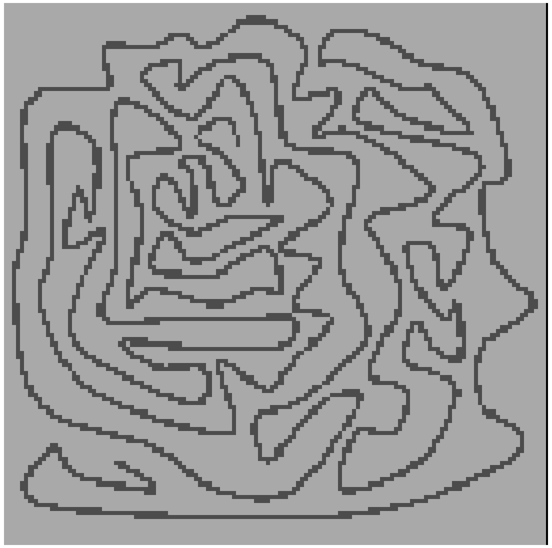
Fingerprint



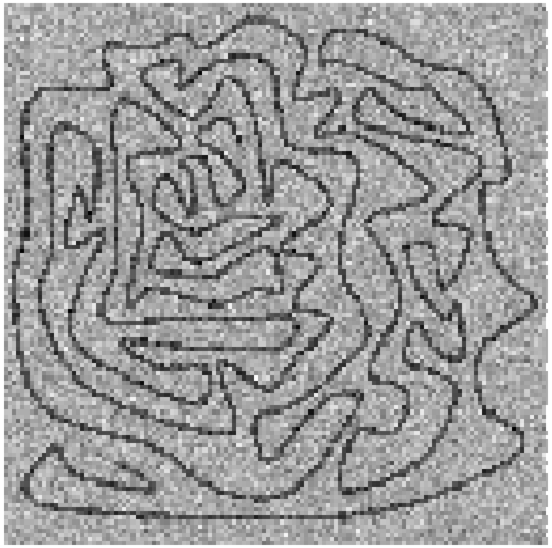
Fingerprint



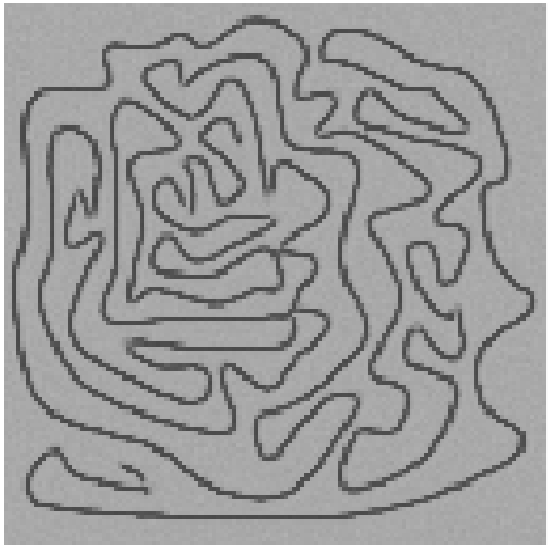
Results



Original

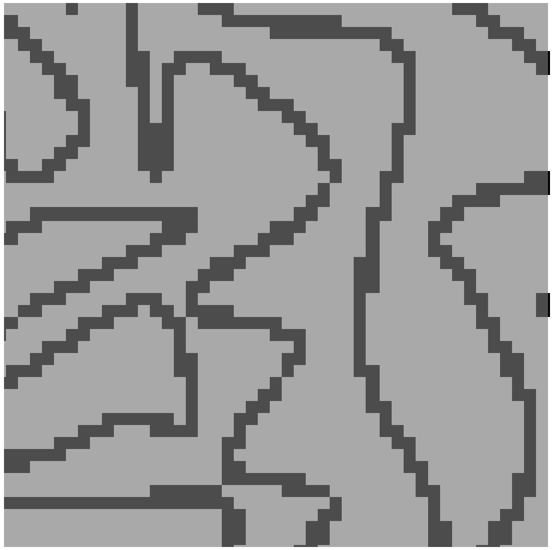


Noisy

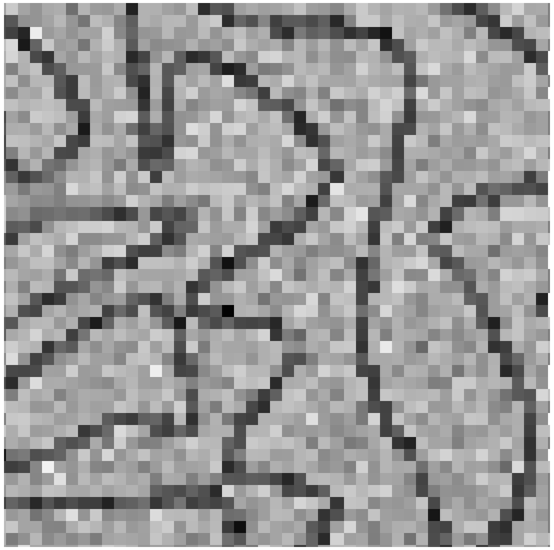


Filtered

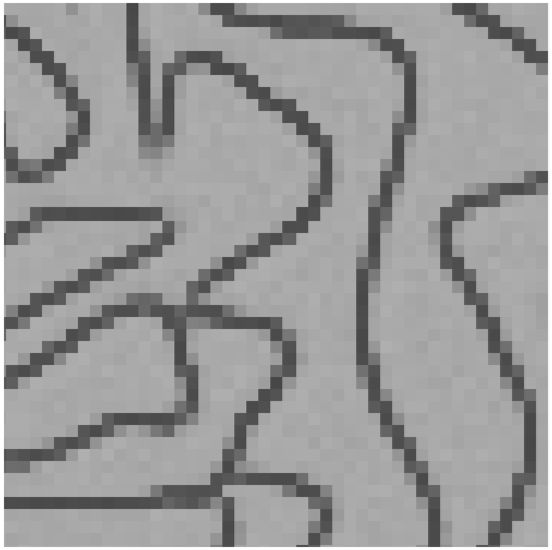
Results



Original

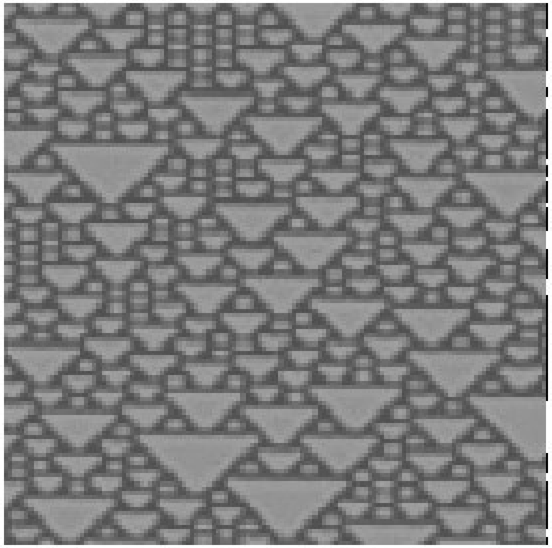


Noisy

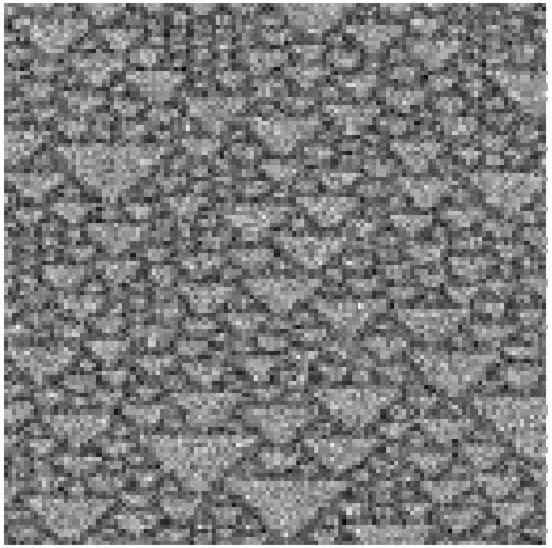


Filtered

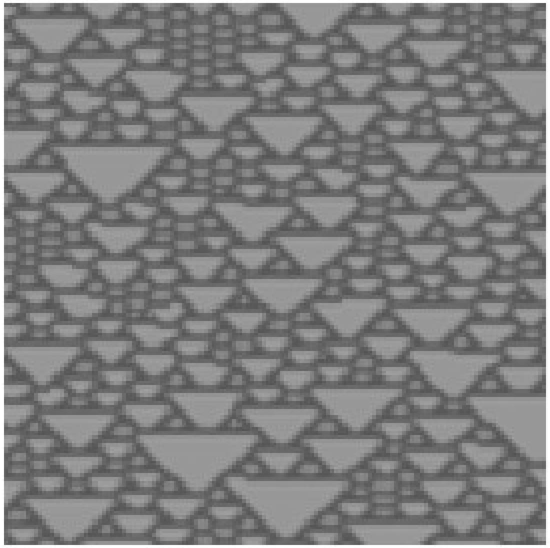
Results



Original

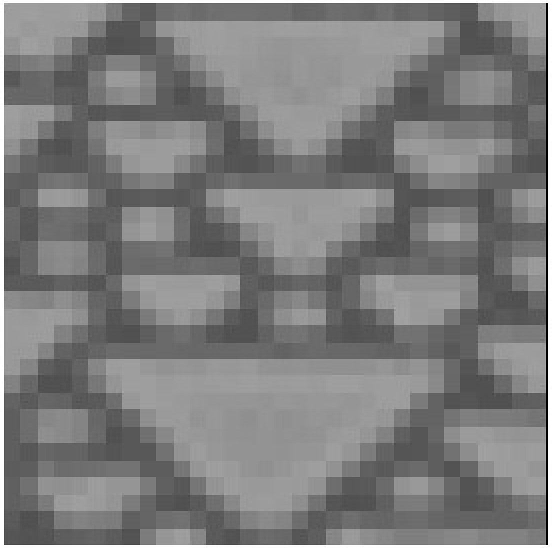


Noisy

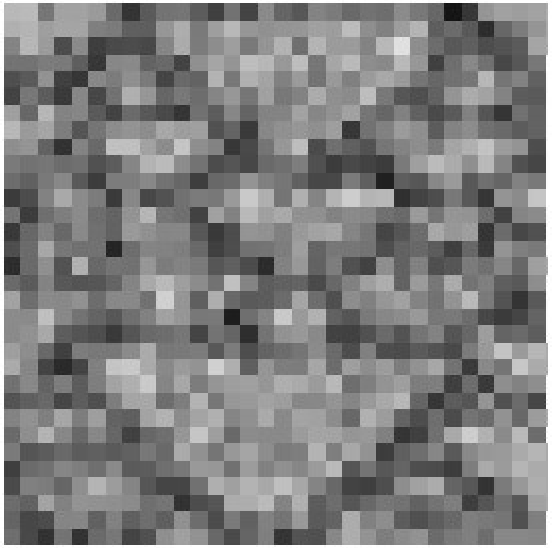


Filtered

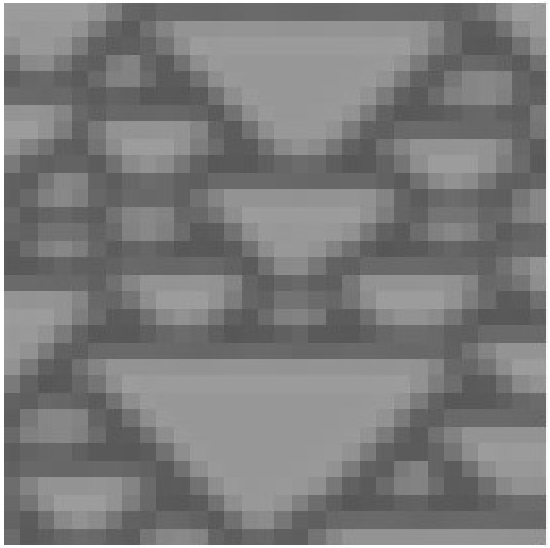
Fractal



Original



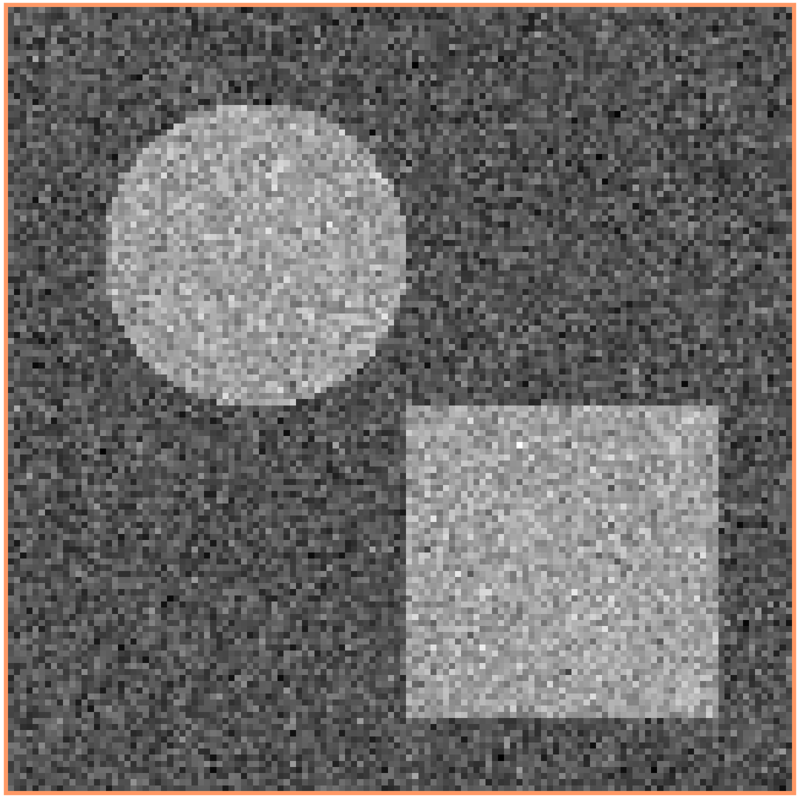
Noisy



Filtered

Piecewise Constant

- Several 10s of Iterations
- Tends to obliterate rare events



Texture, Structure