CDTM Drones Lecture 2016

Basics of Computer Vision
Part 2
Image Processing

Seyed-Ahmad Ahmadi
Dept. Neurology, LMU Klinikum Großhadern

Last year‘s approaches for semi-autonomous drone flight

Outline of this part

To understand fundamentals of image processing...

- Better understand digital images and how to manipulate them...
- Colors and color space conversions
- Intensity distributions and histograms
- Histograms and thresholding
- Binary filters: Morphological Operations
- Linear filtering in spatial domain
- Linear filtering in frequency domain
- To solve a concrete (but simple) computer vision task with easy-to-use tools from Image Processing/Computer Vision
  - E.g. identify a region of a certain color in an image, and to cleanly segment it using some post-processing steps
- Hough line/circle/arbitrary object detection
- Just one more thing... ;)

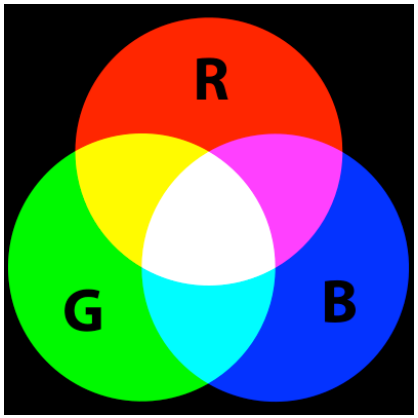
An example algorithm for simple, (semi-)autonomous drone navigation

- Colored ball in front of neutral BG, with noise and/or (similar-colored) clutter

© Simon Levy (Mar 8, 2011)
https://www.youtube.com/watch?v=_3697dtyOz4

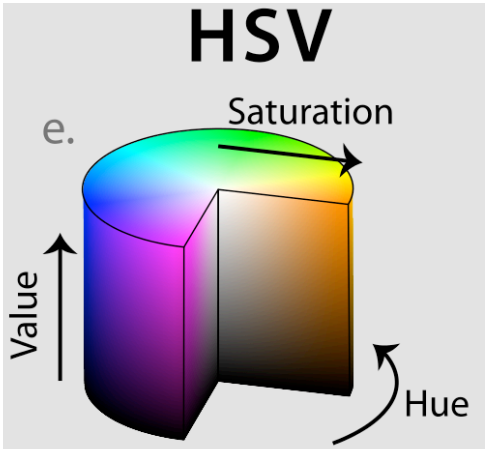
Handling a „colored ball“ - Colorspaces

- RGB/HSV/YCbCr/L*a*b* etc. (>140 in OpenCV)
- RGB:
  - Red-green-blue
  - Monitors/cameras
  - Unintuitive
- HSV:
  - Hue-saturation-value (-brightness)
  - Explicit color channel („hue“)



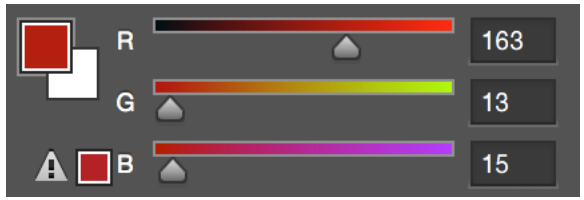
"AdditiveColor" by SharkD at English Wikipedia

"Hsl-hsv models" by Jacob Rus, via Wikimedia Commons

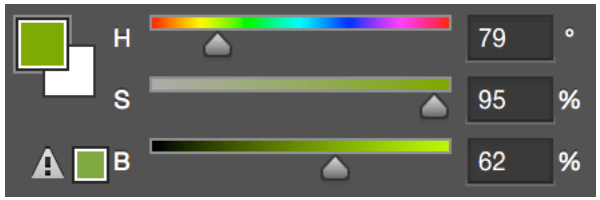


Colorspace in practice

RGB

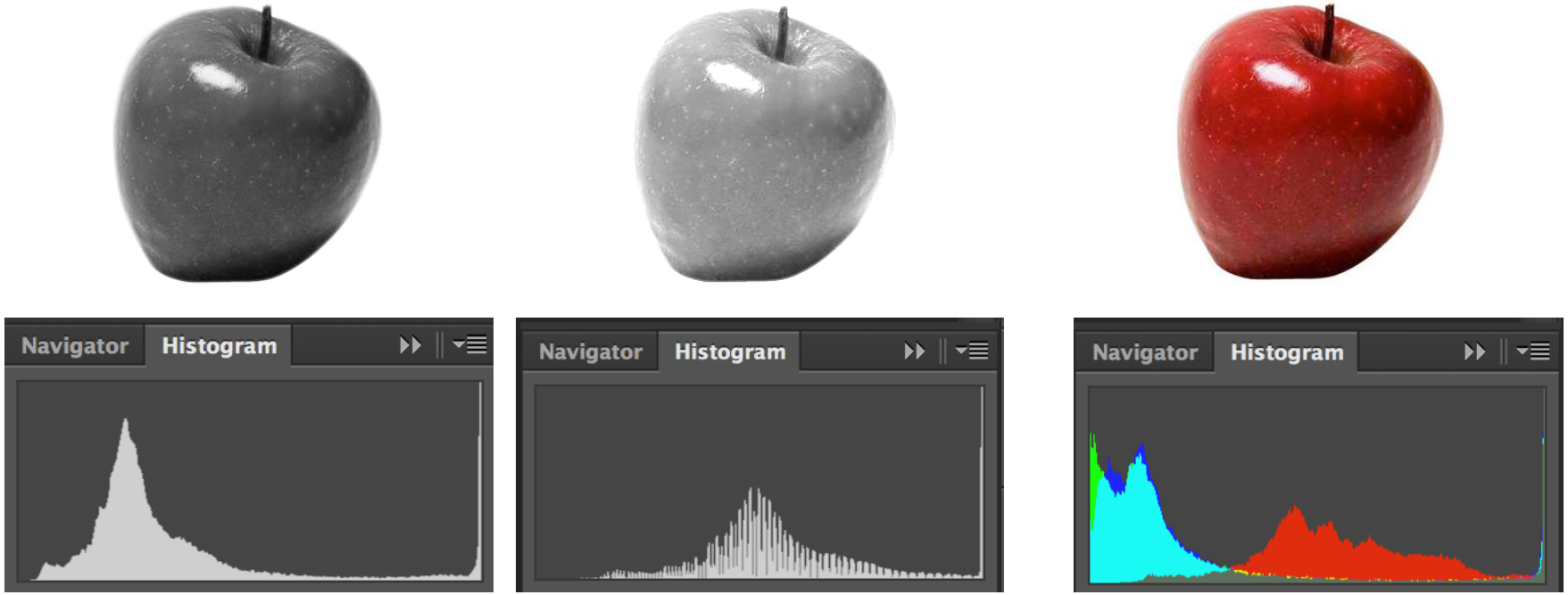


HSV

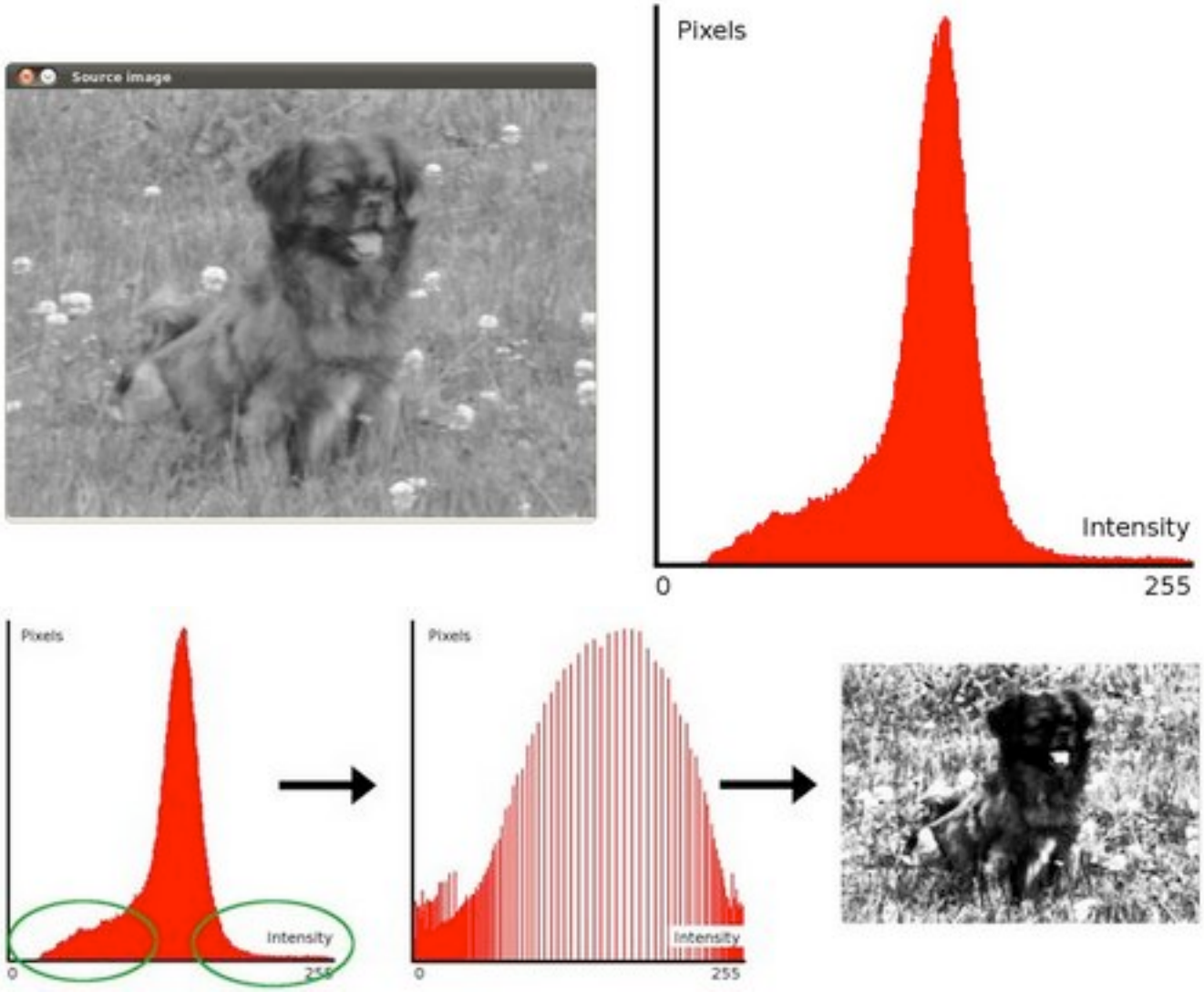


Intensity distributions - Histograms

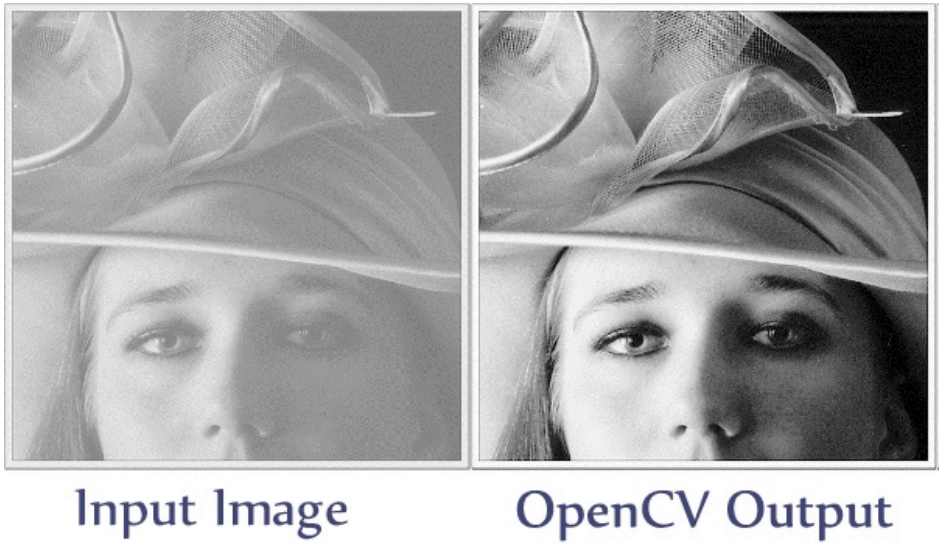
- Count pixels with certain intensities (gray or color), collect in bins and plot distribution



Histogram Equalization

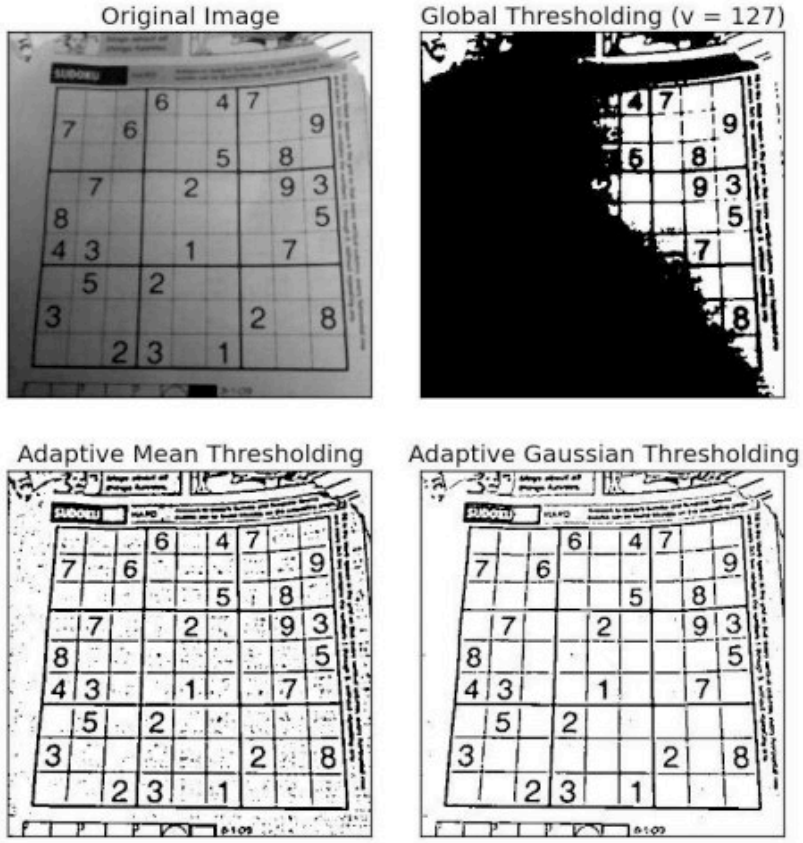


Histogram Equalization



Binary image processing: Thresholding

- Binarization of gray values with lower/ upper thresholds
- „Mathematically“: $I > v_{threshold}$
- Global vs. Local thresholds



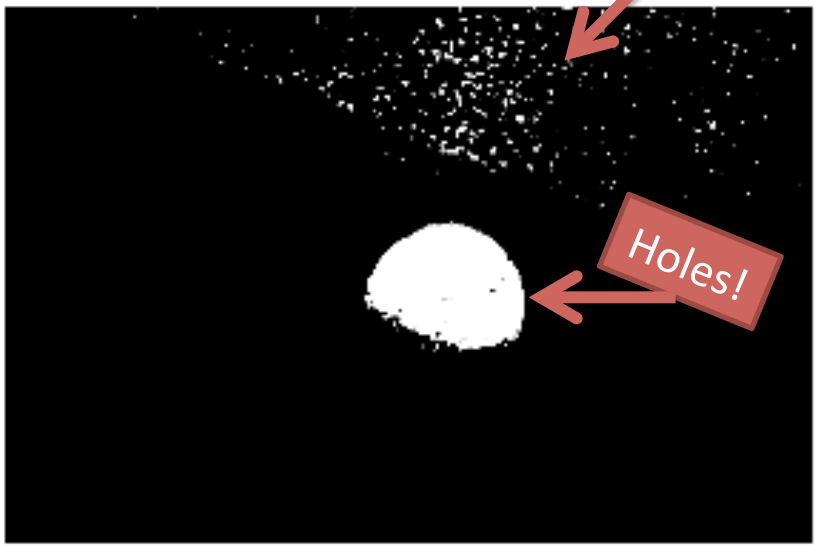
From OpenCV Python tutorials:
http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html#thresholding

In practice: thresholding of a colored ball

- Yellow in RGB: [255,255,0]
- Yellow in HSV: [30,255,255]
- Threshold HSV image between [20,50,50] to [40,255,255]

Image

Mask



Binary image processing: Morphological Operations

- Convolution of binary image with a binary convolution kernel
  - Very fast
  - Diverse usability and effective
- Different operations (and kernel shapes):
  - Erosion / dilation
  - Opening / closing
  - Gradient (edge detector)

Morphological operations
Dilation - Step-by-step

© Smar lear (published Jul 7, 2012)
https://www.youtube.com/watch?v=xO3ED27rMHs
https://www.youtube.com/user/TheSmartElearning

Dilation

(expanding existing white areas)

Morphological operations
Erosion - Step-by-step

© Smar lear (published Jul 7, 2012)
https://www.youtube.com/watch?v=fmyE7DiaIYQ
https://www.youtube.com/user/TheSmartElearning

Erosion
(erasing white areas)

Opening
(=Erode+Dilate, for noise removal)

Closing

(=Dilation+Erosion, fill gaps)

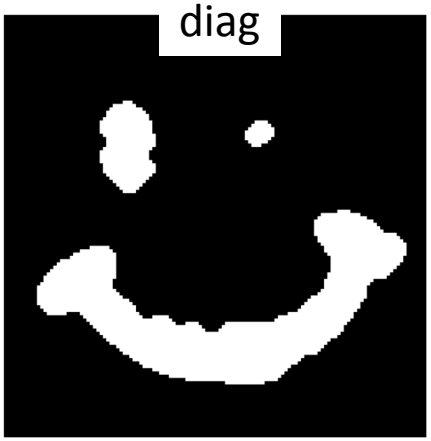
Arbitrary Combinations

(=Opening then closing, fill gaps reconstruct shape)

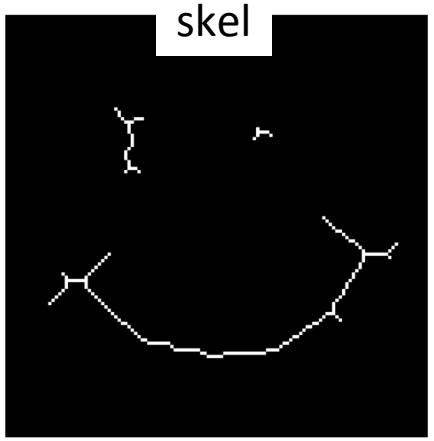
Advanced Filters

After opening+closing

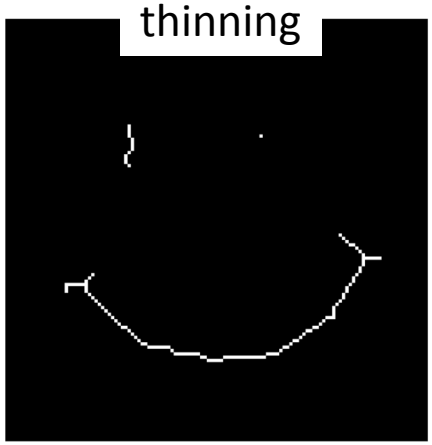
diag



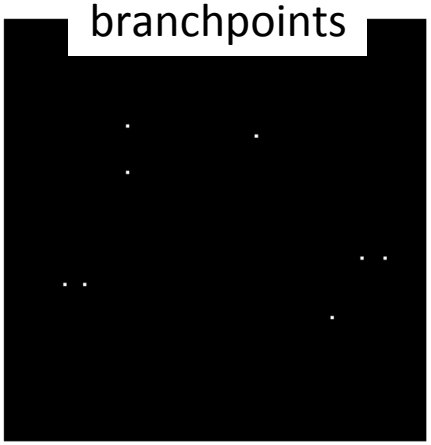
skel



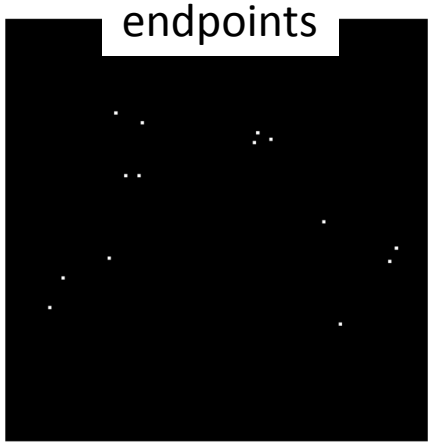
thinning



branchpoints

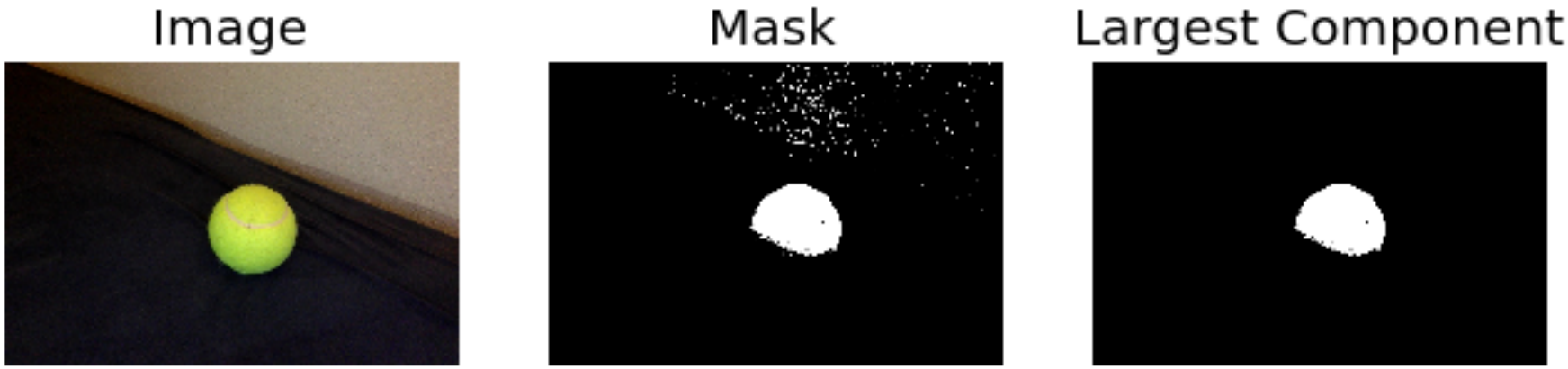


endpoints



Binary image processing: Connected Components

- Find connected blobs
- Choose largest connected component to exclude spurious detections



„Colored Ball Detection“
Pipeline so far…

- RGB2HSV conversion and color thresholding
- Connected components for largest object
- Morphological post-processing of mask

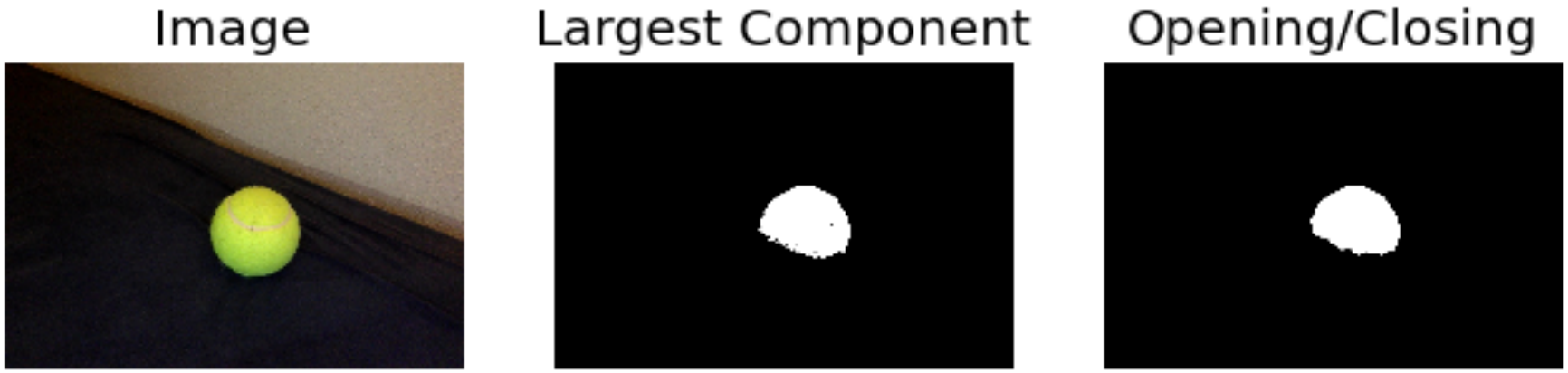


Image Filtering in Spatial Domain

- Morphological operations are filtering operations
- Image filtering is performed similarly, however instead of filtering with a binary mask („kernel“) as in binary morphological operations, we now use continuous-valued kernels.
- Filtering with a kernel is called **convolution**.

What is convolution?

Spatial image filtering in 2D: Correlation and Convolution

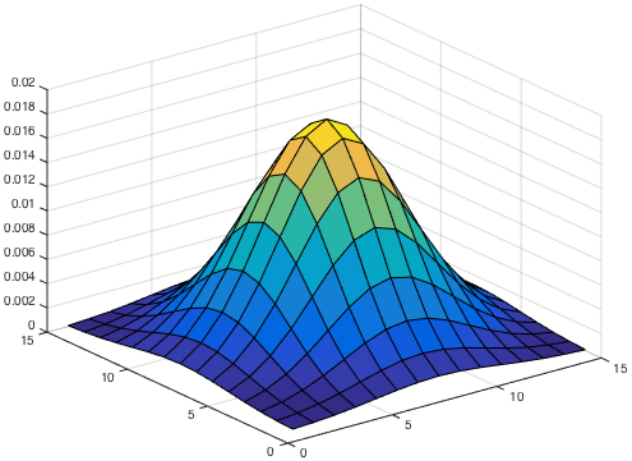
Filter guessing time...

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

Gaussian filter

$$\begin{bmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{bmatrix}$$



Colored ball - Pipeline update...

- Instead of color thresholding and complex post-processing (chain of morphological operations)...
- Gaussian blur and color thresholding

Image

Mask of raw

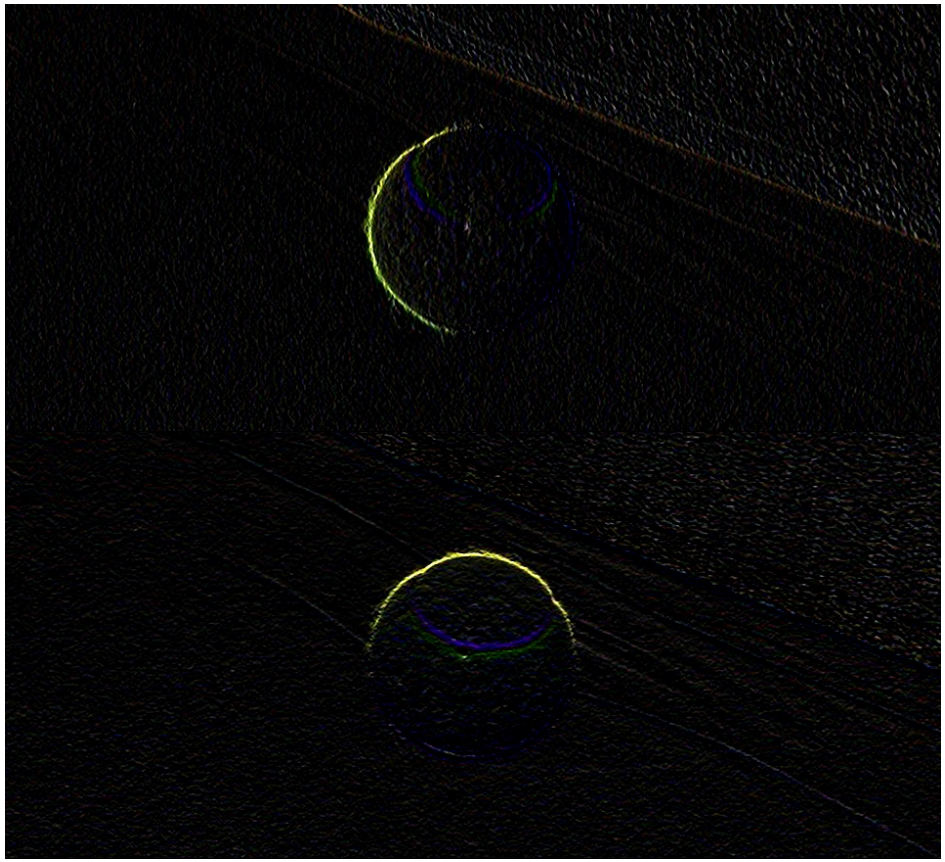
Mask of smoothed

Other important spatial filters
Sobel edge detector

- Sobel edge detector

x-Gradient $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

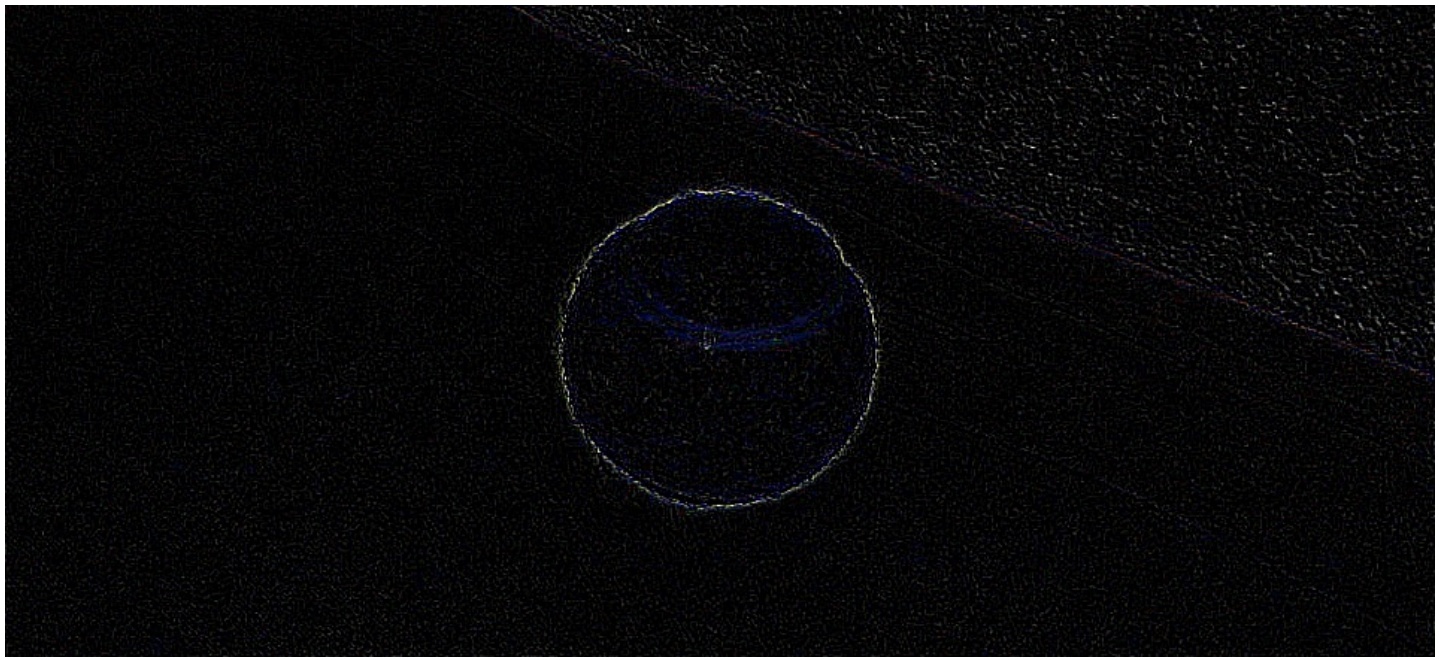
y-Gradient $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$



Other important spatial filters
Laplacian edge detector

- Laplacian edge detector

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



Canny Edge Detector

- Developed by John F. Canny in 1986
- **~22.000 citations** (!!!!!!!!!!!!!!!!!!!!!!!!!!!!)
- Algorithm
  - Gaussian Filter
  - Finding the Intensity Gradient of the Image
  - Non-maximum Suppression
  - Double Threshold
  - Edge Tracking by Hysteresis

Convolution!

Sobel!

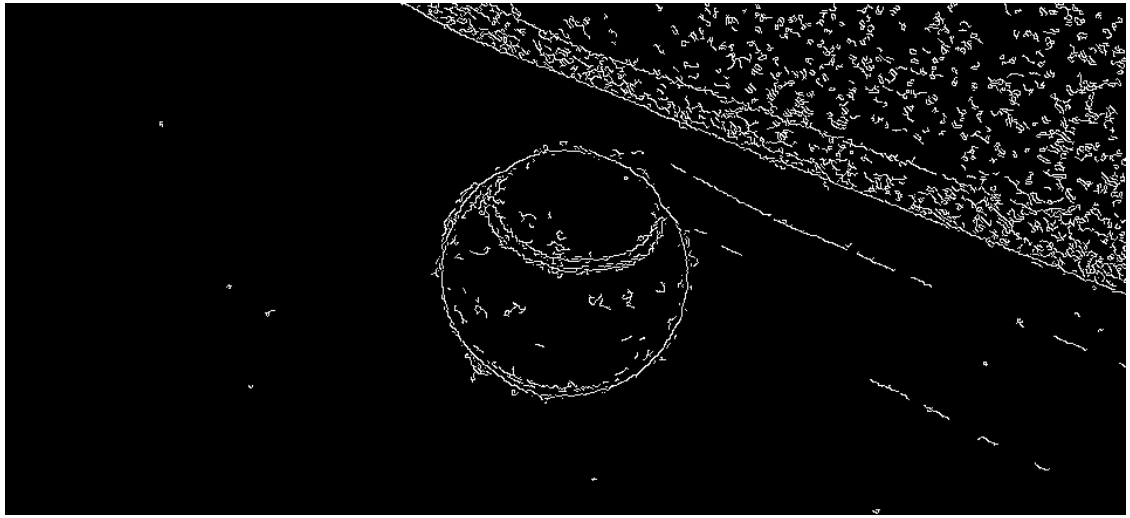
Thinning!

Thresholding!

Connected components!

Canny Edge Detector Result

Lower threshold: 50
Upper threshold: 100



Lower threshold: 100
Upper threshold: 150

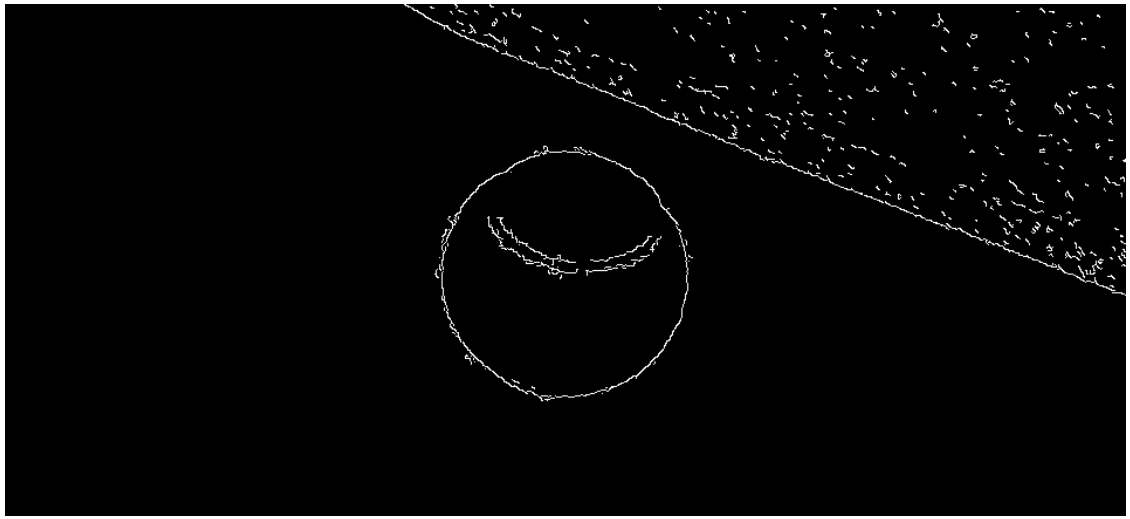
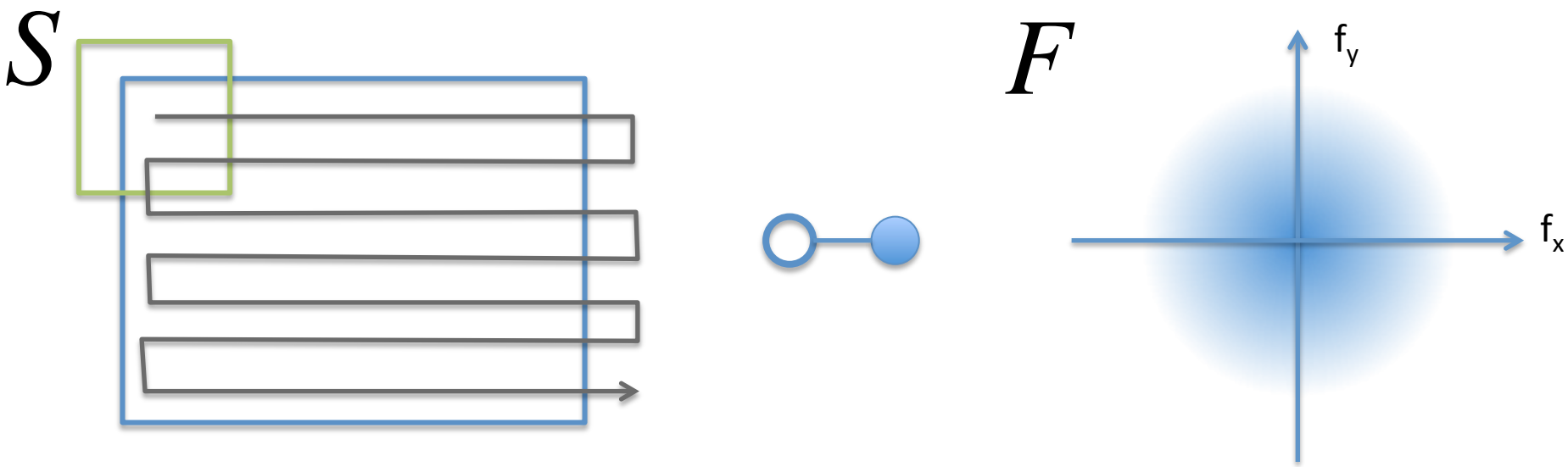
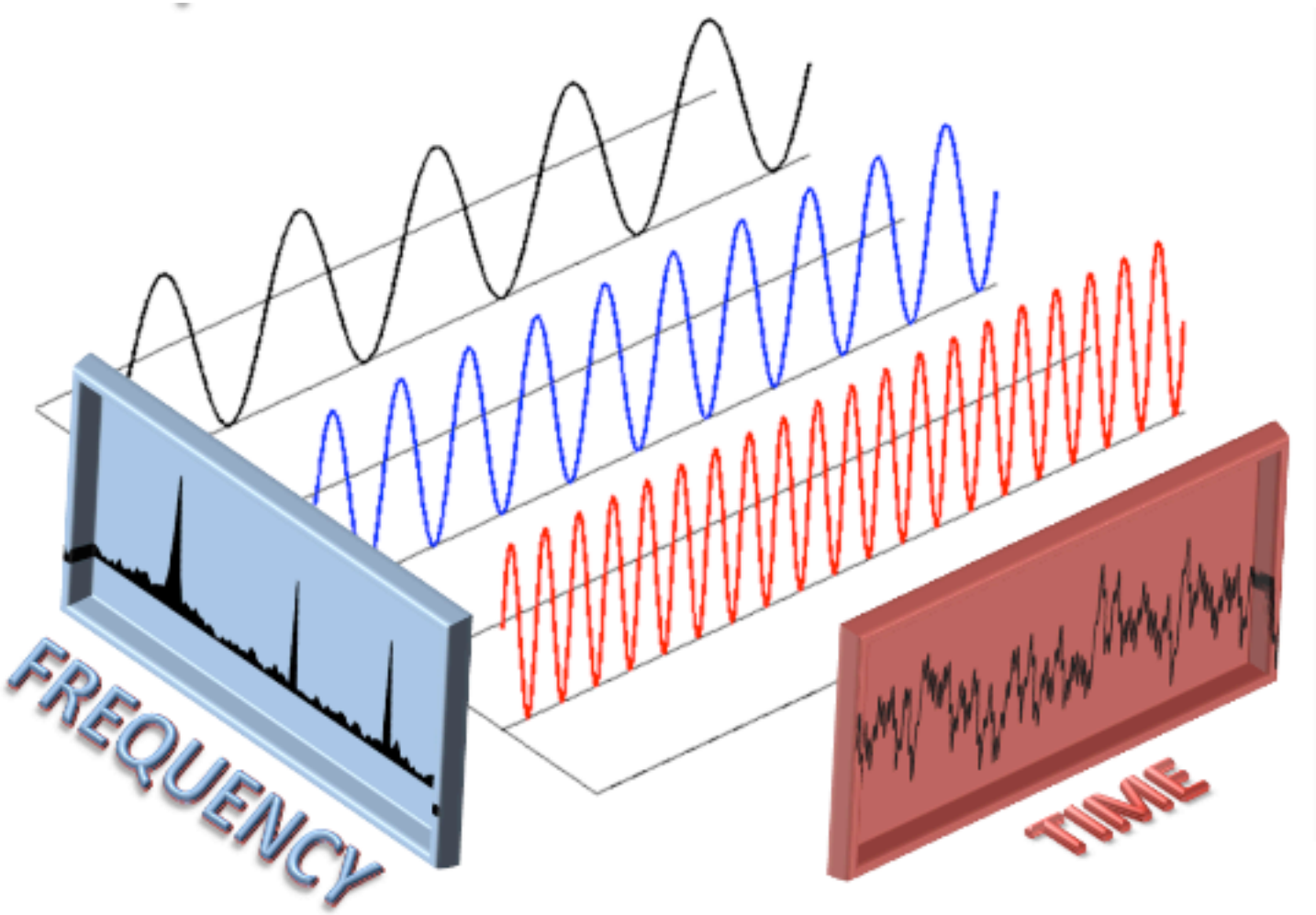


Image Filtering in Frequency Domain

- Complexity of spatial filtering is $O\left(I_x I_y h_x h_y\right)$
- Can be slow, especially for large images and filters
- Instead of working in spatial domain, we can achieve certain tasks faster and easier in frequency domain

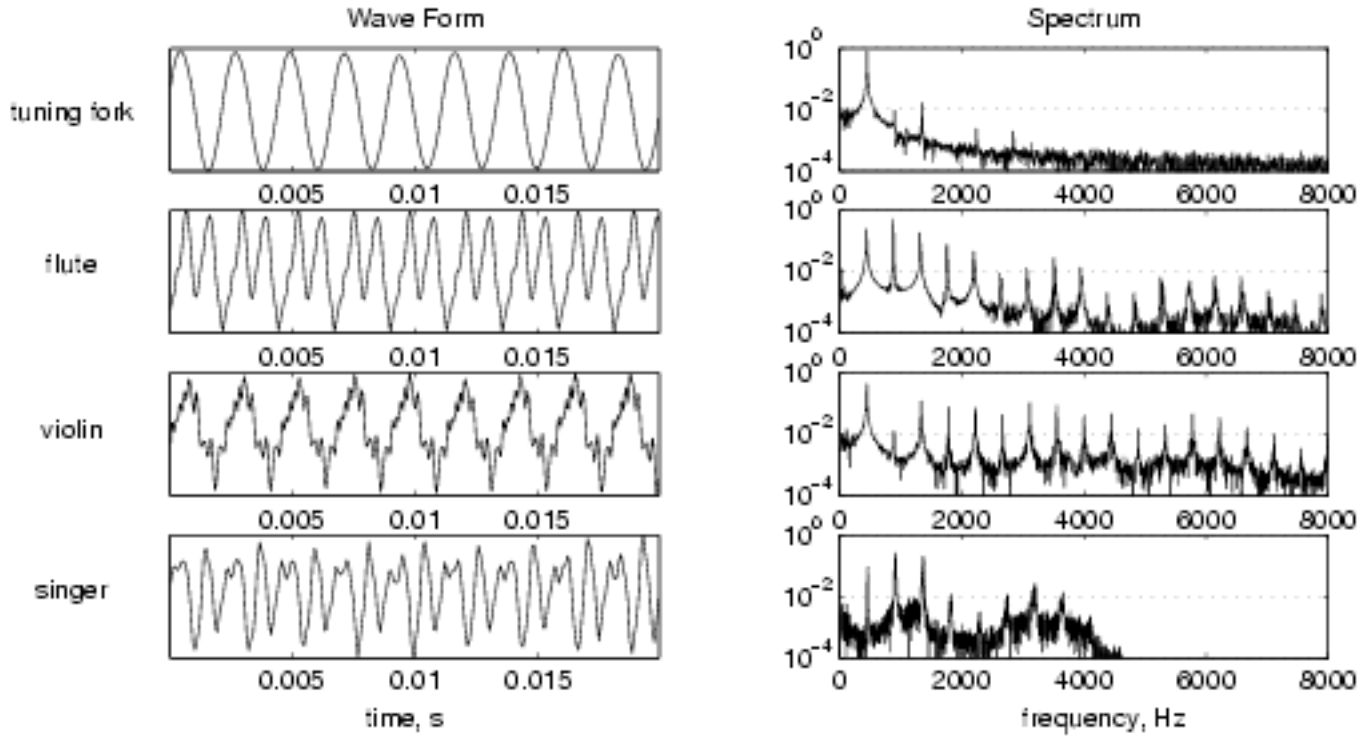


Recap Fourier Transform / Frequency Space



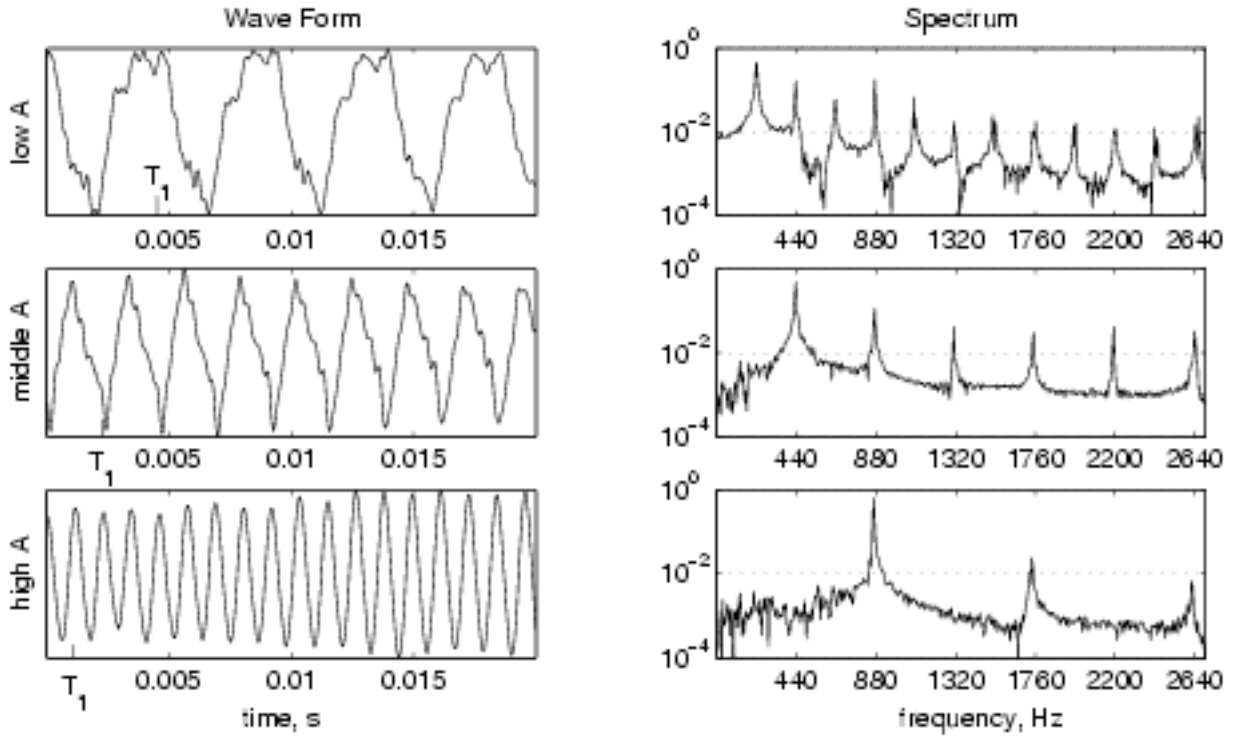
© MIT CSAIL
http://groups.csail.mit.edu/netmit/wordpress/projects/sparse-fourier-transform/

Recap Fourier Transform / Frequency Space



http://amath.colorado.edu/pub/matlab/music/

Recap Fourier Transform / Frequency Space

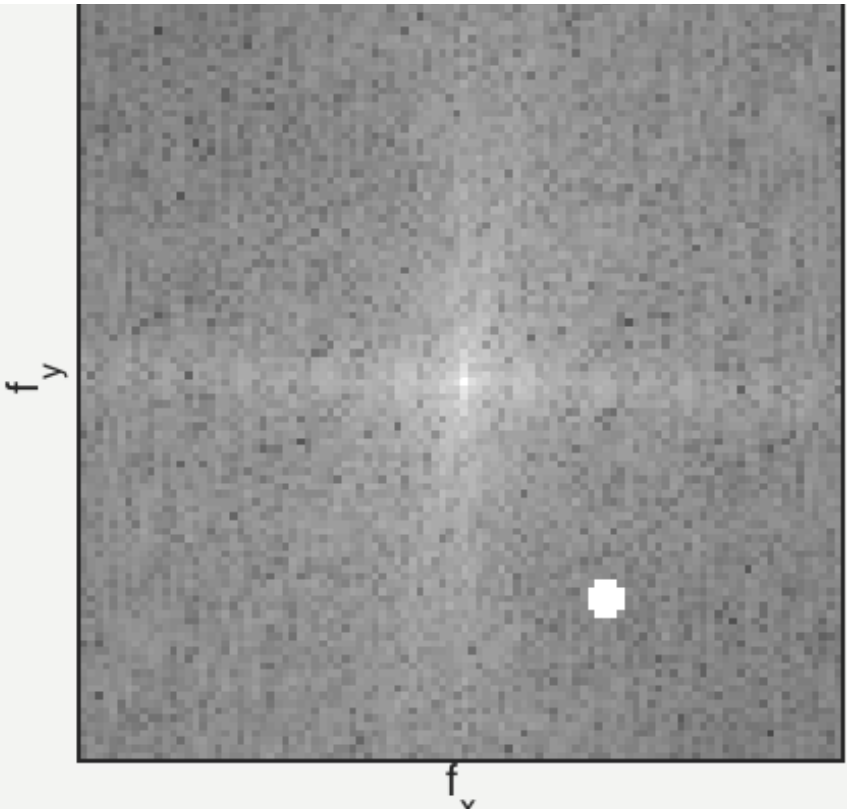


http://amath.colorado.edu/pub/matlab/music/

Recap: 2D Fourier Transform

Image Domain

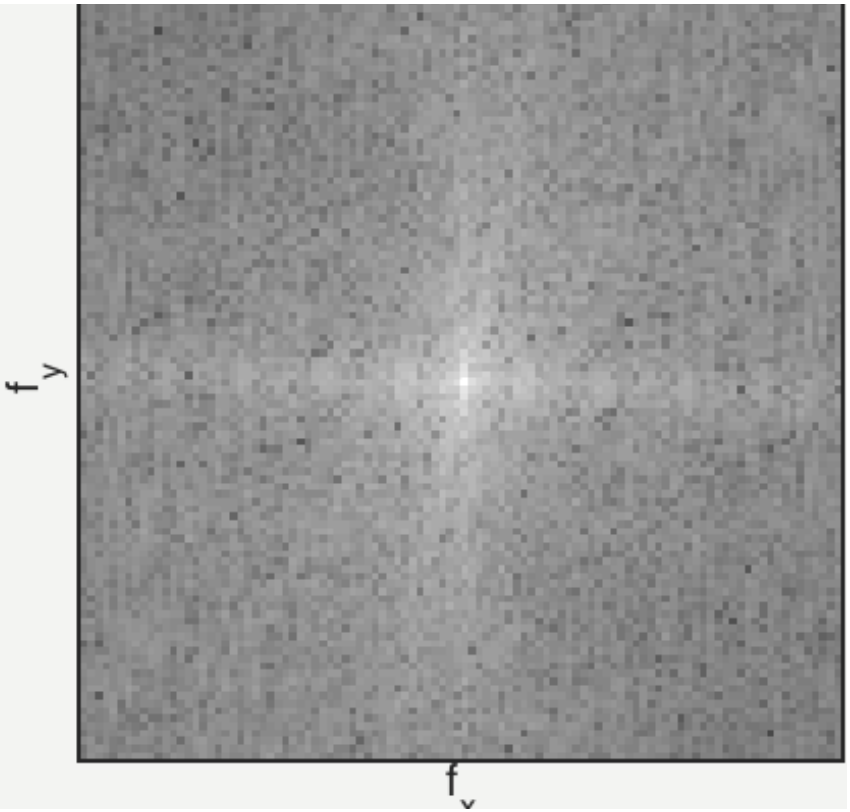
Frequency Domain



Filtering in Fourier Space

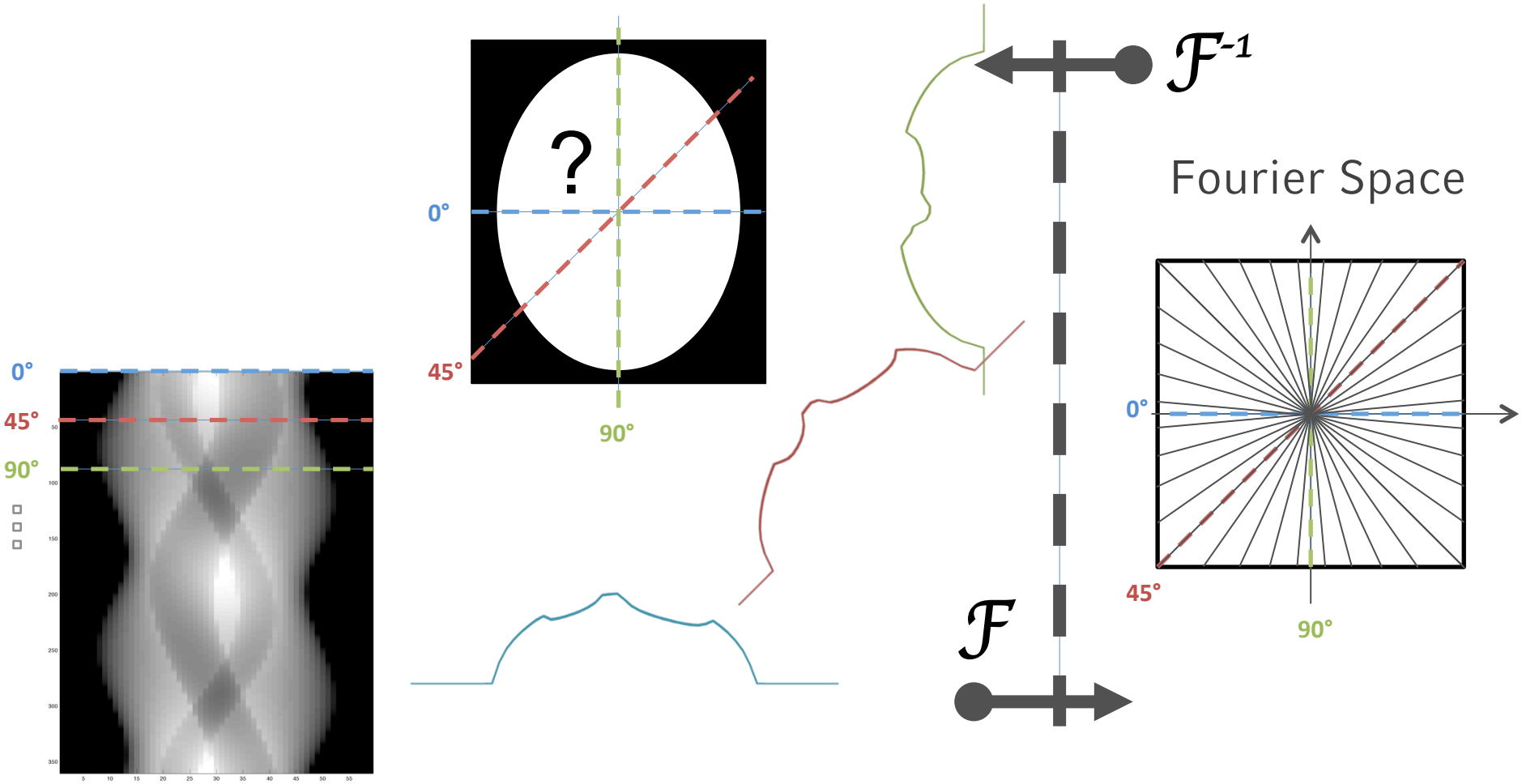
Image Domain

Frequency Domain

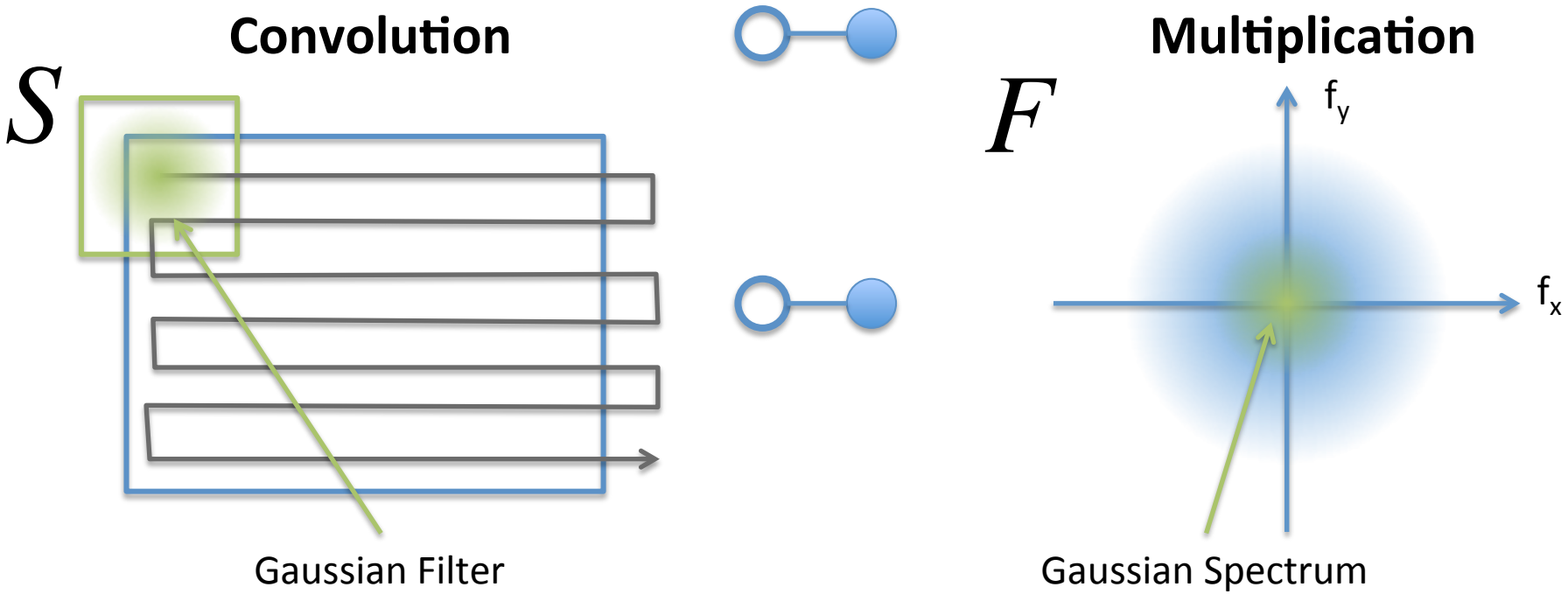


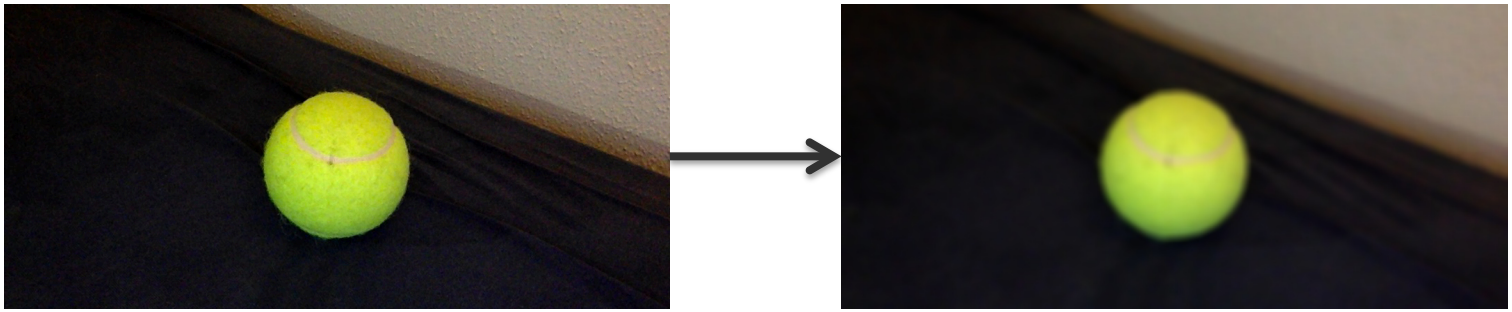
Arbitrary „filling“ of Frequency Space

CT Image Reconstruction
Projection-Slice Theorem



Linking Spatial and Frequency Filtering





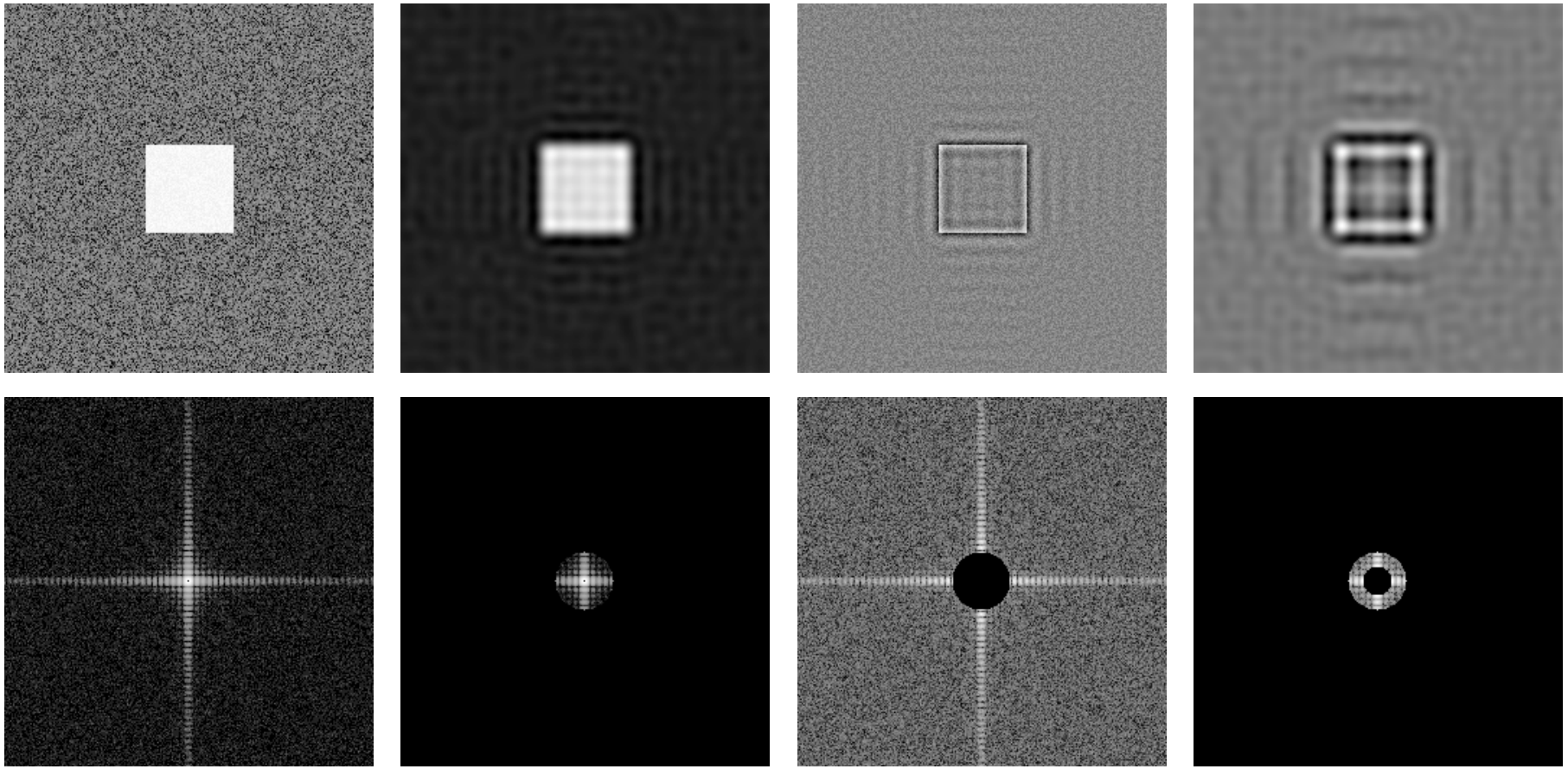
Most common frequency filters

Original

Lowpass
→ Smoothing

Highpass
→ Edges + noise

Bandpass
→ Smoothed edges



http://paulbourke.net/miscellaneous/imagefilter/

Summary

- Basic image processing can take you far, for **simple** object detection and tracking
  - Color conversion to HSV space
  - Histogram equalization / thresholding
  - Morphological operations
- Filters are powerful tools for image manipulation
  - Multipilication in frequency domain
  - Sliding of convolution kernel in spatial domain
- Disadvantages of simple methods so far
  - Sensitive to noise and outlier regions
  - Not robust to clutter in images, not very robust to illumination changes

Next step

- Create more robust tracking algorithms
- Tracking of textured objects

Questions?