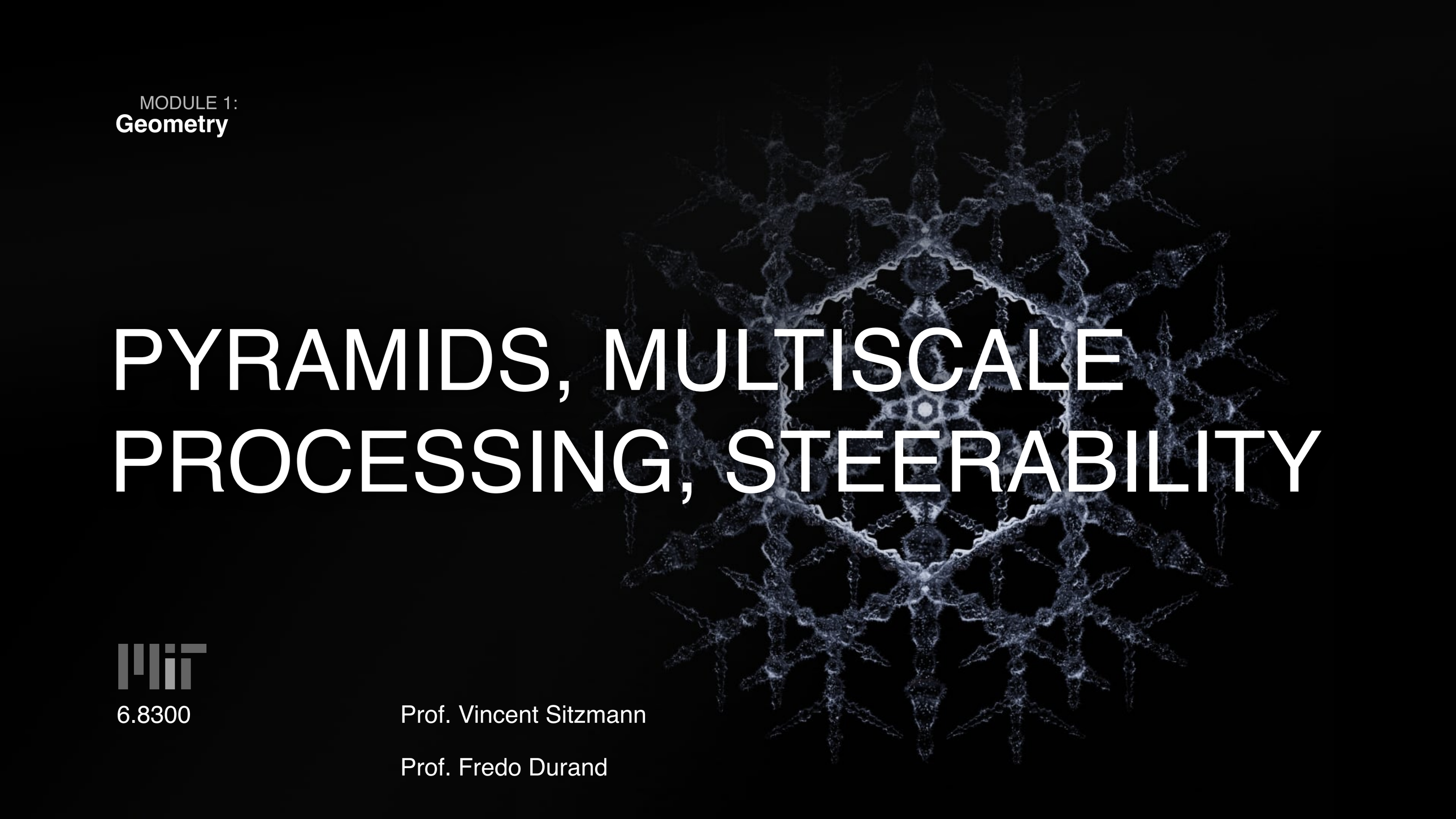
MODULE 1:
Geometry



PYRAMIDS, MULTISCALE PROCESSING, STEERABILITY

6.8300

Prof. Vincent Sitzmann

Prof. Fredo Durand

Demo

- Also visit the Edgerton center and demo in hallway 4th floor building 4 https://edgerton.mit.edu/




AERONAUTICA
MILITARE
15
52

Clarification about last lecture

- I played fast and loose with 2D vs 1D
- I told you 2D images can be represented as 1D vectors
- I used 1D signals to introduce a lot of concepts

- But applying an operator (convolution, Fourier transform) designed for 1D signals onto 2D images represented as 1D vector will NOT yield the corresponding 2D operator
  - It’s an operator, sure, but probably not the one you want

Careful

- You can apply the 2D Fourier Transform to the 1D vector representation of 2D images
- But it's a different matrix from the 1D Fourier Transform.

- Matrices for stuff that's fundamentally 2D (images) are more messy than those for 1D stuff
  - e.g. convolution is not a nice multidiagonal

Example: convolution

- 3x2 image convolved by 2x2 kernel

$$I = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \triangleright Vectoriaed\ I = \begin{bmatrix} 4 \\ 5 \\ 6 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

$$F = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$$

$$\begin{bmatrix}
30 & 0 & 0 & 0 & 0 & 0 \\
40 & 30 & 0 & 0 & 0 & 0 \\
0 & 40 & 30 & 0 & 0 & 0 \\
0 & 0 & 40 & 0 & 0 & 0 \\
10 & 0 & 0 & 30 & 0 & 0 \\
20 & 10 & 0 & 40 & 30 & 0 \\
0 & 20 & 10 & 0 & 40 & 30 \\
0 & 0 & 20 & 0 & 0 & 40 \\
0 & 0 & 0 & 10 & 0 & 0 \\
0 & 0 & 0 & 20 & 10 & 0 \\
0 & 0 & 0 & 0 & 20 & 10 \\
0 & 0 & 0 & 0 & 0 & 20
\end{bmatrix}$$

https://stackoverflow.com/questions/16798888/2-d-convolution-as-a-matrix-matrix-multiplication

In general

$$\triangleright Vectoriaed\ I = \begin{bmatrix} 4 \\ 5 \\ 6 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

- When we vectorize the image to a 1D vector we create “blocks”
- The matrices that act on such vectors are also organized in blocks

Tensors

- Tensors make things cleaner but at the cost of higher dimensions
- Tensors allow us to keep images 2D but then operators like convolution become 4D tensors
- No matter what, things are hard to visualize

Back to high-level discussion

- **Fourier eigenbasis great**
  - Diagonalize shifts and convolution, models aliasing, extracts different frequencies/scale
- **but**
  - 1/ assumes boundary wrapping (cyclic images, torus)
  - 2/ is too global
    (makes sense: to be shift-invariant enough,
    your support needs to be everywhere)
- **A lot of image processing is about striking a balance between Fourier, the boundary issue, and trying to keep things local**
  - e.g bicubic, JPEG, MPEG, pyramids, wavelets

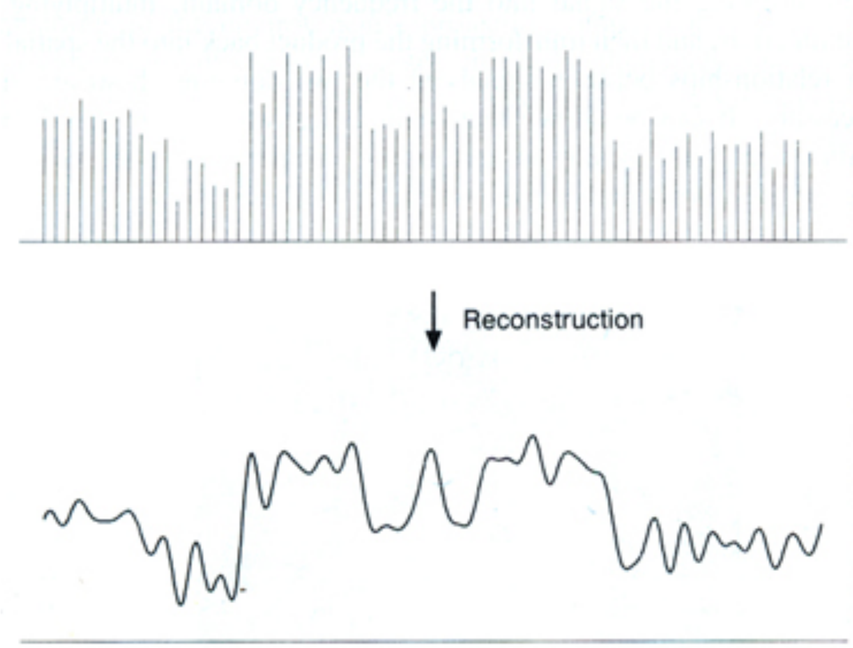
Example of balance: bicubic filters

- For anti-aliasing, image reconstruction, resampling

Antialiasing for resampling:
Before reducing resolution, filter out high frequencies that cannot be handled

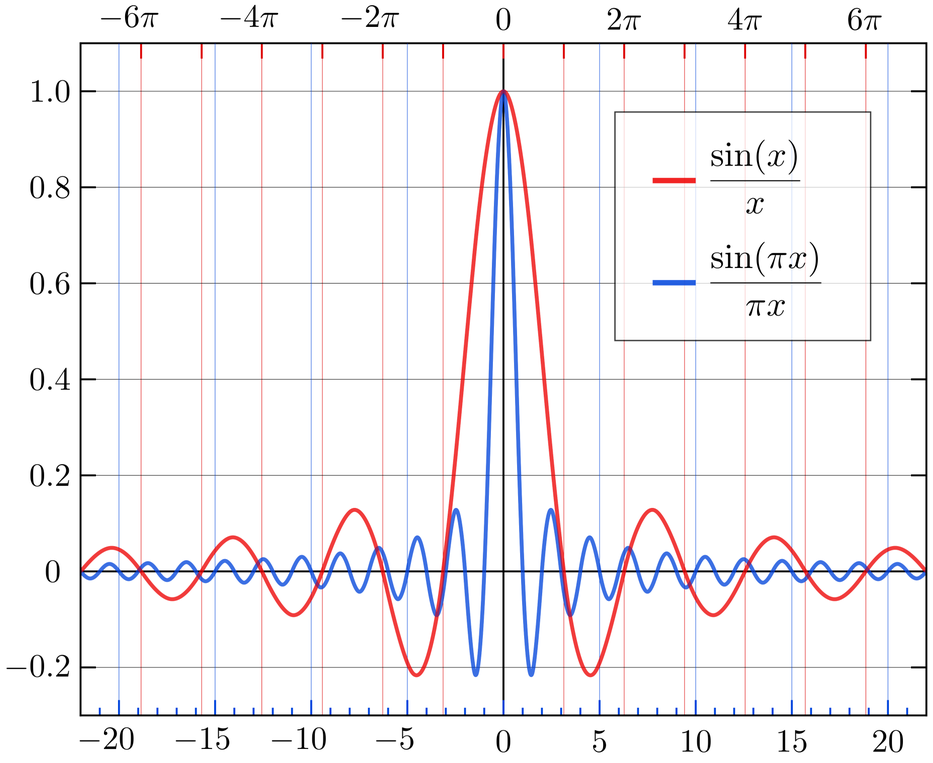


Reconstruction: converting a sampled representation back into a continuous function by "guessing" what happens between the samples



Ideal filter

- Math says the filter that perfectly eliminates frequencies above a certain value is a sinc (sin(x)/x)
  - (inverse Fourier transform of box)
- But sinc has infinite support, assumes wrap-around boundary (torus)



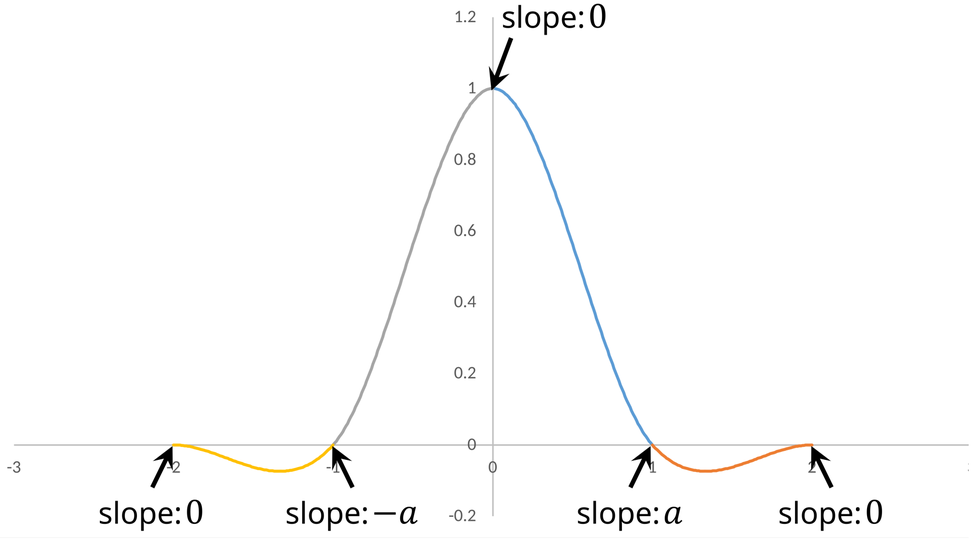
https://en.wikipedia.org/wiki/Sinc_function

Bicubic or Lanczos filters

- **Filters that look enough like a sinc but with finite support**
  - e.g. bicubic, Lanczos, etc.
  - Tradeoff between compact in space vs compact in frequency

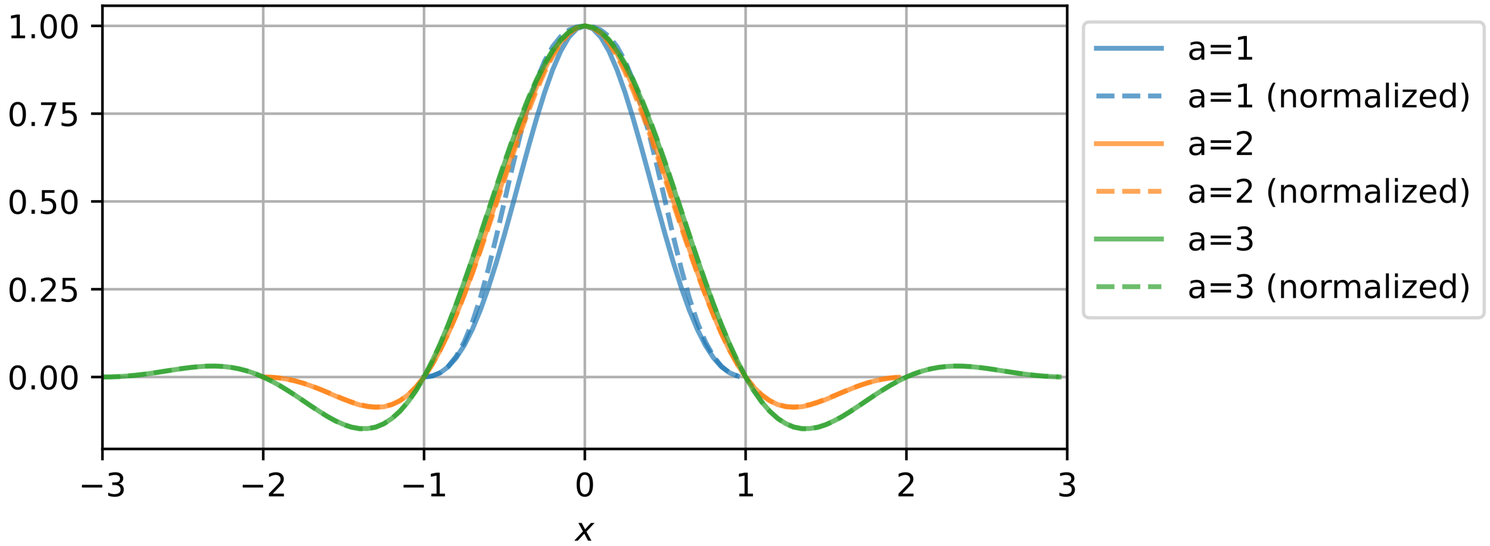
Cubic

$$W(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & \text{for } |x| \le 1, \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & \text{for } 1 < |x| < 2, \\ 0 & \text{otherwise,} \end{cases}$$



Lanczos

$$L(x) = \begin{cases} \operatorname{sinc}(x)\operatorname{sinc}(x/a) & \text{if } -a < x < a, \\ 0 & \text{otherwise.} \end{cases}$$



Compact in space vs. frequency

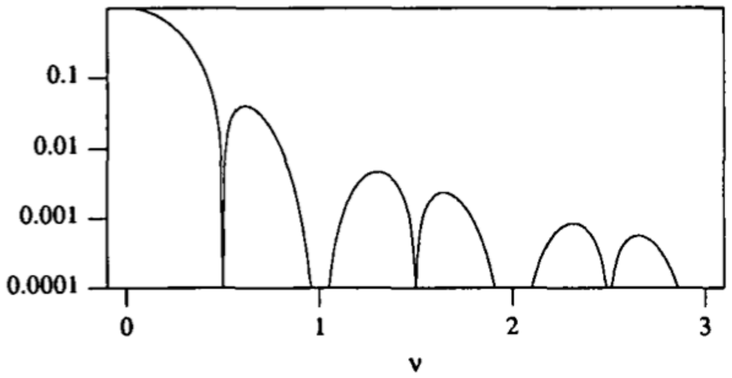


Figure 17. Frequency Response of Cubic Notch Filter

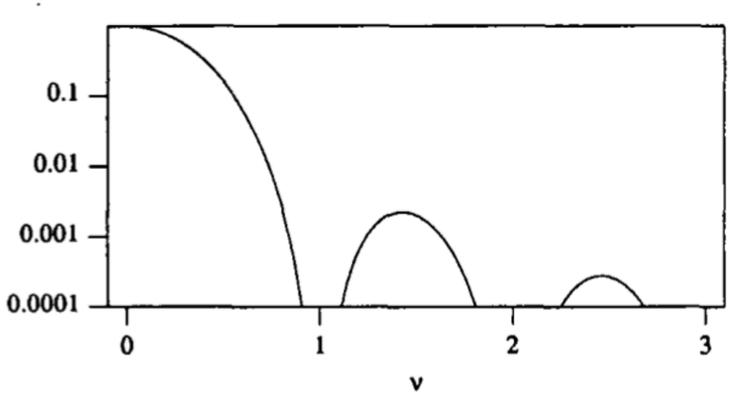
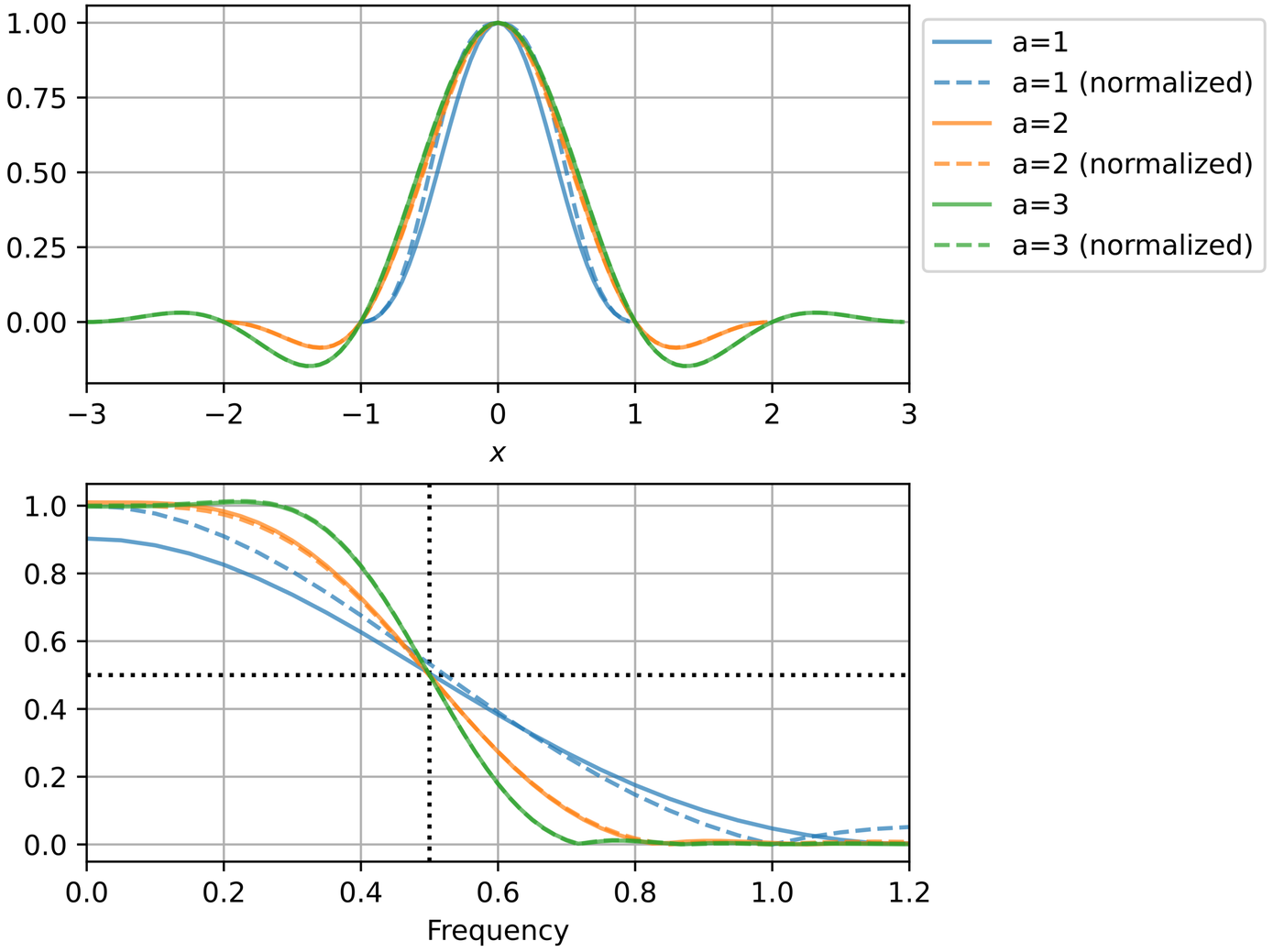


Figure 18. Frequency Response of Cubic B-Spline Filter



https://en.wikipedia.org/wiki/Lanczos_resampling

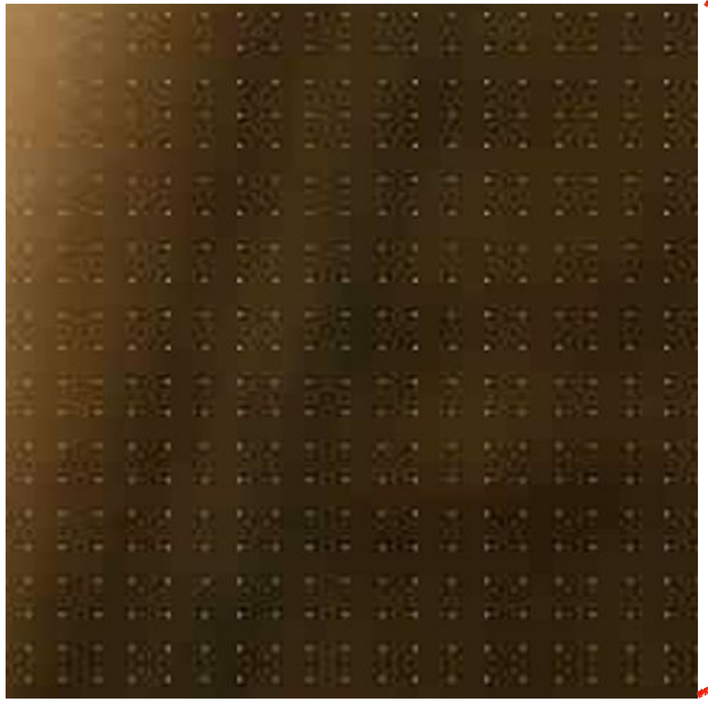
Example of balance: antialiasing



Slide credit: Christian Rupprecht, Oxford

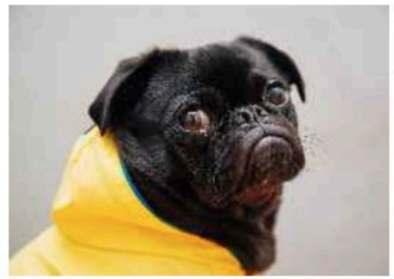
Warning: many implementations of bicubic downsampling are incorrect

- https://twitter.com/AlexTamkin/status/1456149826337263621?fbclid=IwAR3_IH-3KwU3mf3eWfajZ0HiI-gIOAtCGkOxb5JkvzFUwfMCqqrg_PVdz80





Downscaling
in TensorFlow



Manipulated image

Output image

Figure 1: Example of an image-scaling attack. Left: a manipulated image showing a cat. The scaling operation produces the right image with a dog.

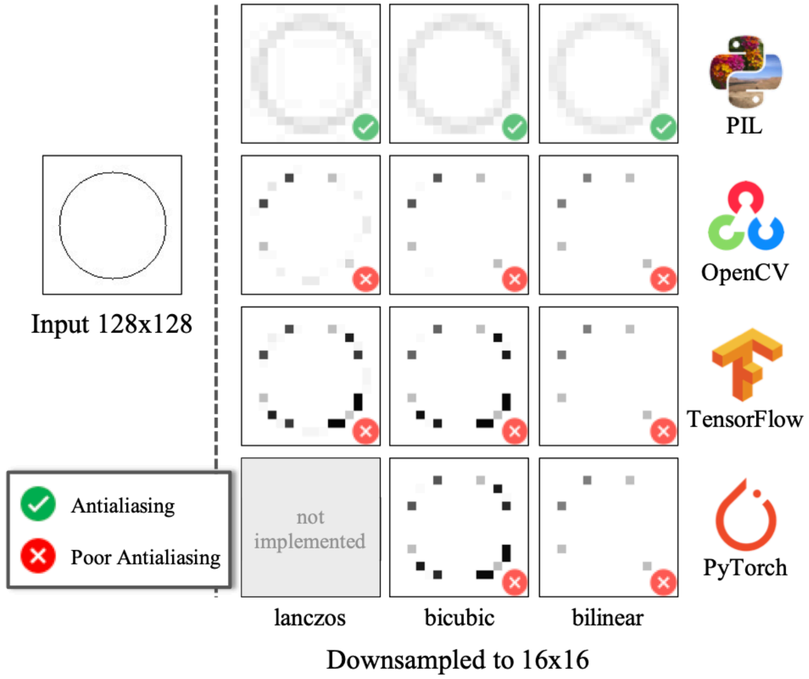
- https://arxiv.org/pdf/2104.11222.pdf

On Buggy Resizing Libraries and Surprising Subtleties in FID Calculation

Gaurav Parmar[1] Richard Zhang[2] Jun-Yan Zhu[1]
[1]Carnegie Mellon University [2]Adobe Research

Abstract

We investigate the sensitivity of the Fréchet Inception Distance (FID) score to inconsistent and often incorrect implementations across different image processing libraries. FID score is widely used to evaluate generative models, but each FID implementation uses a different low-level image processing process. Image resizing functions in commonly-used deep learning libraries often introduce aliasing artifacts. We observe that numerous subtle choices need to be made for FID calculation and a lack of consistencies in these choices can lead to vastly different FID scores. In particular, we show that the following choices are significant: (1) selecting what image resizing library to use, (2) choosing what interpolation kernel to use, (3) what encoding to use



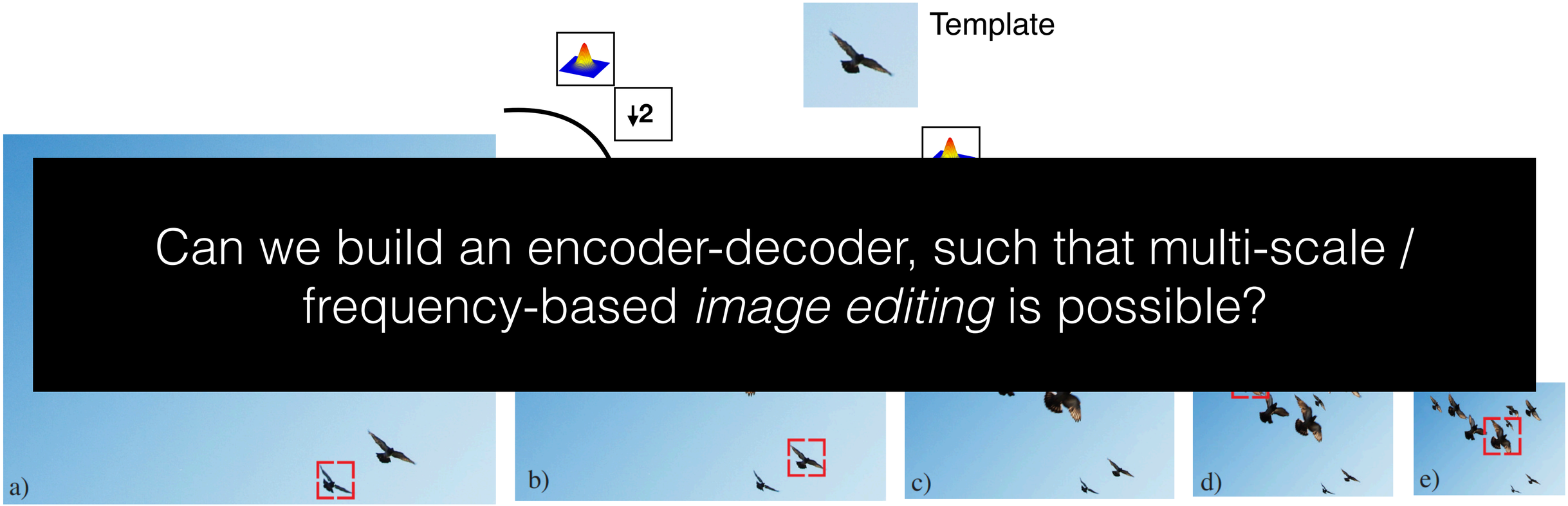
Beyond shifts: Detection across shift & scale





Template

The Gaussian Pyramid



Can we build an encoder-decoder, such that multi-scale / frequency-based *image editing* is possible?

A simple “encoder”: take image and transform it into a “representation” where analysis is simple

Eigenvector approach is possible but challenging

“Mellin” transform

https://en.wikipedia.org/wiki/Mellin_transform

But requires assumption of infinite domain,
boundary problems are way worse than Fourier

Pragmatic solution:

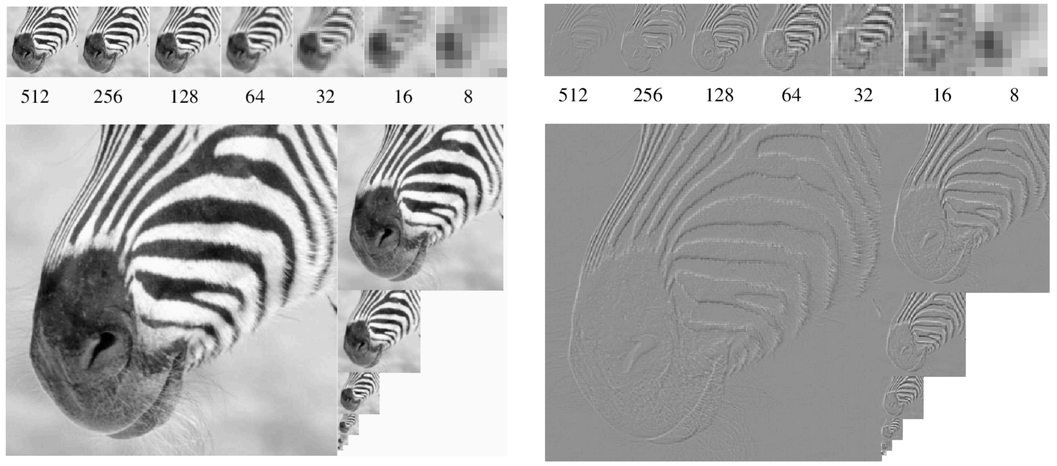
Something half-way between local and frequency

Scaling is simple in this representation, not quite diagonal though

Warning: not even a basis, overcomplete (more coefficients than pixels)

Image pyramid idea

- Process images at different resolutions
- Operating independently at different scales enables many applications
  - Blending
  - Denoising
  - Tone mapping
  - Analysis
  - Compression
  - Optical flow (motion a
  - Key points
  - CNN & UNet



Original papers from the 80s

- Burt & Adelson ; Williams

532 IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

The Laplacian Pyramid as a Compact Image Code

PETER J. BURT, MEMBER, IEEE, AND EDWARD H. ADELSON

A Multiresolution Spline With Application to Image Mosaics

PETER J. BURT and EDWARD H. ADELSON
RCA David Sarnoff Research Center

Computer Graphics Volume 17, Number 3 July 1983

Pyramidal Parametrics

Lance Williams

Computer Graphics Laboratory
New York Institute of Technology
Old Westbury, New York

- Later and slightly different: wavelets

674 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 11, NO. 7, JULY 1989

A Theory for Multiresolution Signal Decomposition: The Wavelet Representation

STEPHANE G. MALLAT

The Lifting Scheme:
A New Philosophy in Biorthogonal Wavelet Constructions

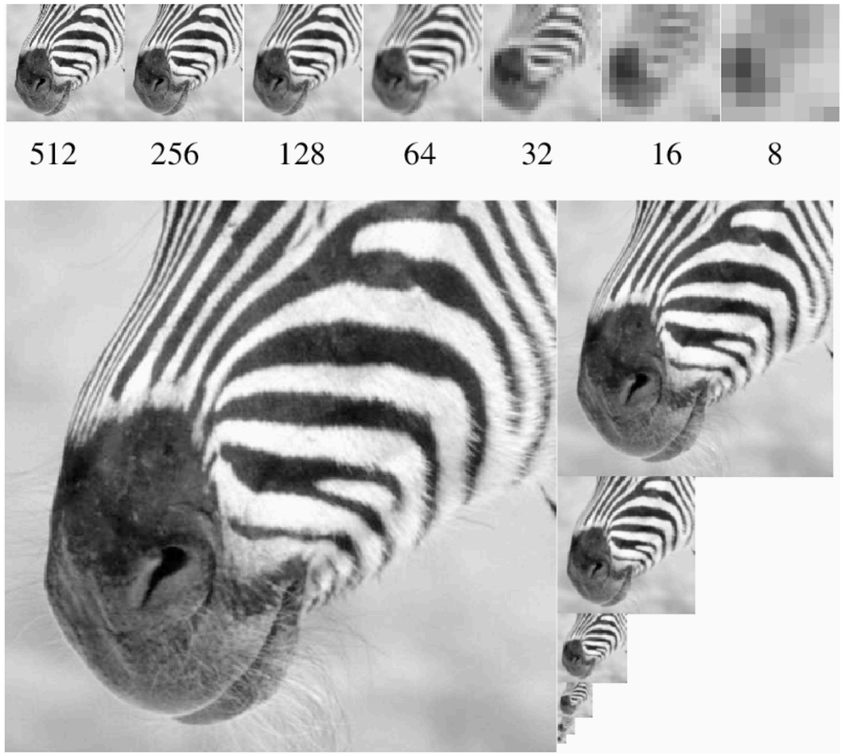
Wim Sweldens

Katholieke Universiteit Leuven Belgium, Department of Computer Science

Wim.Sweldens@cs.kuleuven.ac.be

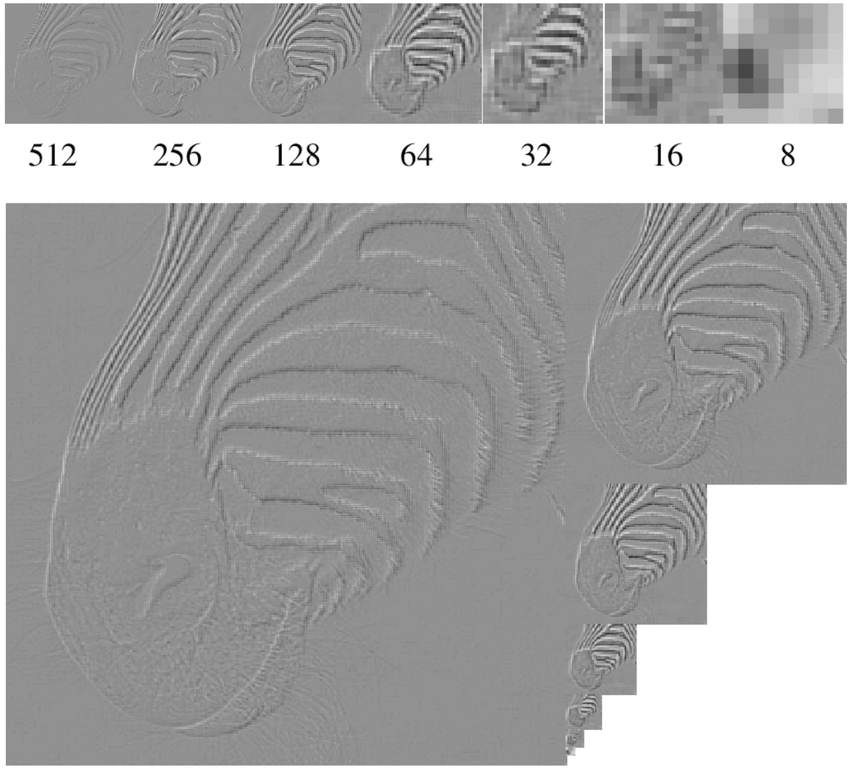
Two main families

Low pass
Each level is a lower resolution of the original



Gaussian pyramid

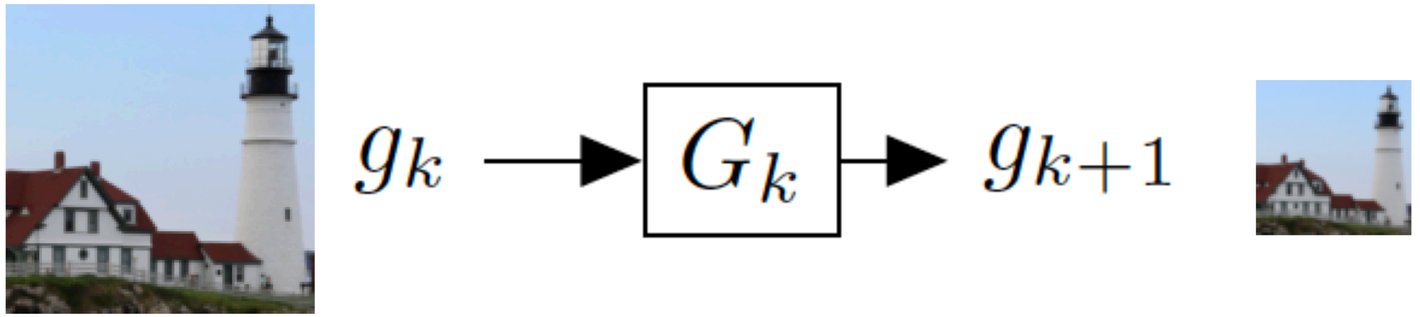
Band pass:
Only narrow frequencies per level



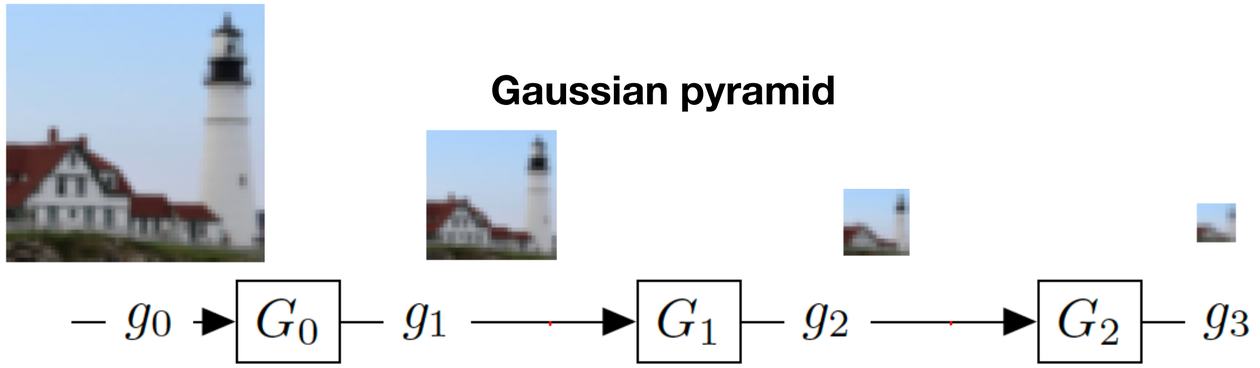
Laplacian pyramid

The Gaussian Pyramid

At each level, blur with Gaussian blur then downsample



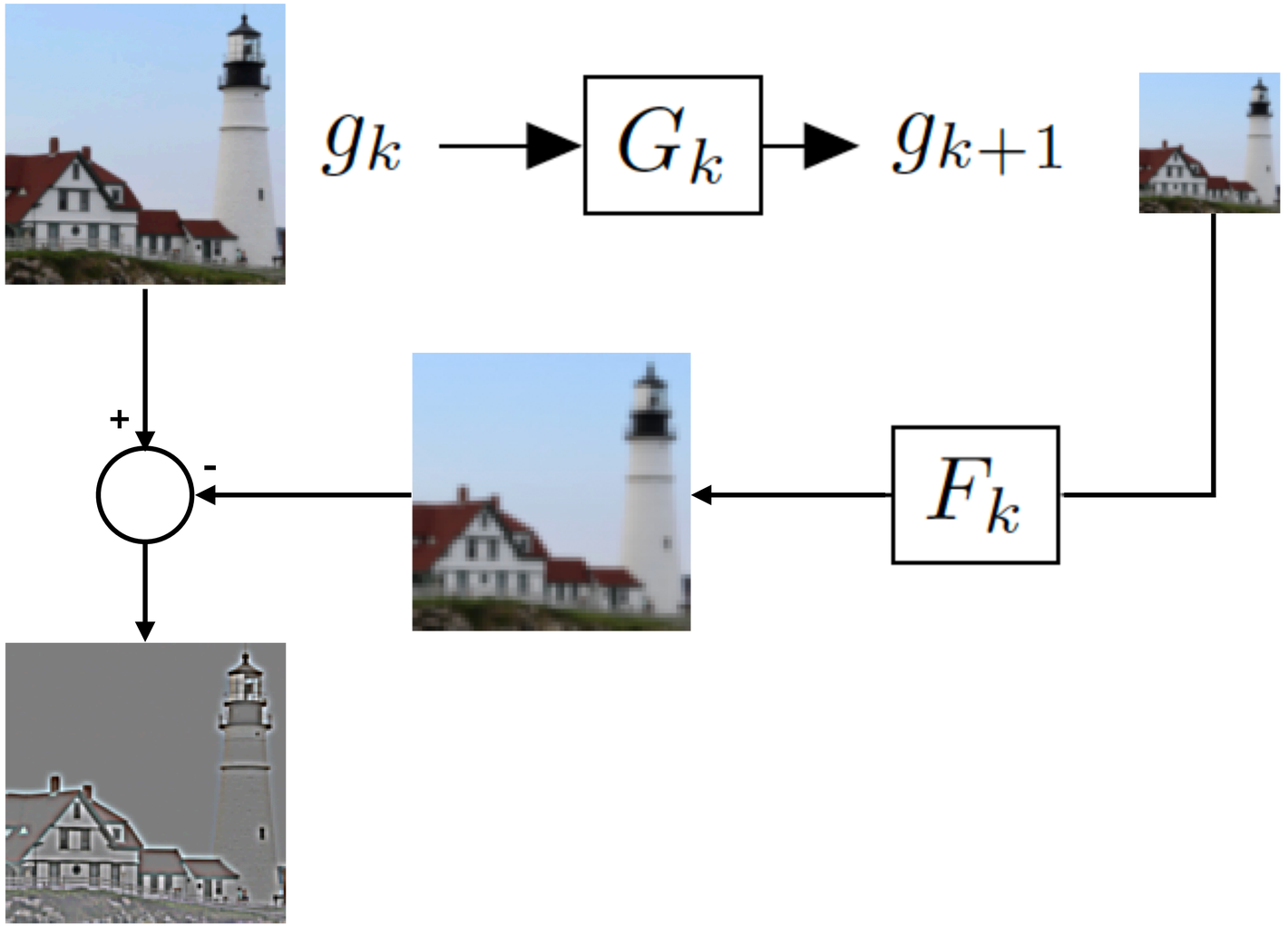
The Gaussian Pyramid



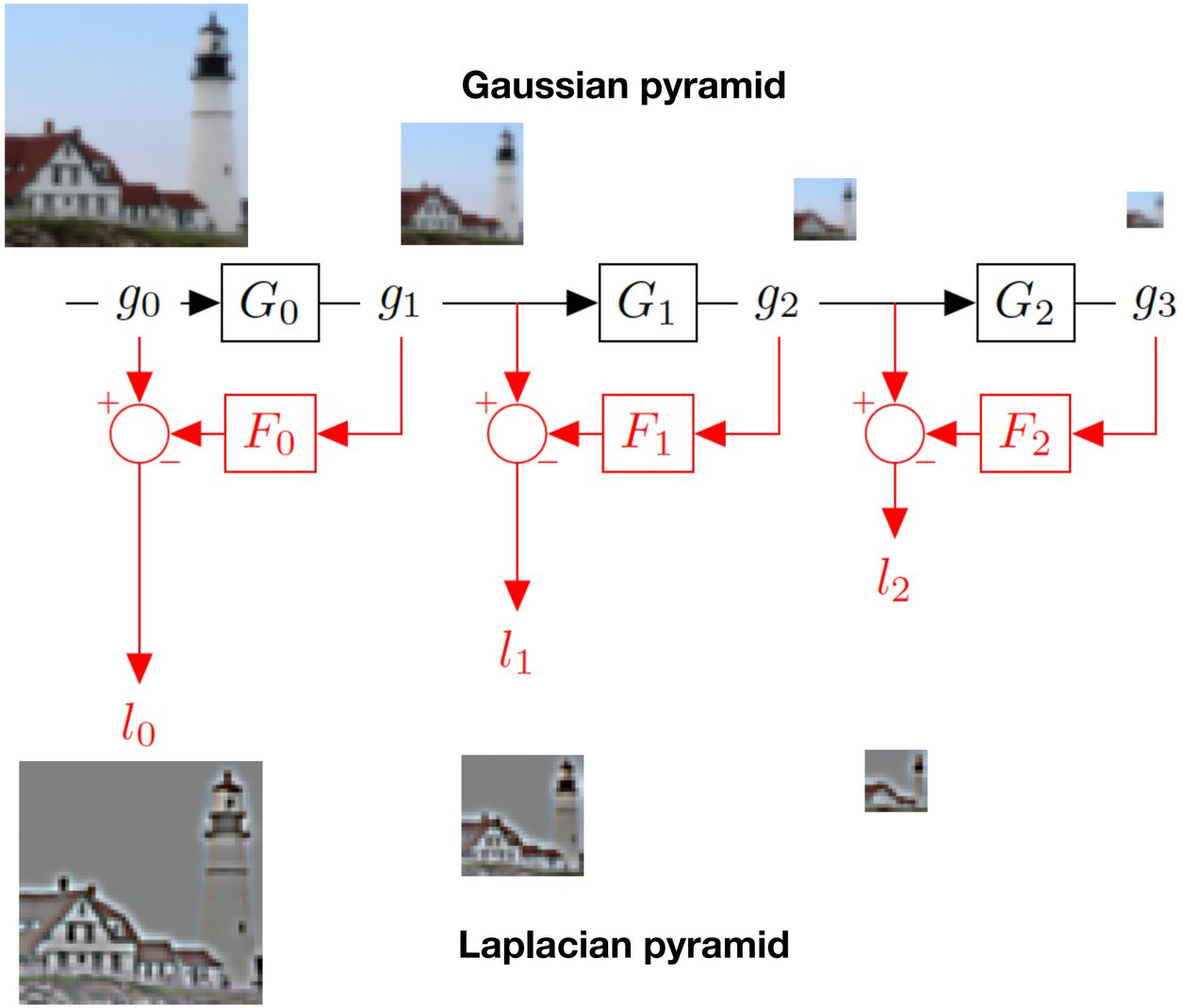
Very over complete but makes information at different scales directly accessible

The Laplacian Pyramid

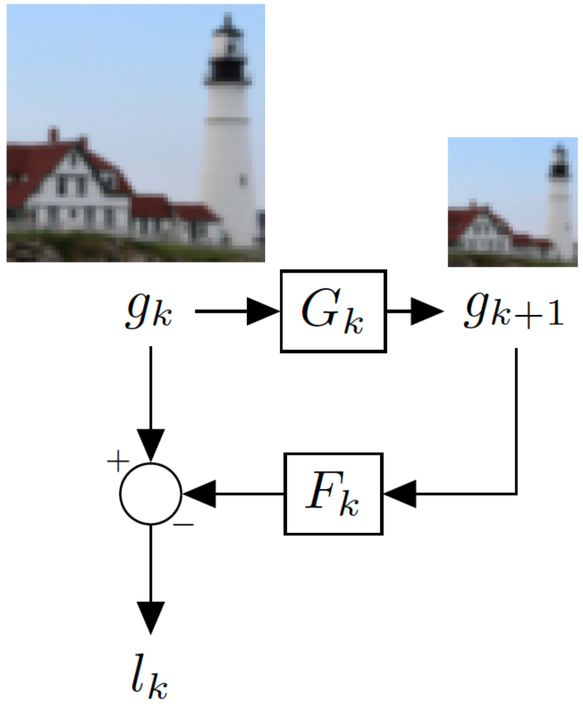
Compute the difference between upsampled Gaussian pyramid level k+1 and Gaussian pyramid level k.



The Laplacian Pyramid



The Laplacian Pyramid operators in matrix form



Blurring and downsampling:

$$G_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \frac{1}{16} \begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \end{bmatrix}$$

(Downsampling by 2)

(blur)

Upsampling and blurring:

$$F_0 =$$

Tangent on vectors, matrices, and tensors

- Matrices re useful here to place the work in full linear algebra context
  - But nobody ever stores these matrices
  - For example, for convolution, just store the kernel and write code that applies the operator.
  - The same is true for CNNs
- Plus here I show you matrices for 1D images

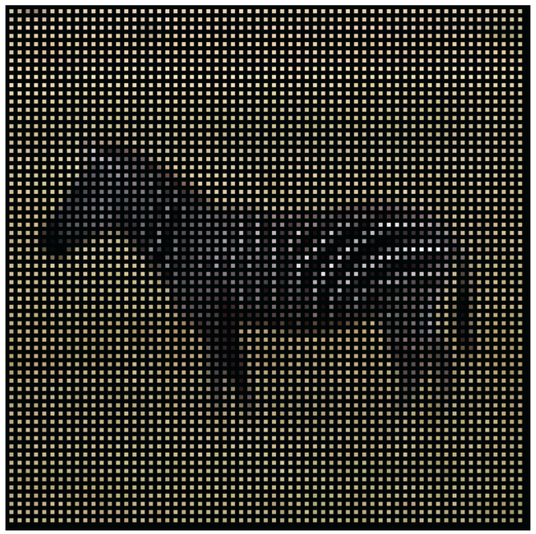
Upsampling

64x64



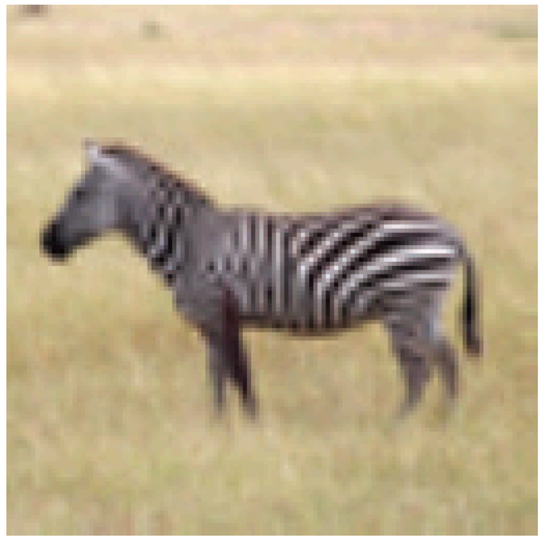
Insert zeros

128x128

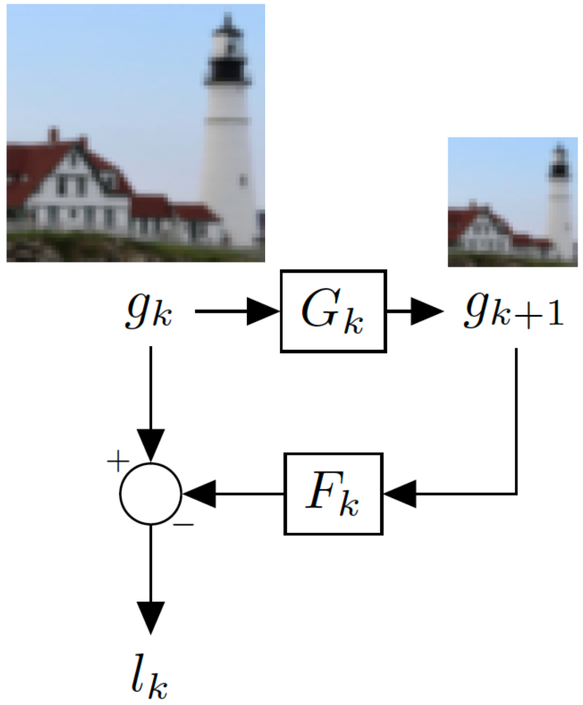


$\bigcirc \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{bmatrix} =$

128x128



The Laplacian Pyramid operators in matrix form



Blurring and downsampling:

$$G_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \frac{1}{16} \begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \end{bmatrix}$$

(Downsampling by 2) (blur)

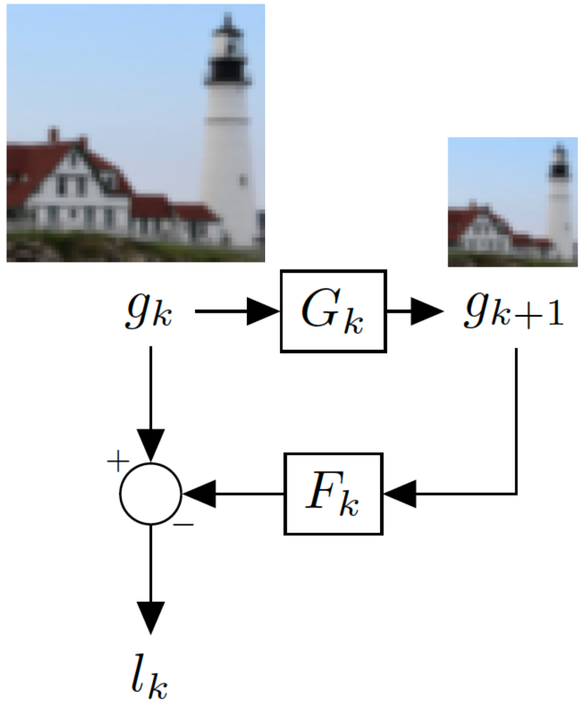
Upsampling and blurring:

$$F_0 = \frac{1}{8} \begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(blur) (Upsampling by 2)

$$l_0 = (I_0 - F_0 G_0) g_0$$

The Laplacian Pyramid

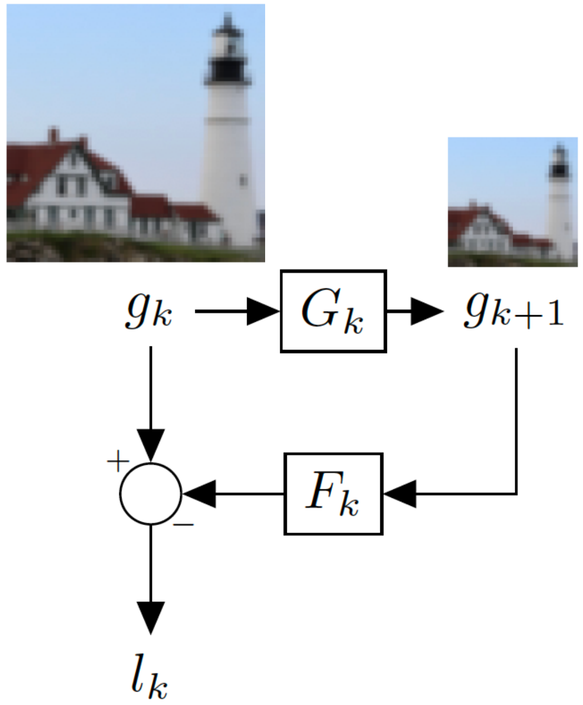




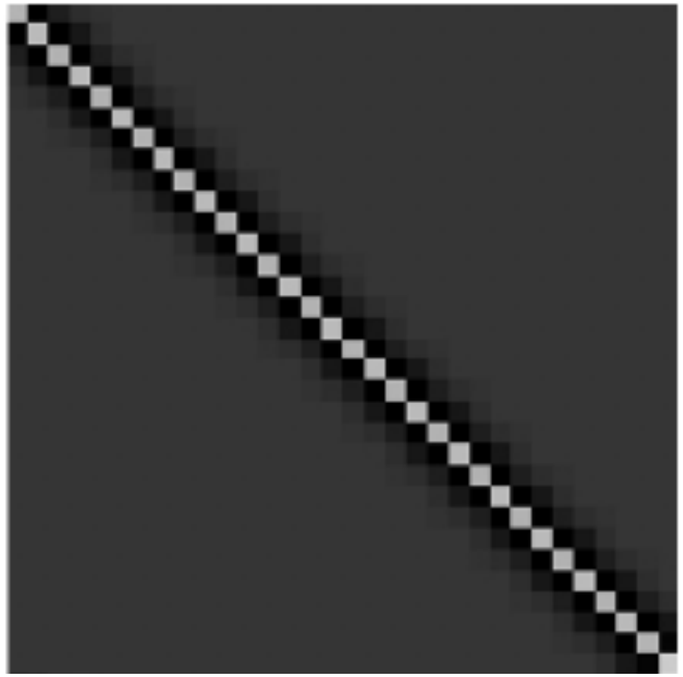
$$l_0 = (I_0 - F_0 G_0) g_0$$

$$= \frac{1}{256} \begin{bmatrix} 182 & -56 & -24 & -8 & -2 & 0 & 0 & 0 \\ -56 & 192 & -56 & -32 & -8 & 0 & 0 & 0 \\ -24 & -56 & 180 & -56 & -24 & -8 & -2 & 0 \\ -8 & -32 & -56 & 192 & -56 & -32 & -8 & 0 \\ -2 & -8 & -24 & -56 & 180 & -56 & -24 & -8 \\ 0 & 0 & -8 & -32 & -56 & 192 & -56 & -32 \\ 0 & 0 & -2 & -8 & -24 & -56 & 182 & -48 \\ 0 & 0 & 0 & 0 & -8 & -32 & -48 & 224 \end{bmatrix} x$$

The Laplacian Pyramid



$$l_0 = (I_0 - F_0 G_0) g_0$$



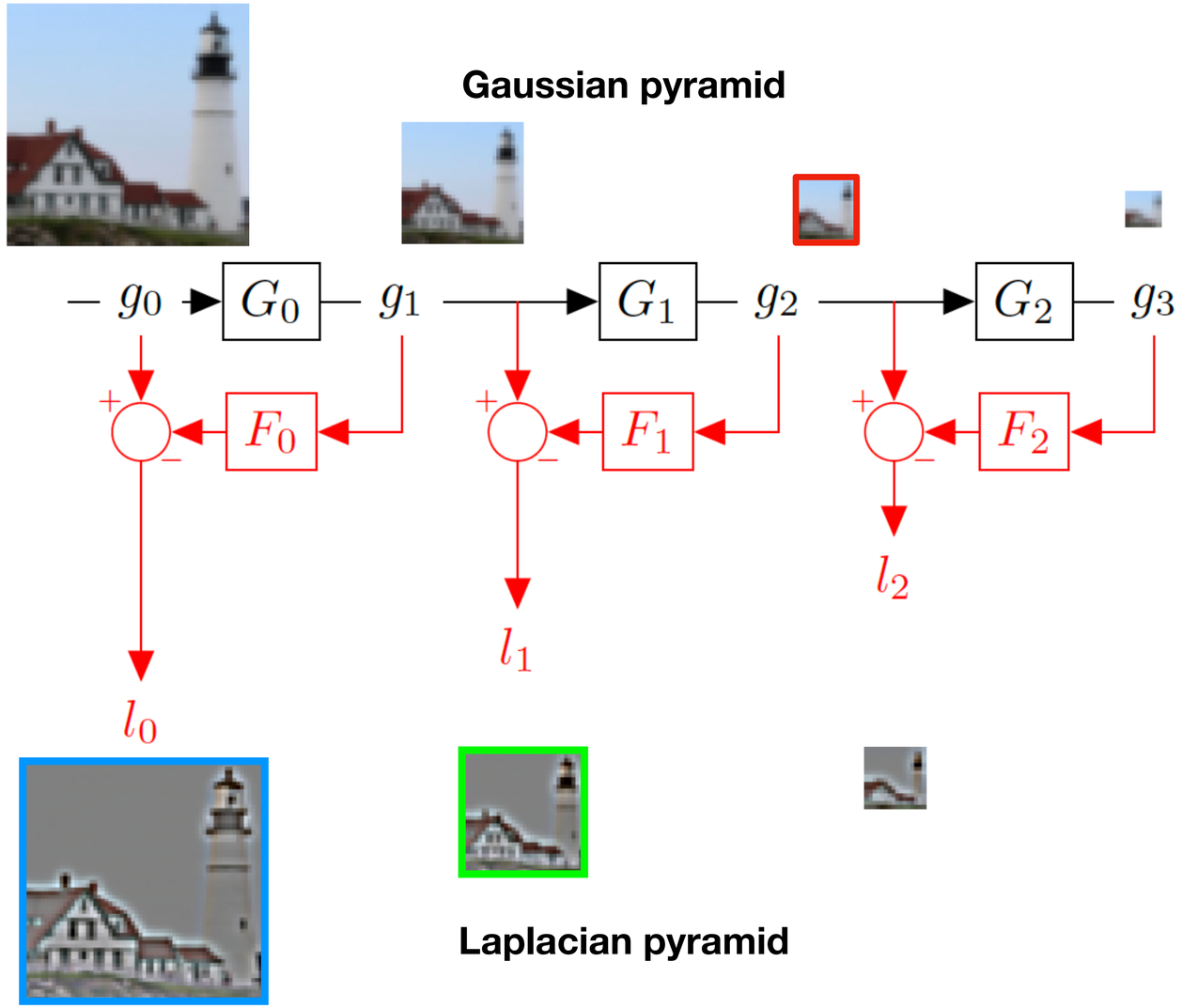
$$\begin{bmatrix}
182 & -56 & -24 & -8 & -2 & 0 & 0 & 0 \\
-56 & 192 & -56 & -32 & -8 & 0 & 0 & 0 \\
-24 & -56 & 180 & -56 & -24 & -8 & -2 & 0 \\
-8 & -32 & -56 & 192 & -56 & -32 & -8 & 0 \\
-2 & -8 & -24 & -56 & 180 & -56 & -24 & -8 \\
0 & 0 & -8 & -32 & -56 & 192 & -56 & -32 \\
0 & 0 & -2 & -8 & -24 & -56 & 182 & -48 \\
0 & 0 & 0 & 0 & -8 & -32 & -48 & 224
\end{bmatrix}$$

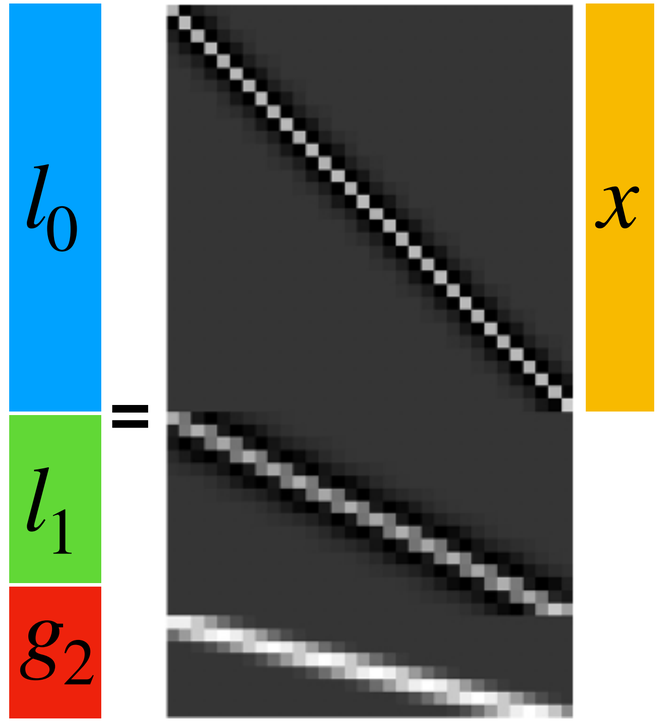


"The Laplace Operator": How much does the value of a function differ from the average of its neighbors?

Well, an approximation of the Laplacian by a difference of Gaussians

The Laplacian Pyramid shown in matrix form



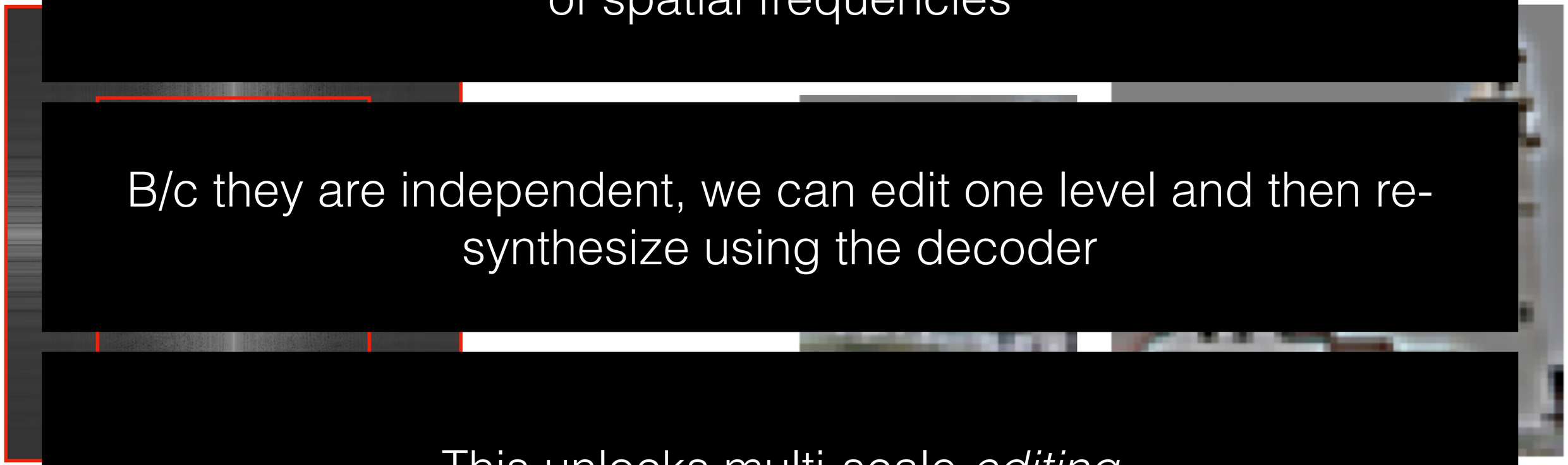


Laplacian Pyramid - Frequency Perspective

Essentially, each level of the laplacian pyramid is a particular band of spatial frequencies

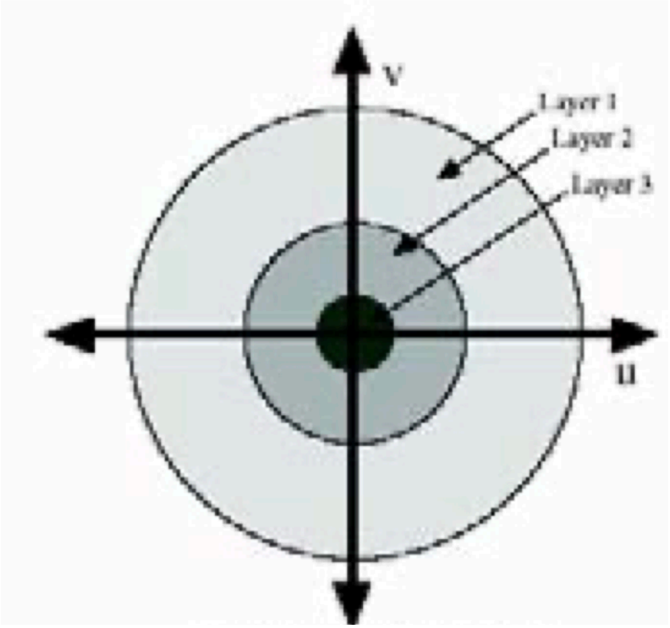
B/c they are independent, we can edit one level and then re-synthesize using the decoder

This unlocks multi-scale *editing*



Locality in space vs. frequency

- You can’t get both perfectly
- Laplacian pyramids are a good tradeoff
- Subdivide the frequency domain into discrete concentric bands
  - The rest is spatially encoded



Laplacian pyramid in Fourier space

Inverting the Laplacian Pyramid









Gaussian residual

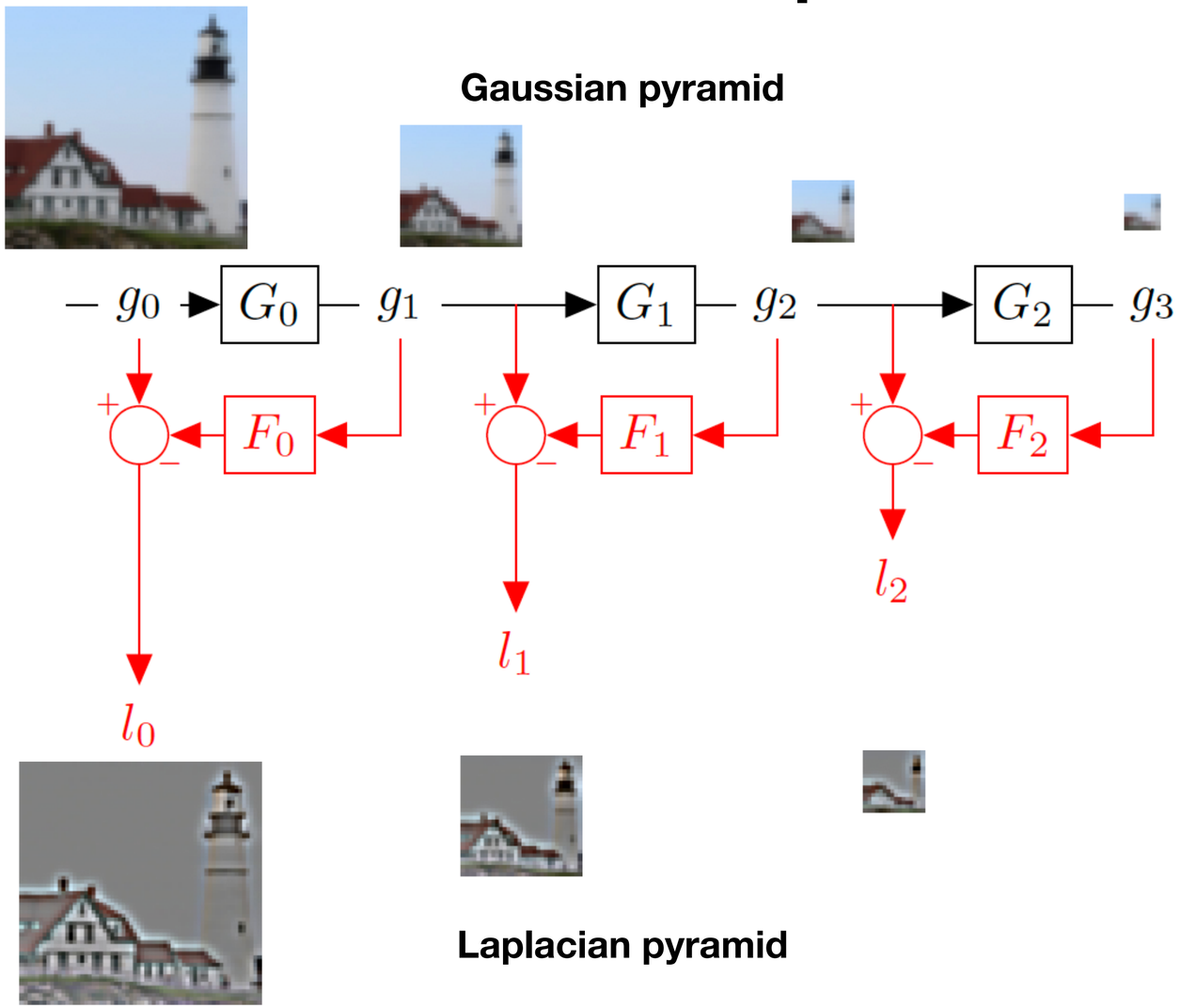
?



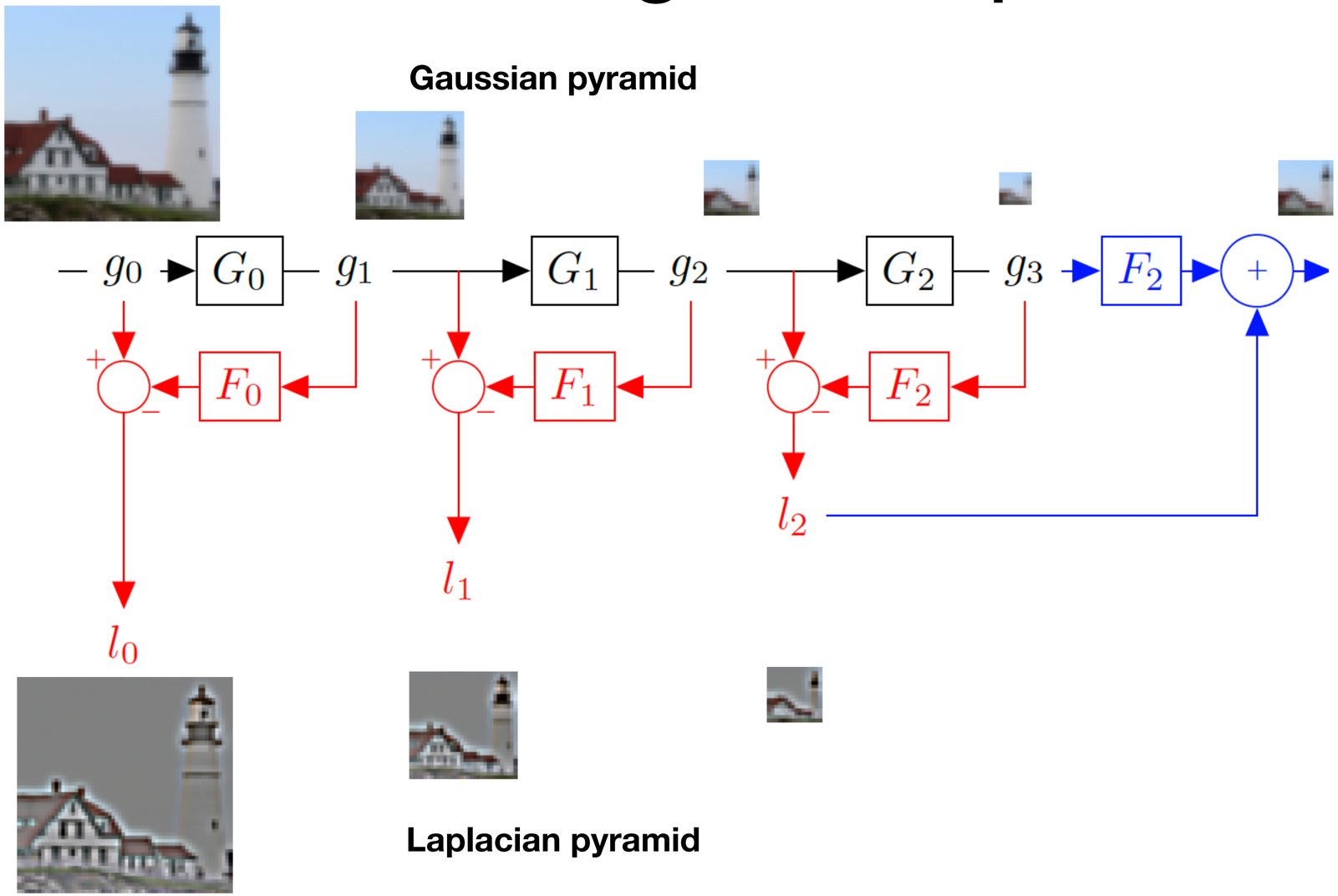
Laplacian pyramid

Can we invert the Laplacian Pyramid?

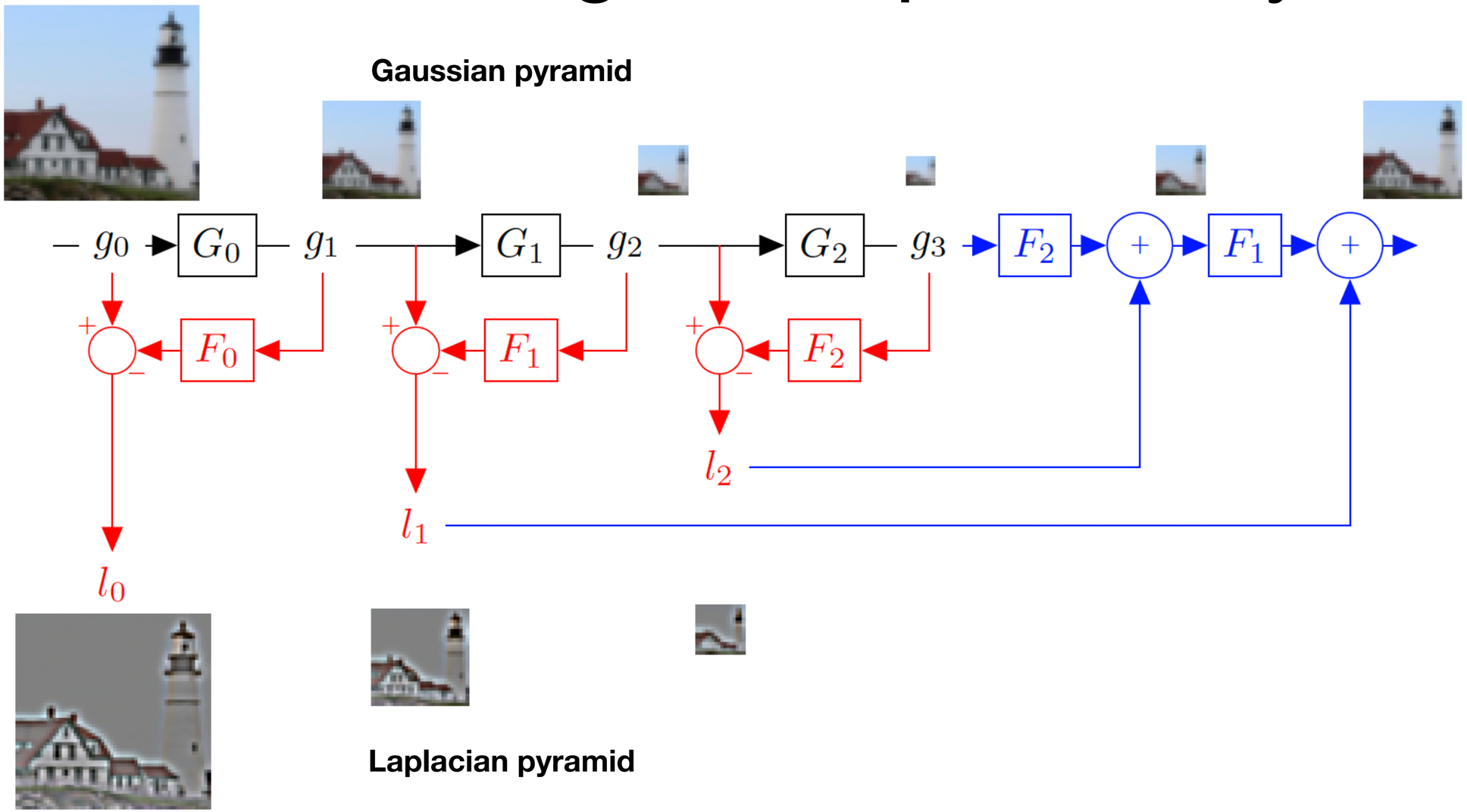
Laplacian Pyramid



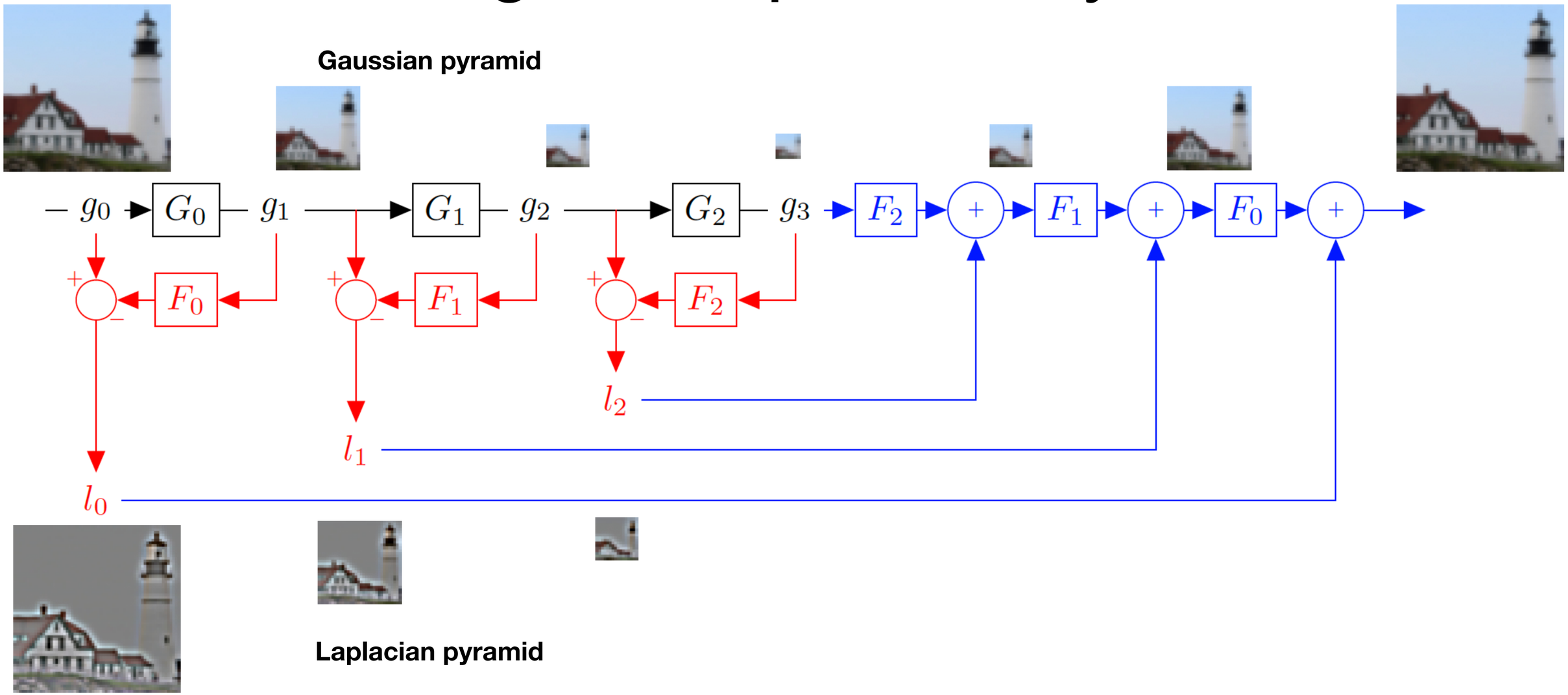
Inverting the Laplacian Pyramid



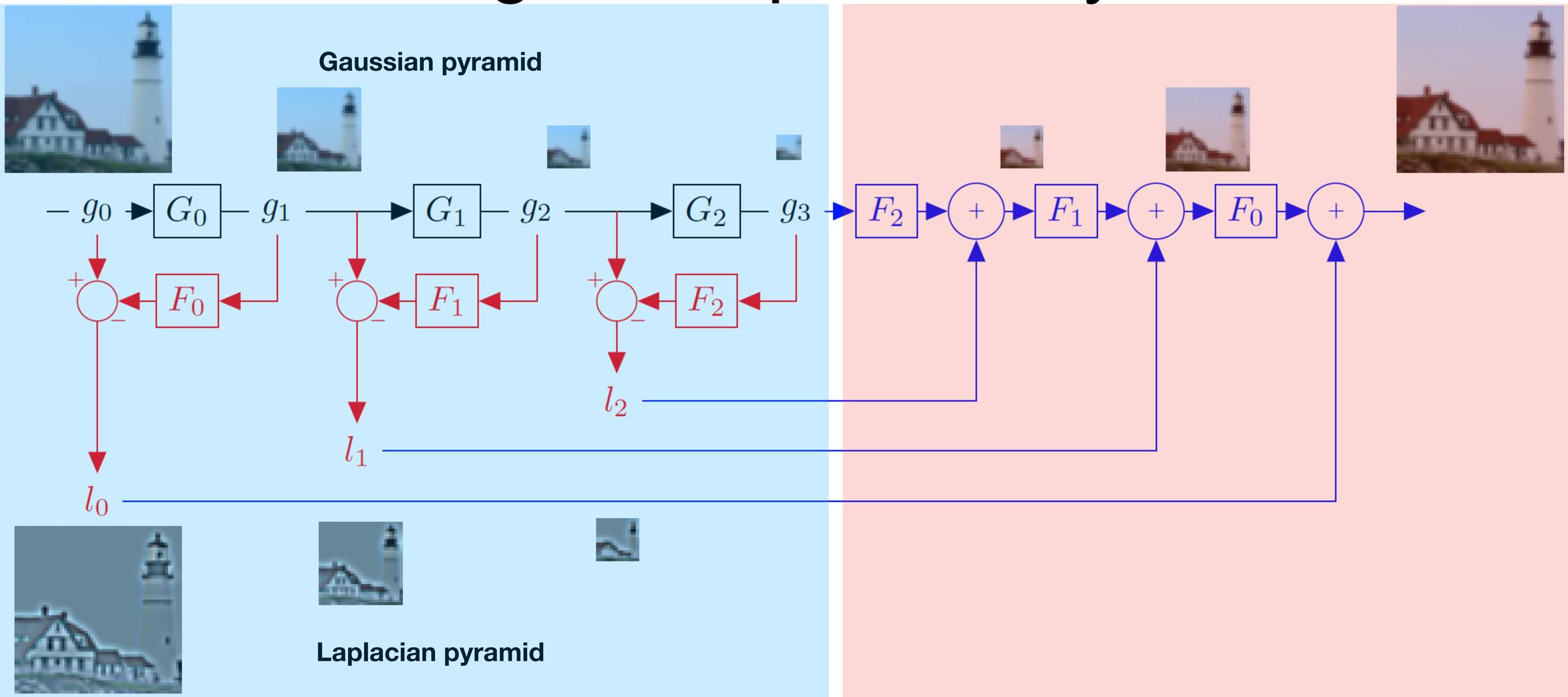
Inverting the Laplacian Pyramid



Inverting the Laplacian Pyramid

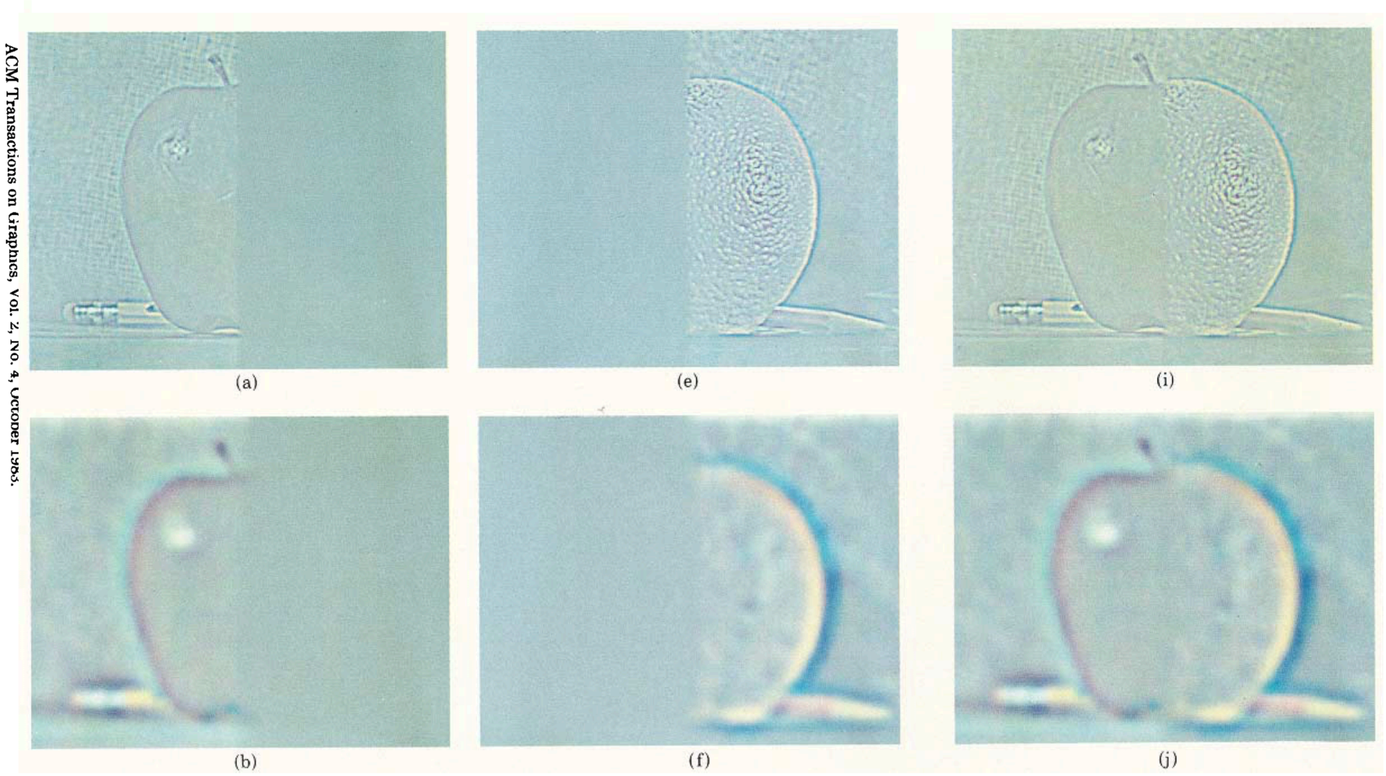


Inverting the Laplacian Pyramid



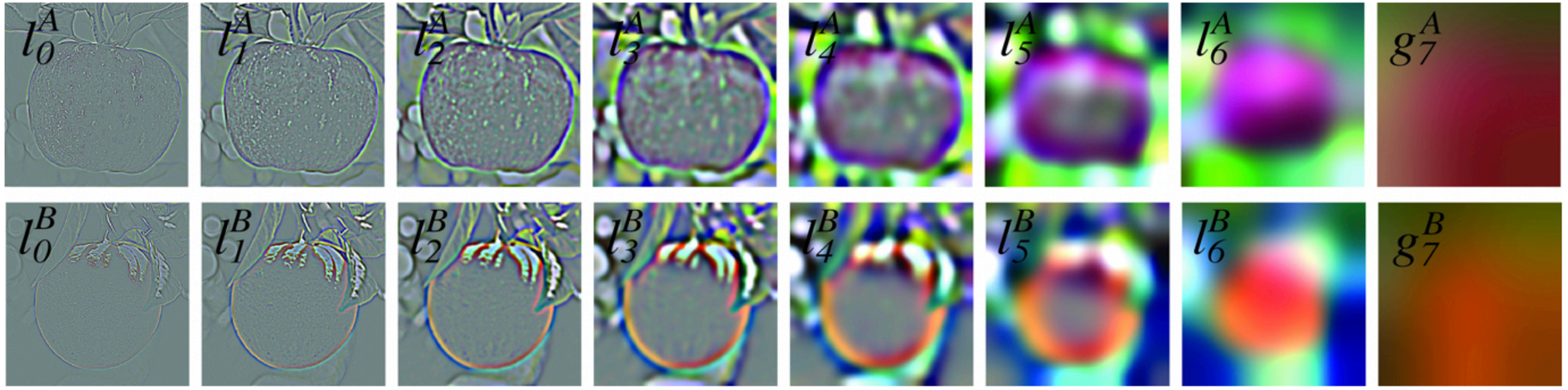
Application: Pyramid blending

- Transition width proportional to pyramid level
  - Decompose mask with Gaussian (NOT Laplacian) Pyramid

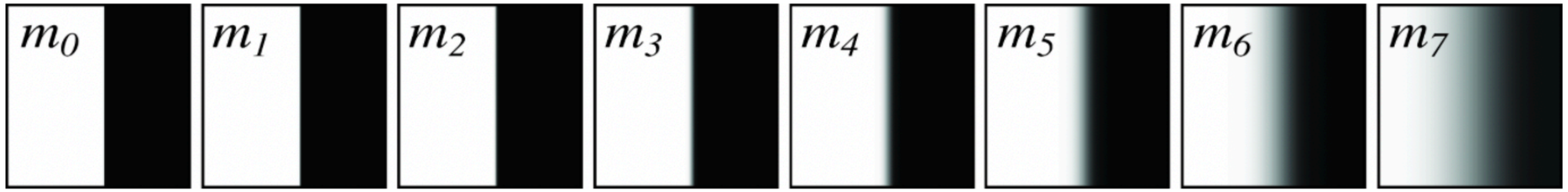


ACM Transactions on Graphics, Vol. 2, No. 4, October 1983.

Fig. 10. Contributions of various band-pass filtered component images to the apple-orange mosaic. Figures 10a–10c show high-, medium- and low-frequency components of the orange obtained by expanding levels 0, 2 and 4 of its Laplacian pyramid (in these figures, a constant has been added to each band-pass image so that zero values appear gray and negative values are dark. Image contrast has also been increased to improve visibility.) Note that low-frequency components



Sources images (LAPLACIAN pyramid)



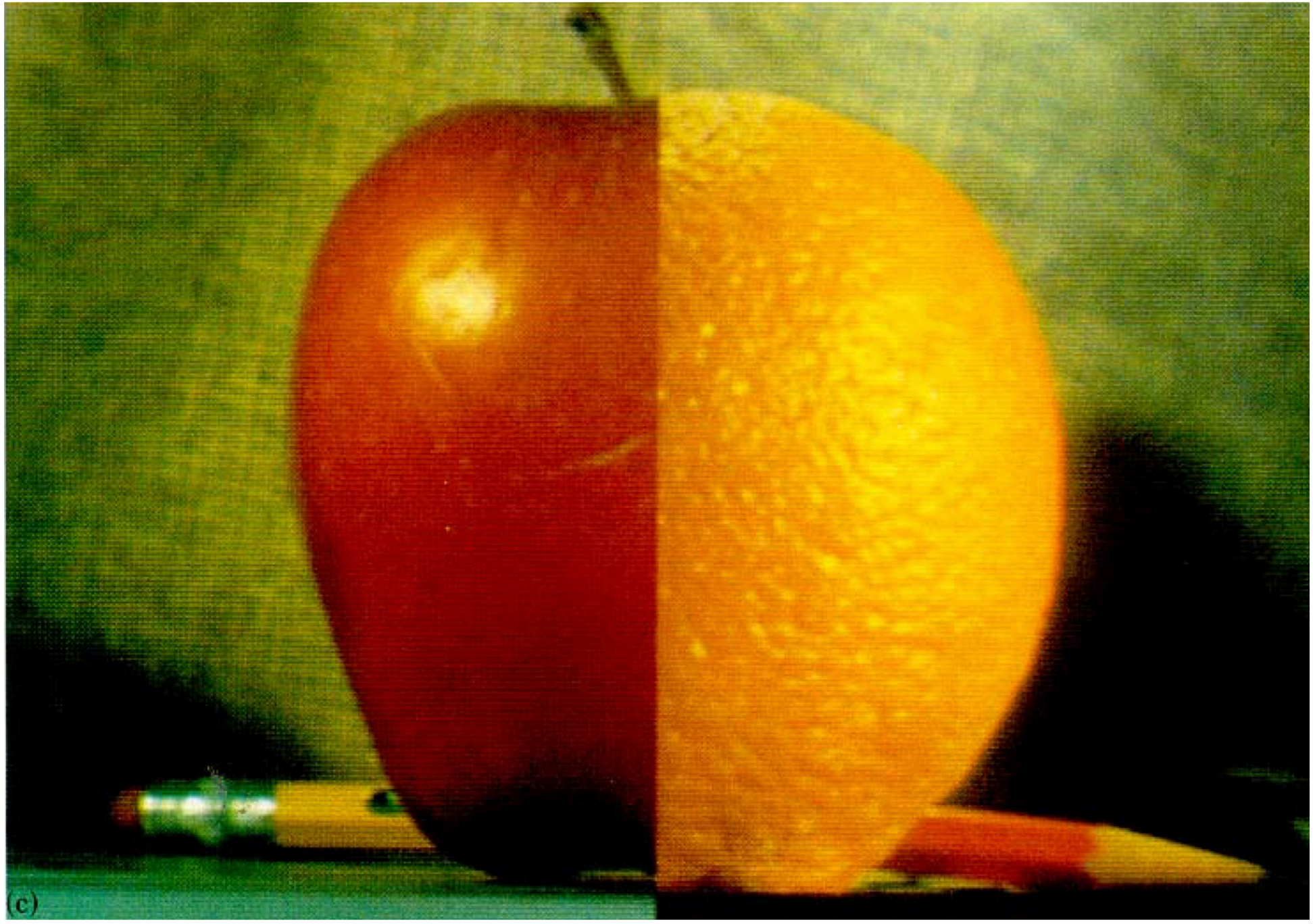
Mask (GAUSSIAN pyramid)

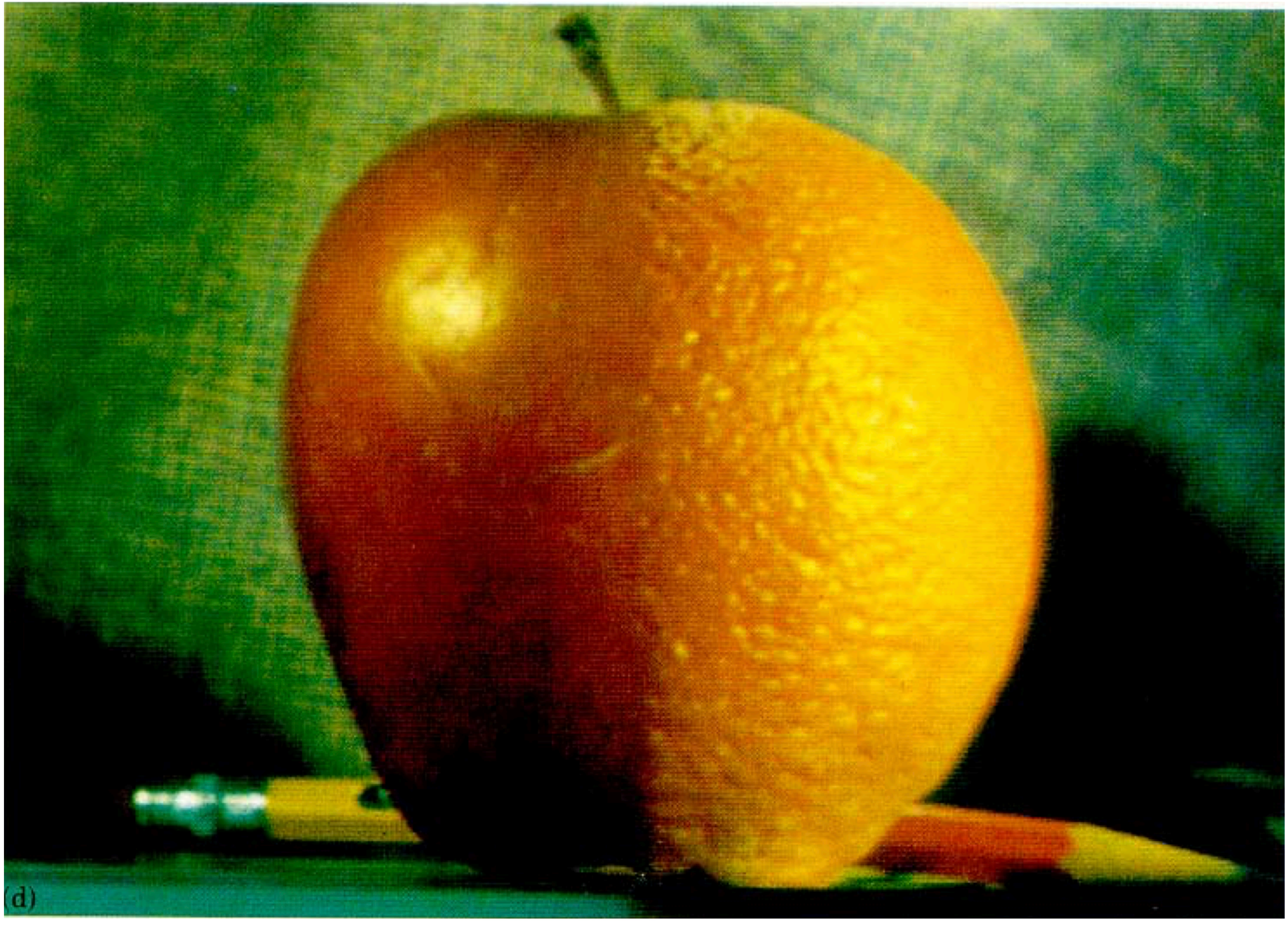


For each output coefficient:

Output = (1-mask) im1 + mask*im2

https://visionbook.mit.edu/pyramids_new_notation.html

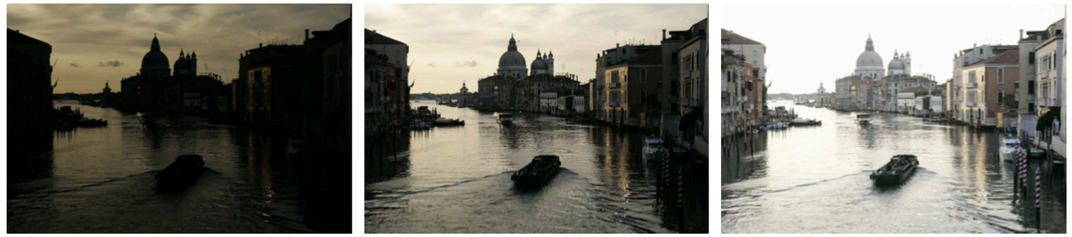

(c)


(d)


a
b
c
d

HDR tone mapping

- Kautz & Mertens
- Idea: combine HDR merging & tone mapping
- Laplacian pyramid blending

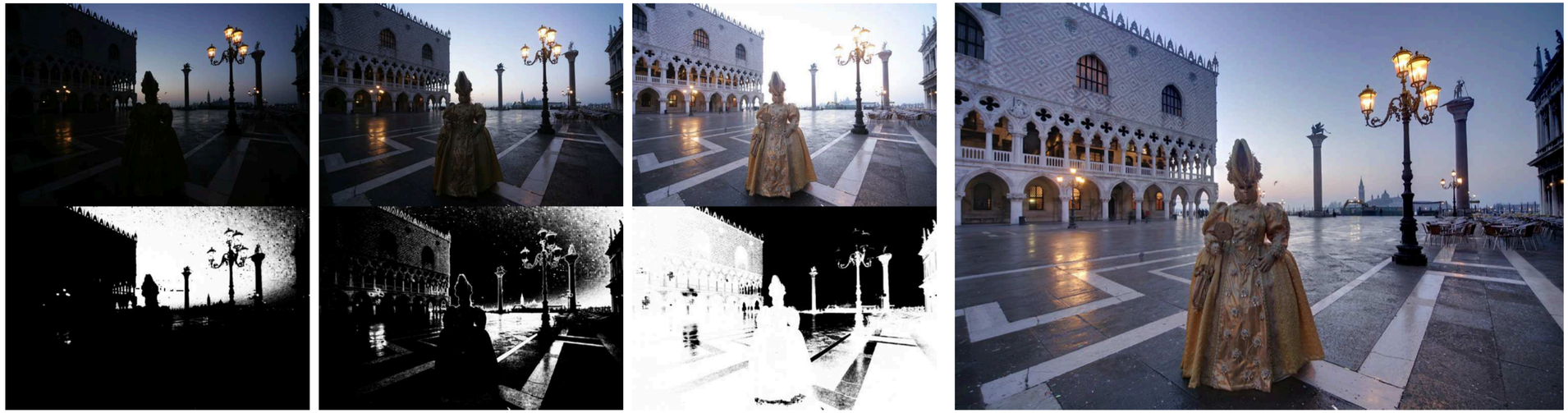


(a) Exposure bracketed sequence



(b) Fused result

https://web.stanford.edu/class/cs231m/project-1/exposure-fusion.pdf



(a) Input images with corresponding weight maps

(b) Fused result

HDR+ on Android

- http://static.googleusercontent.com/media/hdrplusdata.org/en//hdrplus.pdf
- https://ai.googleblog.com/2018/02/introducing-hdr-burst-photography.html
- Uses HDR Laplacian exposure fusion

Burst photography for high dynamic range and low-light imaging on mobile cameras

Samuel W. Hasinoff Dillon Sharlet Ryan Geiss Andrew Adams
Jonathan T. Barron Florian Kainz Jiawen Chen Marc Levoy
Google Research



Extension: Local Laplacian Filtering

- https://people.csail.mit.edu/sparis/publi/2011/siggraph/
- https://people.csail.mit.edu/sparis/publi/2014/tog/Aubry_14-Fast_Local_Laplacian_Filters.pdf

Local Laplacian Filters: Edge-aware Image Processing with a Laplacian Pyramid

Sylvain Paris
Adobe Systems, Inc.

Samuel W. Hasinoff
Toyota Technological Institute at Chicago and MIT CSAIL

Jan Kautz
University College London



(a) input HDR image tone-mapped with a simple gamma curve (details are compressed)

(b) our pyramid-based tone mapping, set to preserve details without increasing them

(c) our pyramid-based tone mapping, set to strongly enhance the contrast of details

Figure 1: *We demonstrate edge-aware image filters based on the direct manipulation of Laplacian pyramids. Our approach produces high-quality results, without degrading edges or introducing halos, even at extreme settings. Our approach builds upon standard image pyramids and enables a broad range of effects via simple point-wise nonlinearities (shown in corners). For an example image (a), we show results of tone mapping using our method, creating a natural rendition (b) and a more exaggerated look that enhances details as well (c). Laplacian pyramids have previously been considered unsuitable for such tasks, but our approach shows otherwise.*

Local Laplacian filtering

- Inspired by exposure fusion but for a single HDR input
- Motivation: exposure fusion depends on the set of images and if you take too many of them the algorithm reduces to a very uniform image
- Can be seen as generalization of pyramid denoising
- Has replaced bilateral filtering in Adobe products

Example Application: Increase Level of Detail



input

Naïve Solution: Boost High Frequencies



unsharp mask

Taking Edges into Account for Halo-free Result



our result

Pyramid Denoising

Pyramid coring/denoising

- Here : Simoncelli & Adelson
- https://www.cns.nyu.edu/pub/eero/simoncelli96c.pdf
- https://www.cns.nyu.edu/pub/lcv/rajashekar08a.pdf

At a high/simplified level

- Construct Laplacian Pyramid
- Clamp small pyramid coefficients to exactly zero

- Removes noise in smooth areas
- Preserves strong edges

Fancy Bayesian version

- **Bayesian inference : infer value based on observation and prior information**
  - Find solution based on optimizing probability, e.g. max likelihood
- **Likelihood term : output should be within noise value of input**
  - Probability is a Gaussian on difference where sigma should be noise level
- **Prior probability based on regularities in natural images**
- **Main observation: most coefficients are zero**

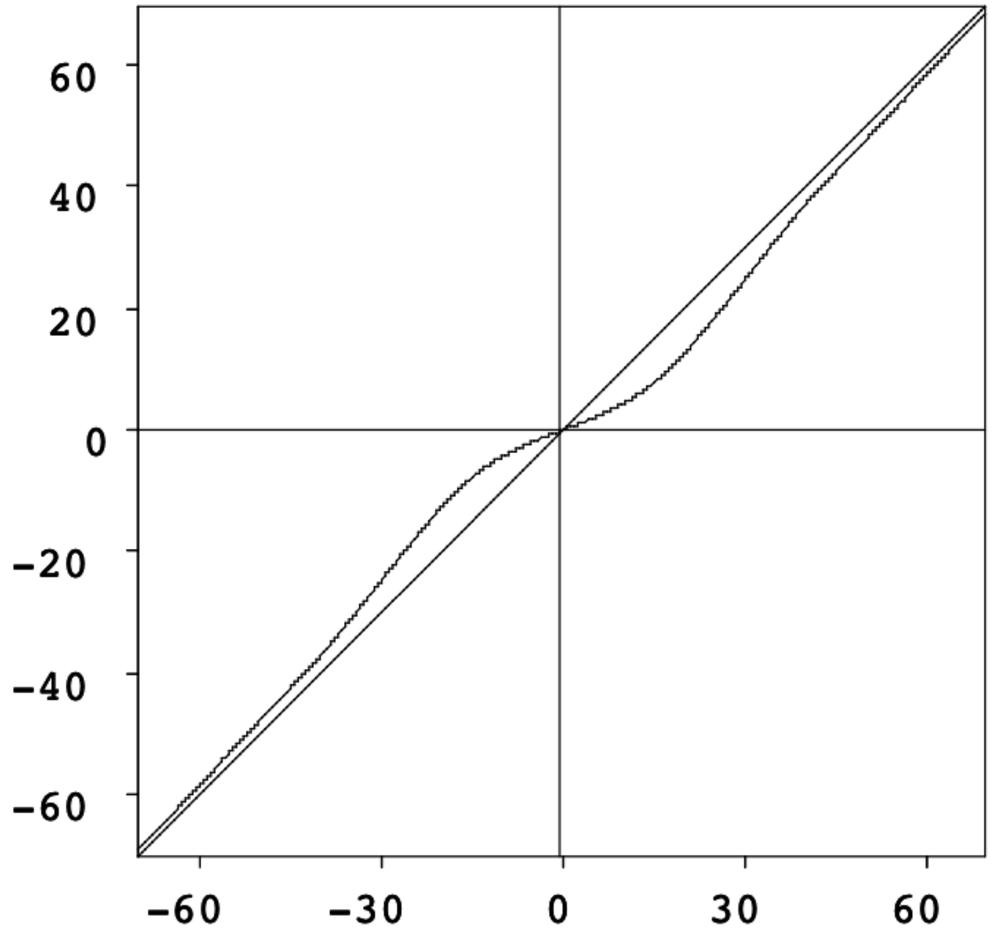


Figure 2. Bayesian estimator (symmetrized) for the signal and noise histograms shown in figure 1. Superimposed on the plot is a straight line indicating the identity function.

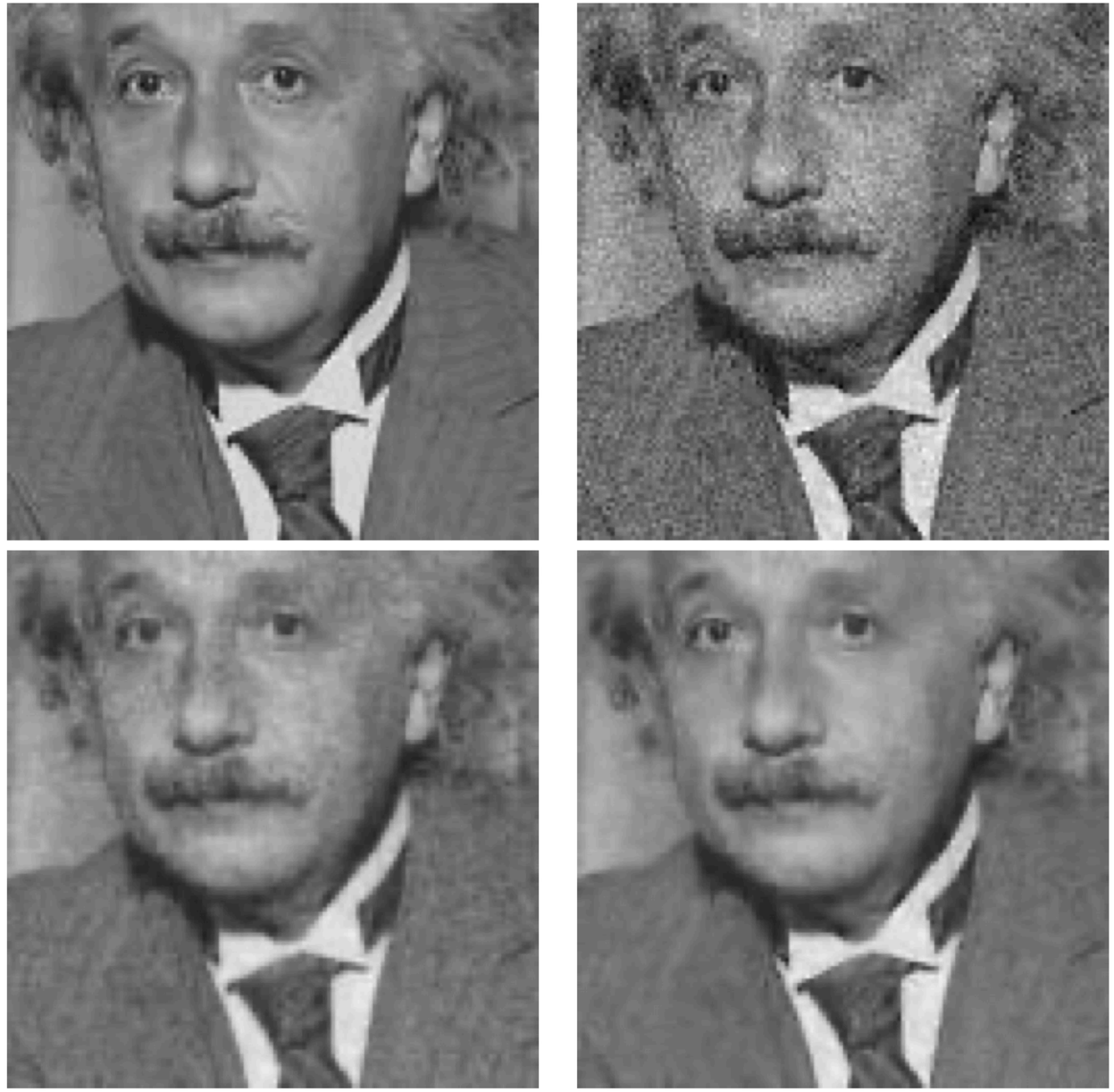


Figure 4. Noise reduction example. (a) Original image (cropped). (b) Image contaminated with additive Gaussian white noise (SNR = 9.00dB). (c) Image restored using (semi-blind) Wiener filter (SNR = 11.88dB). (d) Image restored using (semi-blind) Bayesian estimator (SNR = 13.82dB).

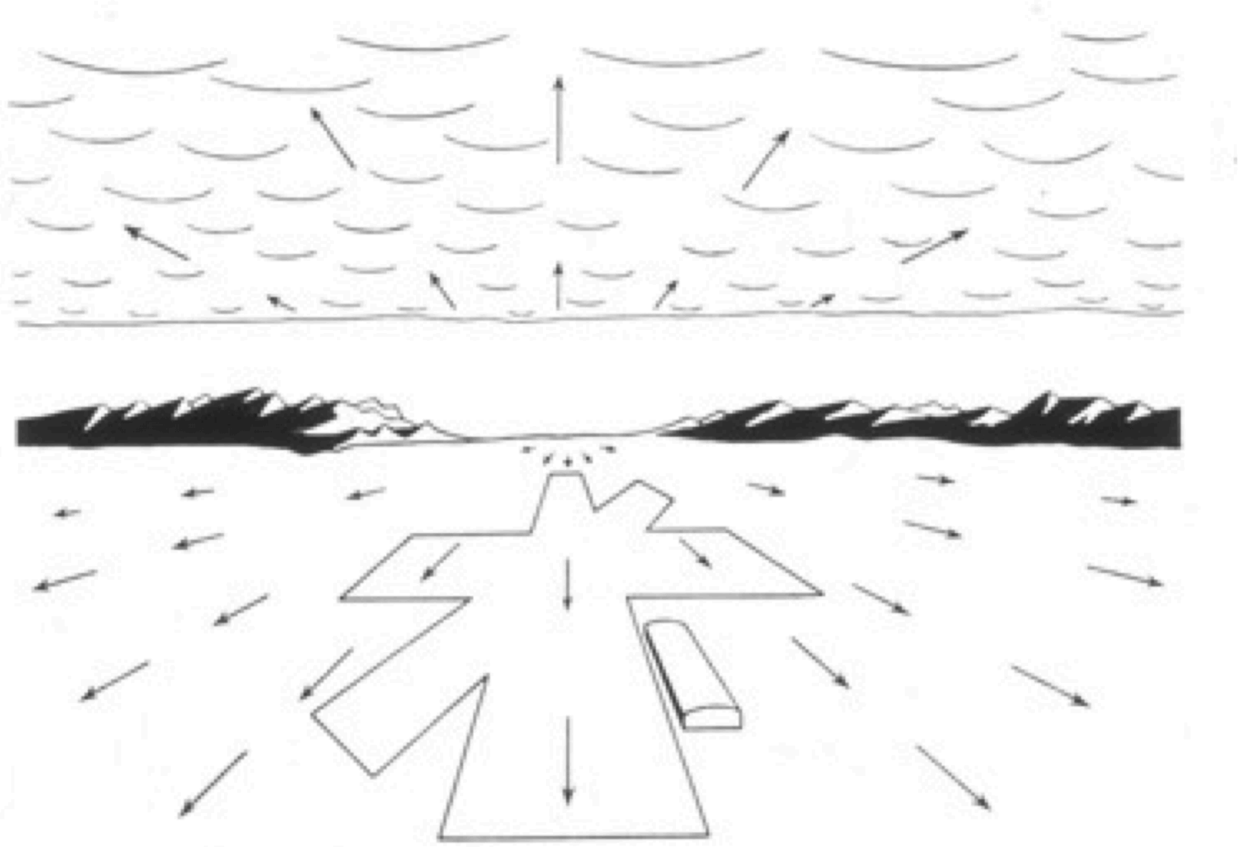
In the real world

- Pyramids also allow us to take into account human contrast sensitivity and other perceptual effects
- People use cookbook recipes which allows them to subjectively tune visual quality
  - They like having different controls at different scales (unlike, say, bilateral filter)
- Used a ton on cell phones

Other multiscale processing

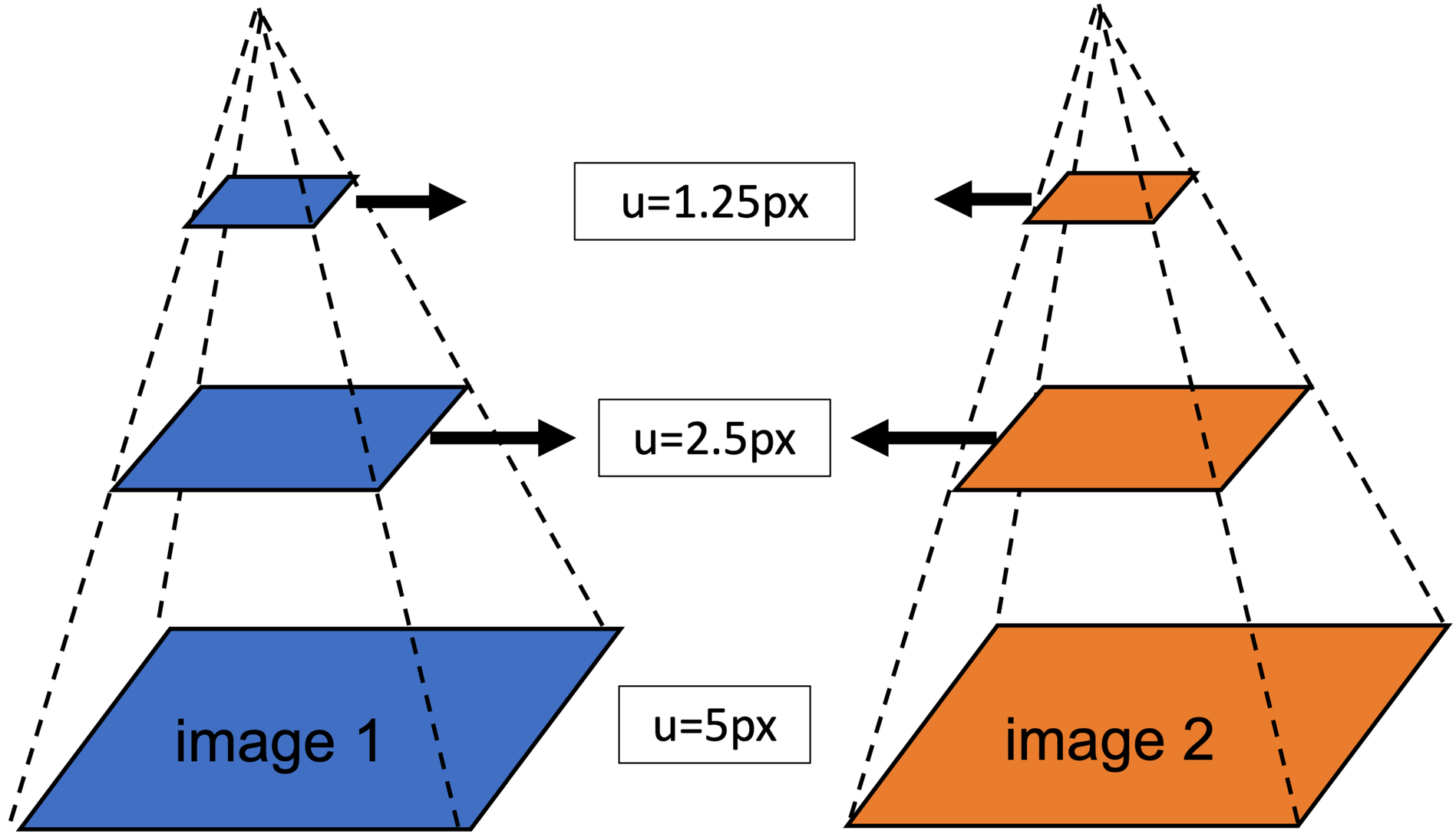
Optical flow

- How do things move locally in image space over time
- First introduce by psychologist Gibson (~1940)



More in two weeks

Coarse-to-fine optical flow estimation



Slide credit: S. Lazebnik

Typically called Gaussian Pyramid

Also SIFT (Scale invariant feature transforms)

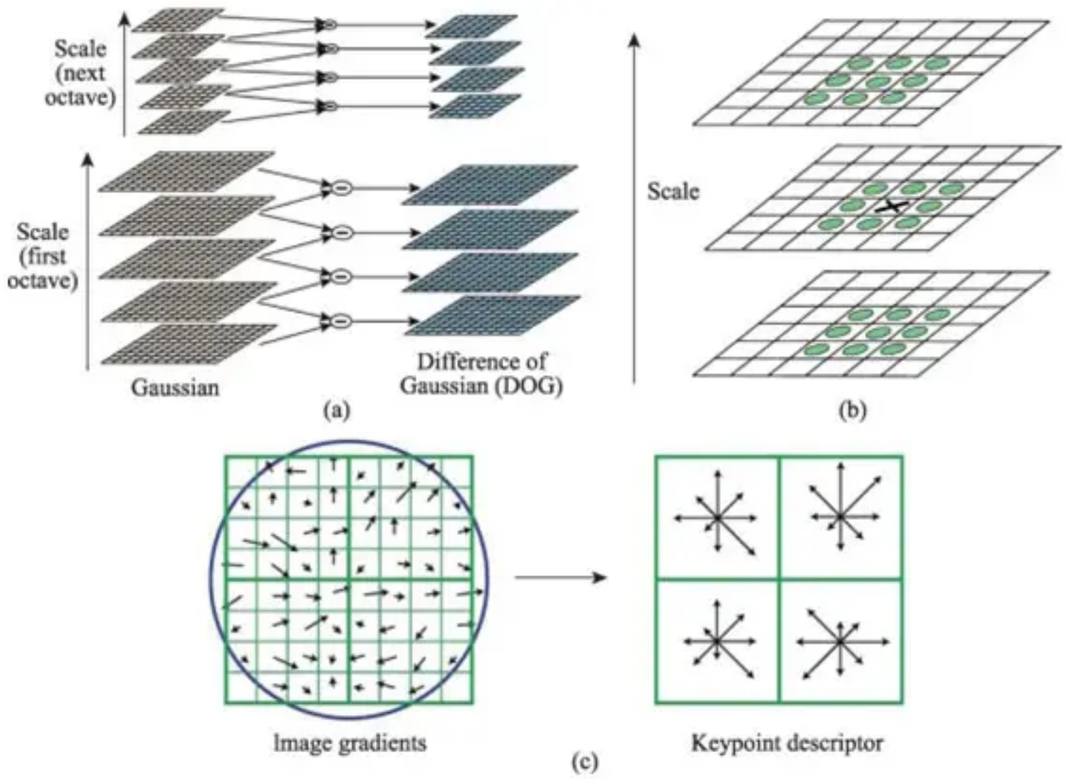
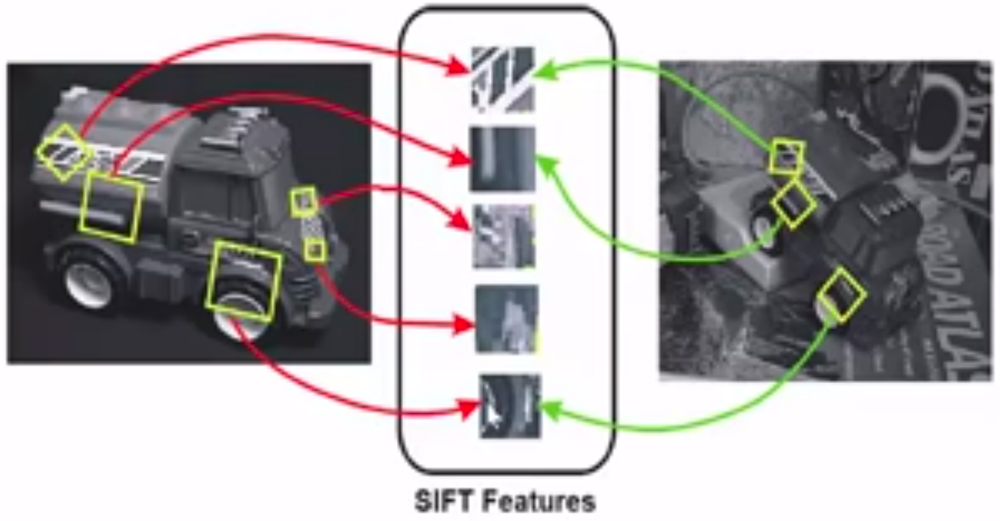
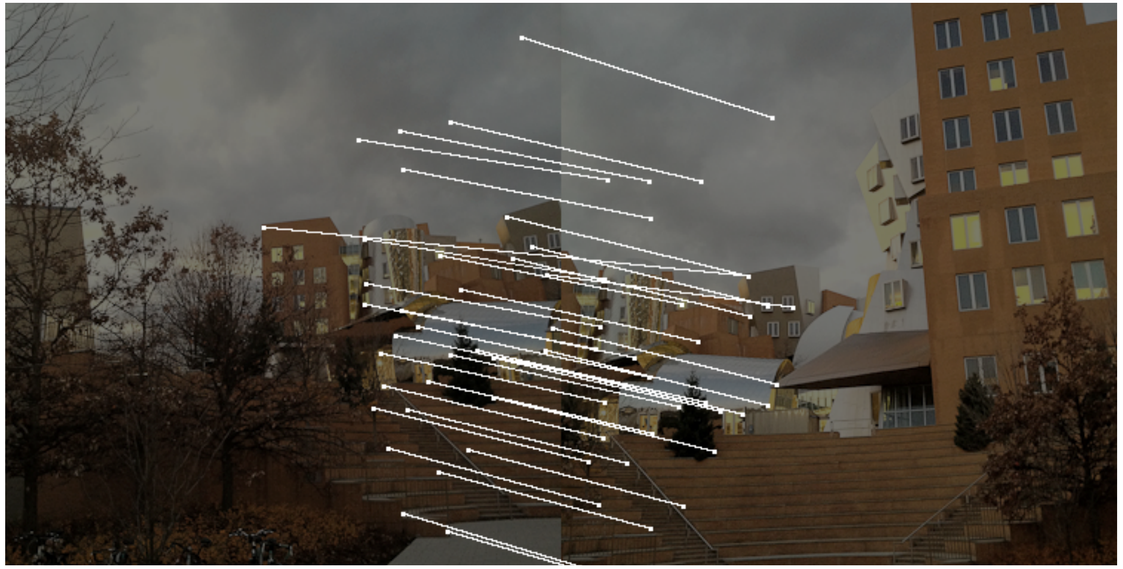


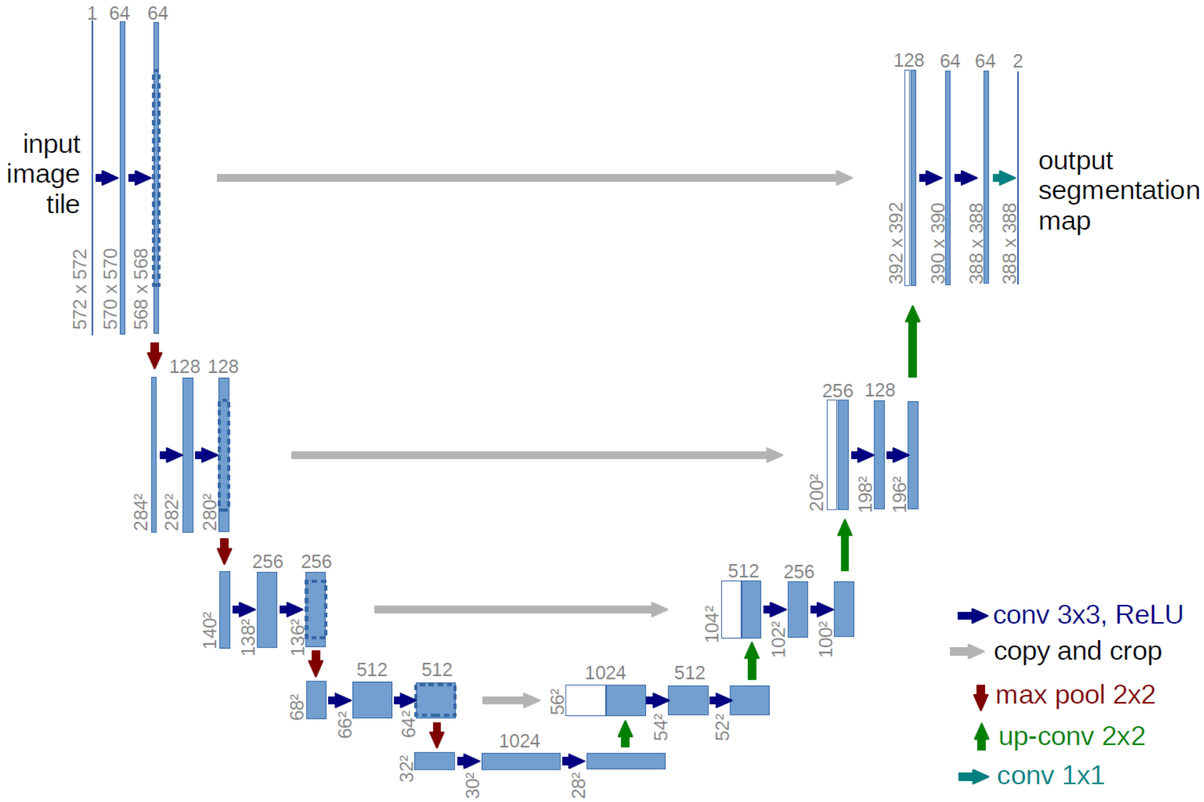
Figure 3: Construction of SIFT descriptor [9] (a) Image pyramid. (b) Extrema detection for DOG pyramid. (c) Creation of keypoint descriptor.



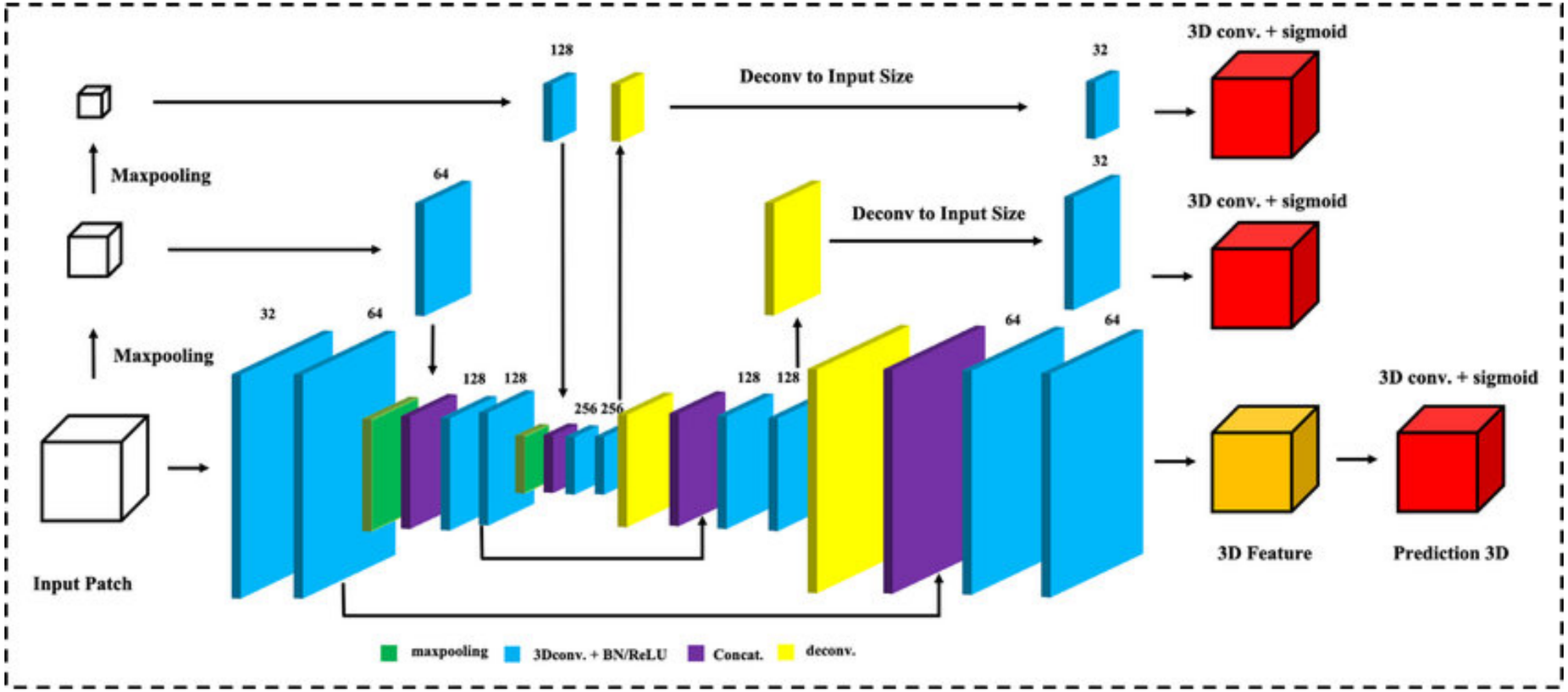


U-Net

- Workhorse of deep learning and generative AI
- In particular Denoising Diffusion Models



U-Net


3D conv. + sigmoid
128
32
Deconv to Input Size
Maxpooling
64
32
3D conv. + sigmoid
Deconv to Input Size
Maxpooling
32
64
64
64
128
128
128
128
256
256
3D conv. + sigmoid
Input Patch
3D Feature
Prediction 3D
maxpooling
3Dconv. + BN/ReLU
Concat.
deconv.

Video magnification



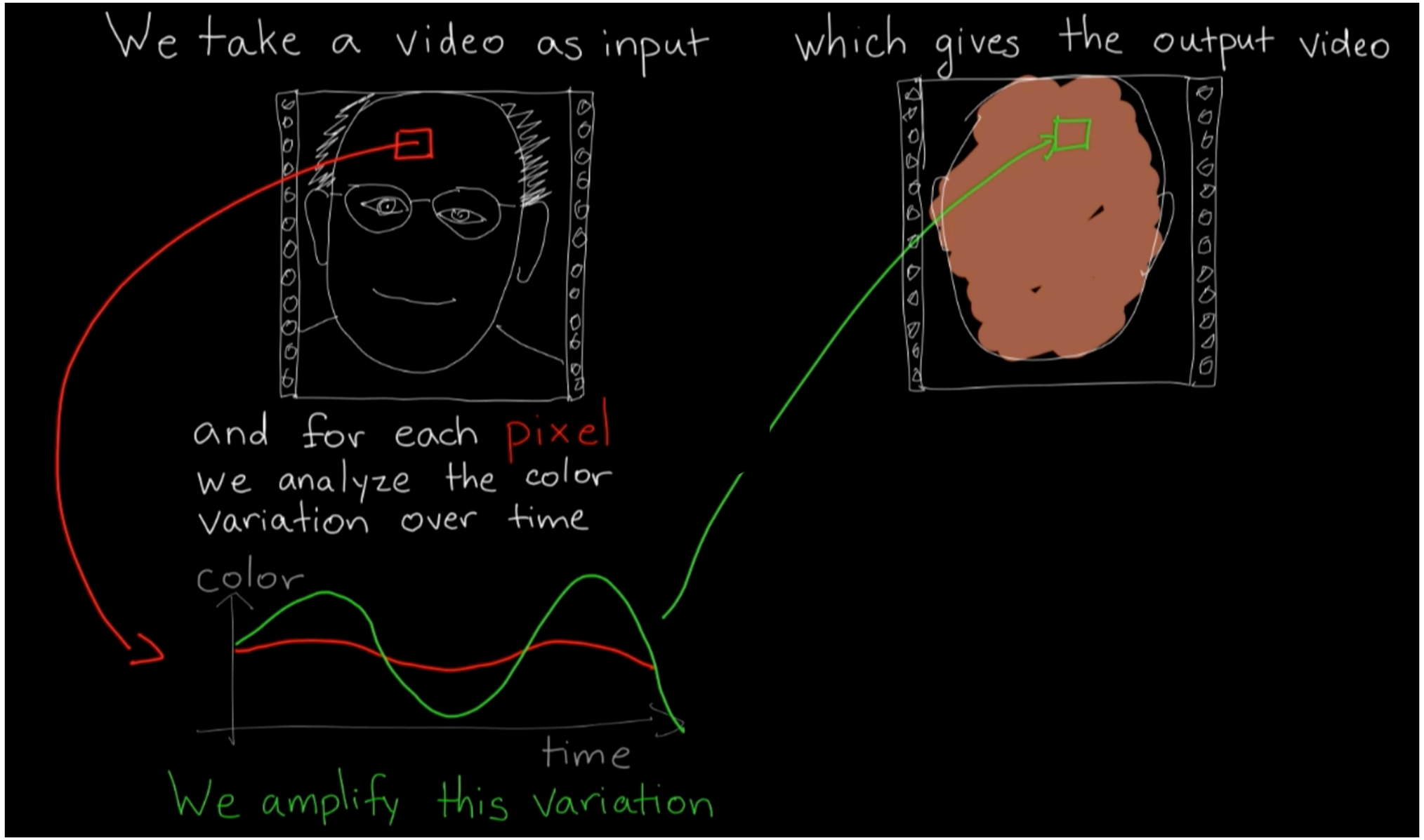
Massachusetts Institute of Technology



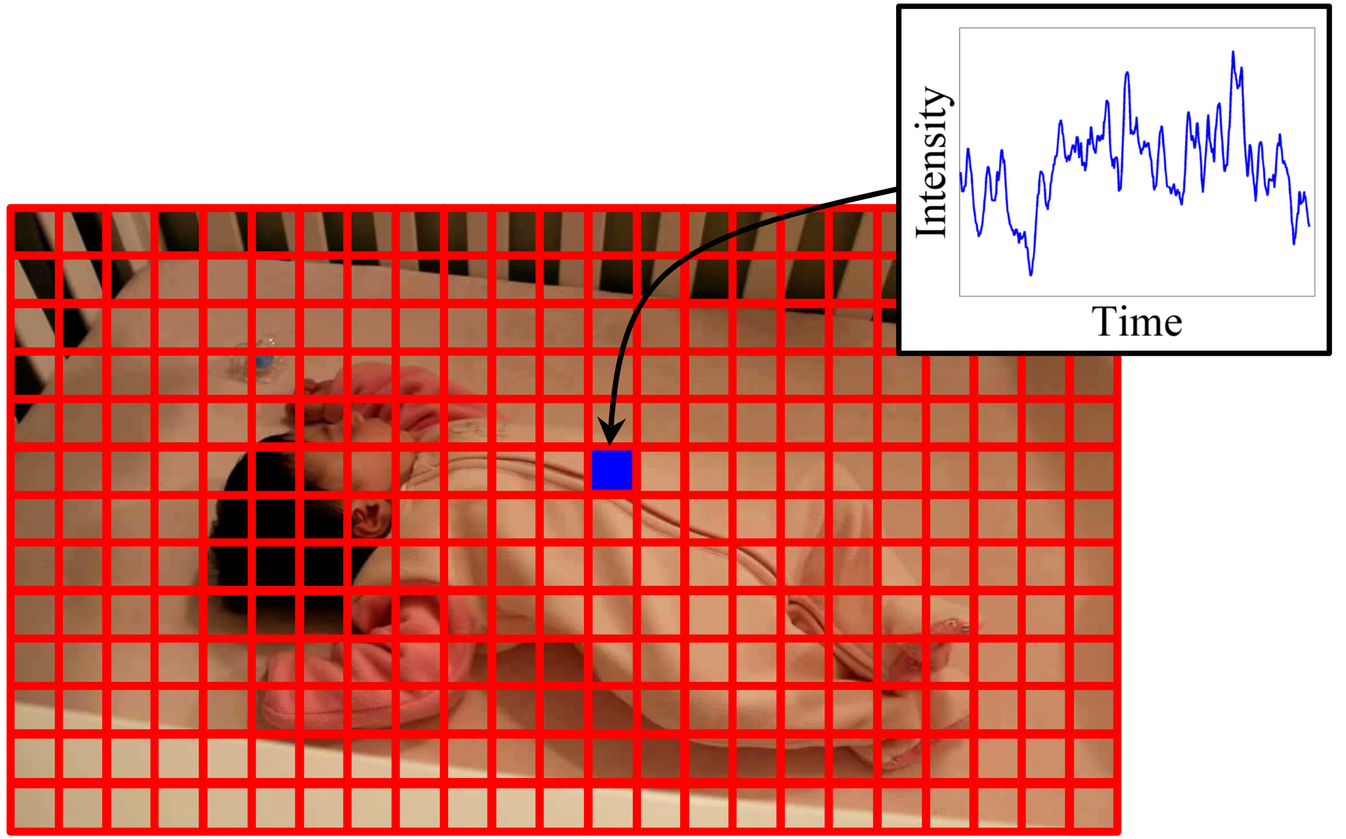
Revealing Invisible Changes In The World

Created for the NSF International Science & Engineering Visualization Challenge 2012

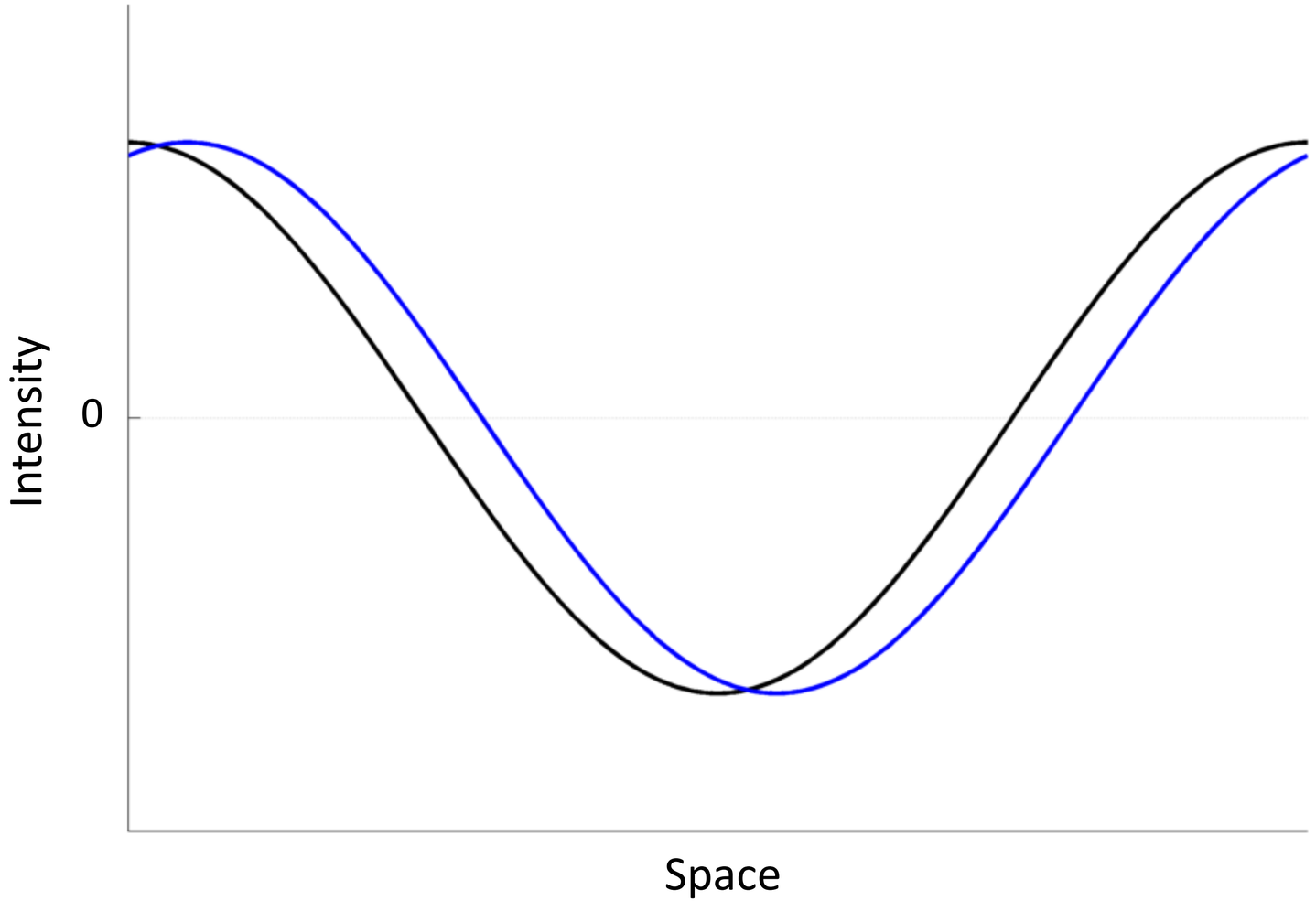
Basic idea


We take a video as input
which gives the output video
and for each pixel
we analyze the color
variation over time
color
time
We amplify this variation

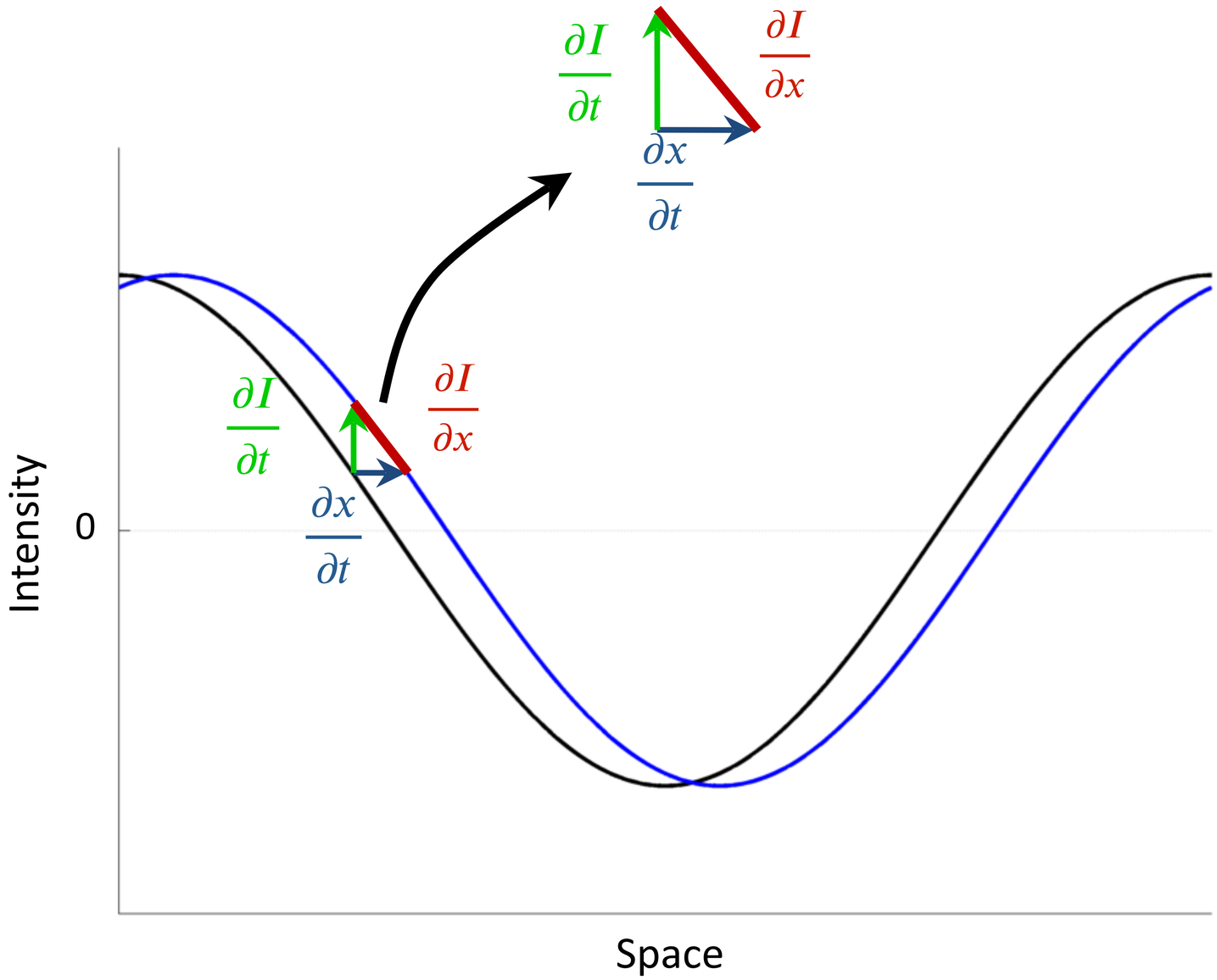
Why It Amplifies Motion?????



Differential Brightness Constancy

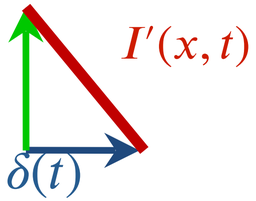


Differential Brightness Constancy



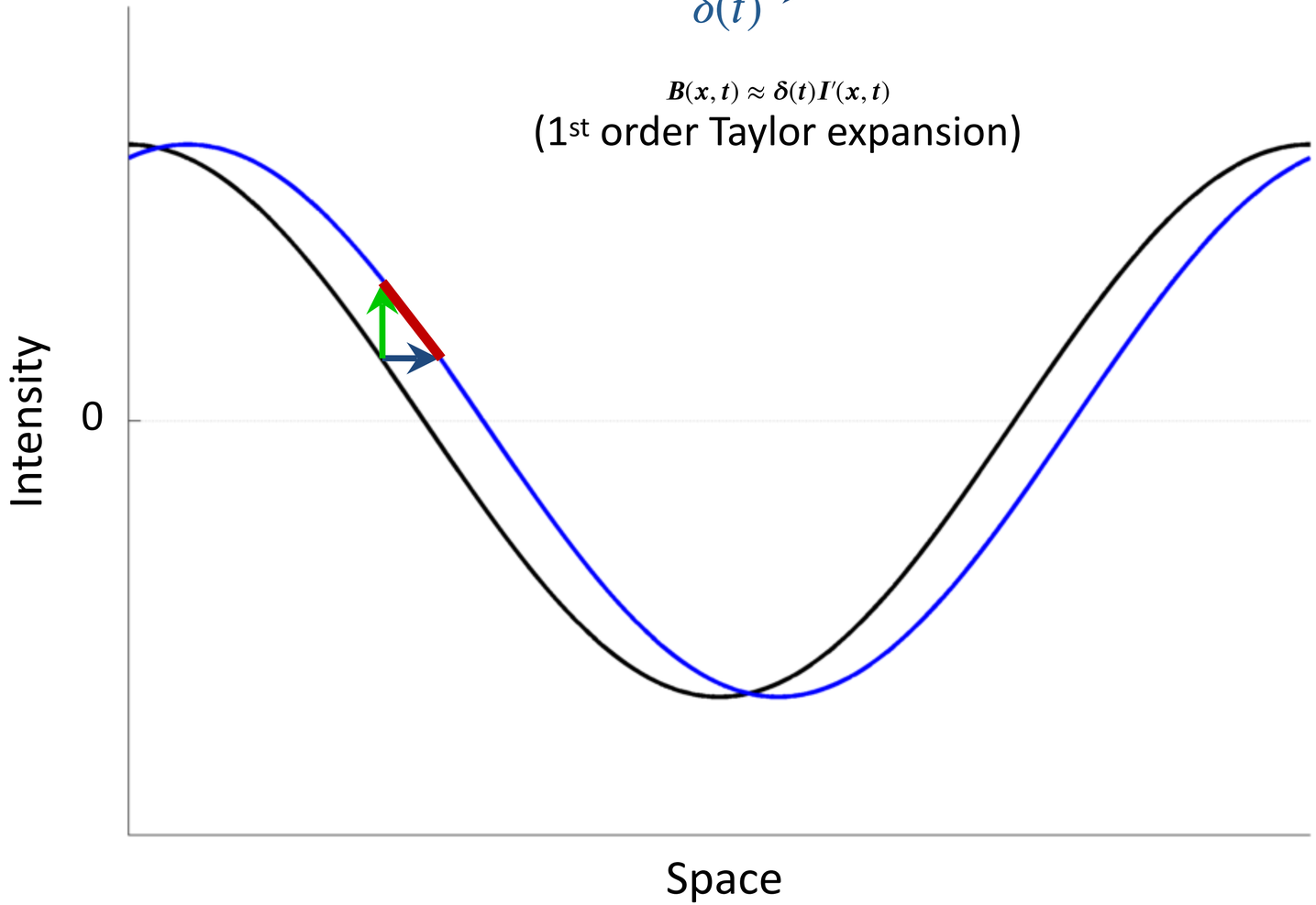
Relating Temporal and Spatial Changes

$$B(x,t) = I(x,t+1) - I(x,t)$$

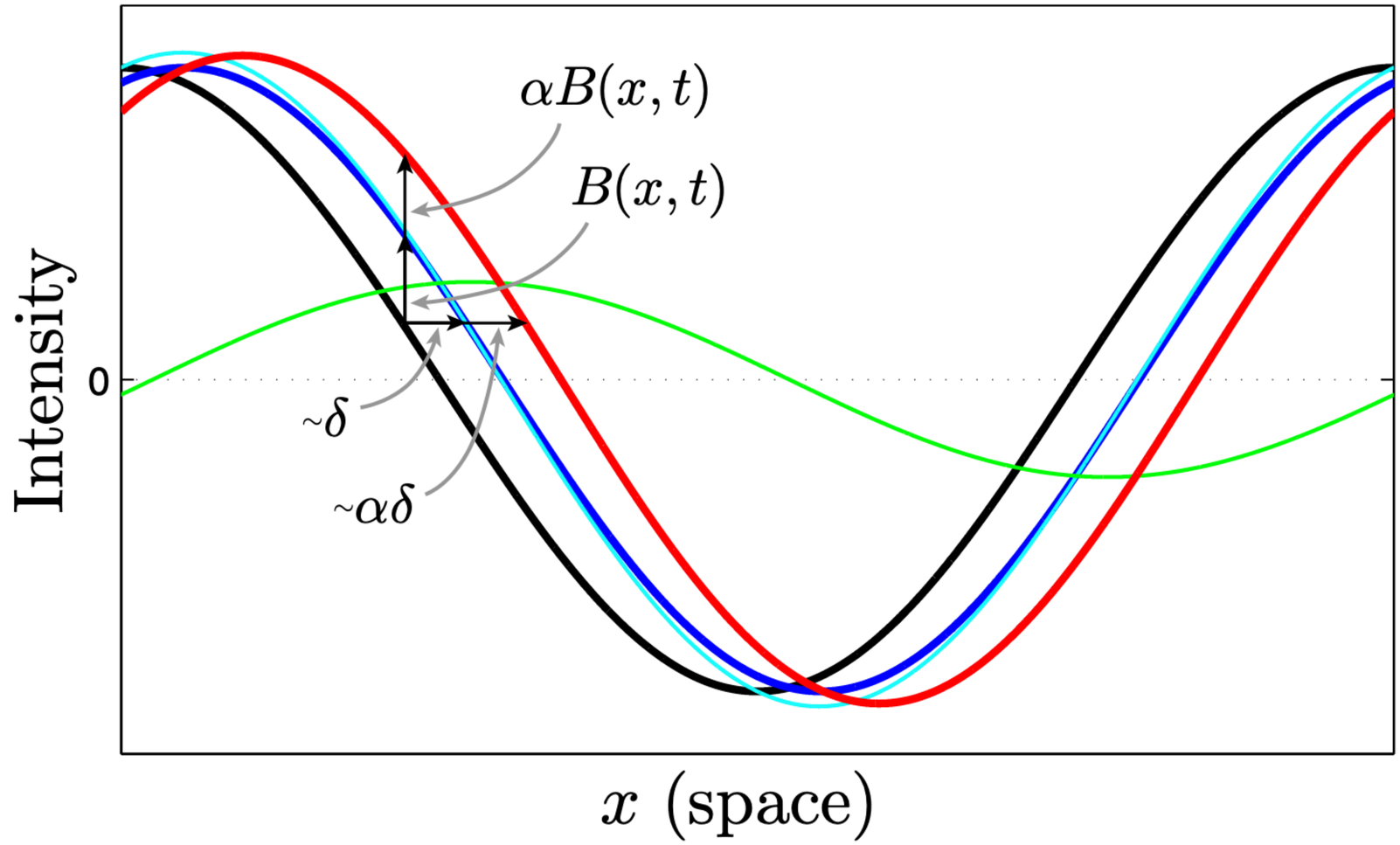


$$\boldsymbol{B}(\boldsymbol{x},\boldsymbol{t}) \approx \delta(\boldsymbol{t})\boldsymbol{I}'(\boldsymbol{x},\boldsymbol{t})$$

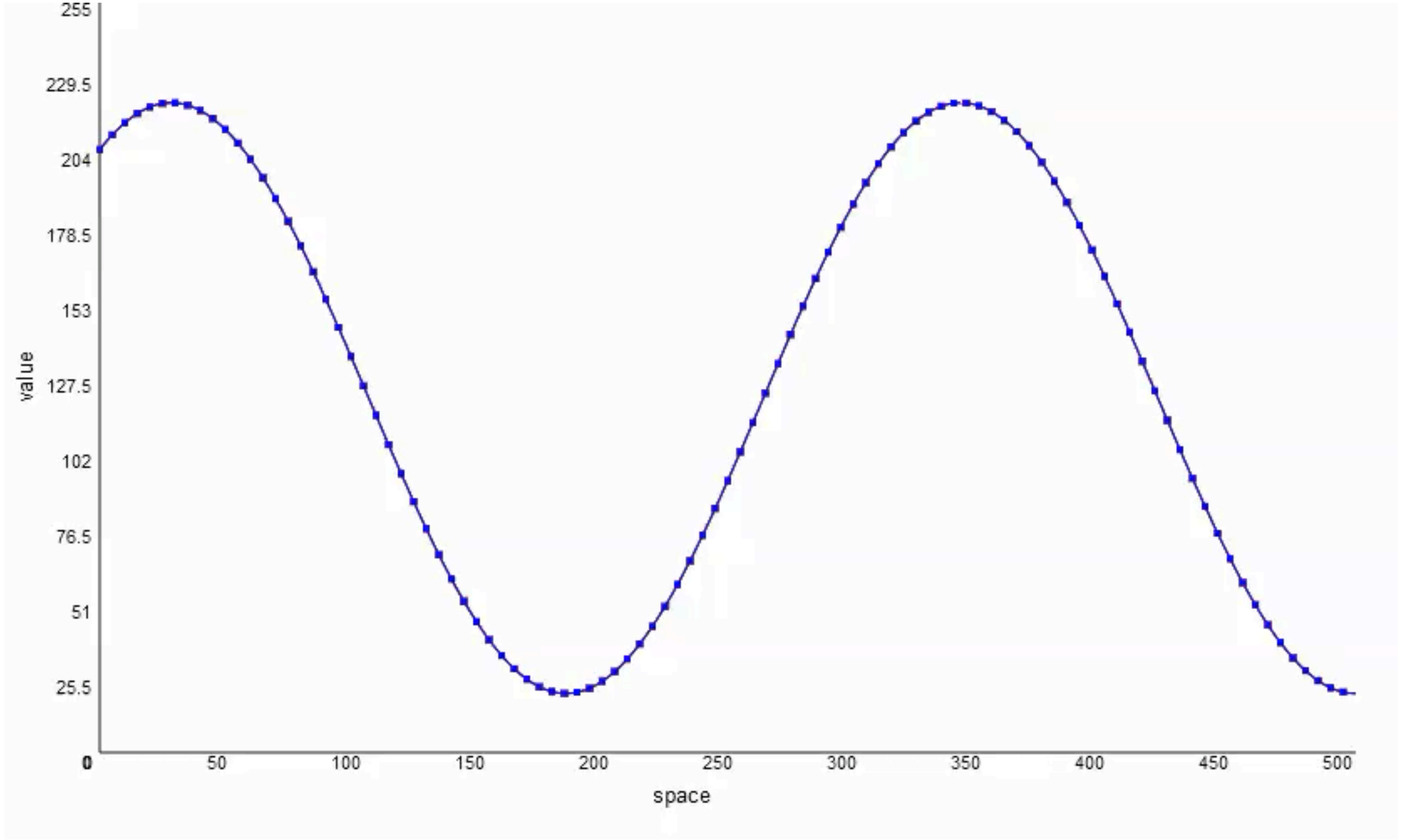
(1st order Taylor expansion)

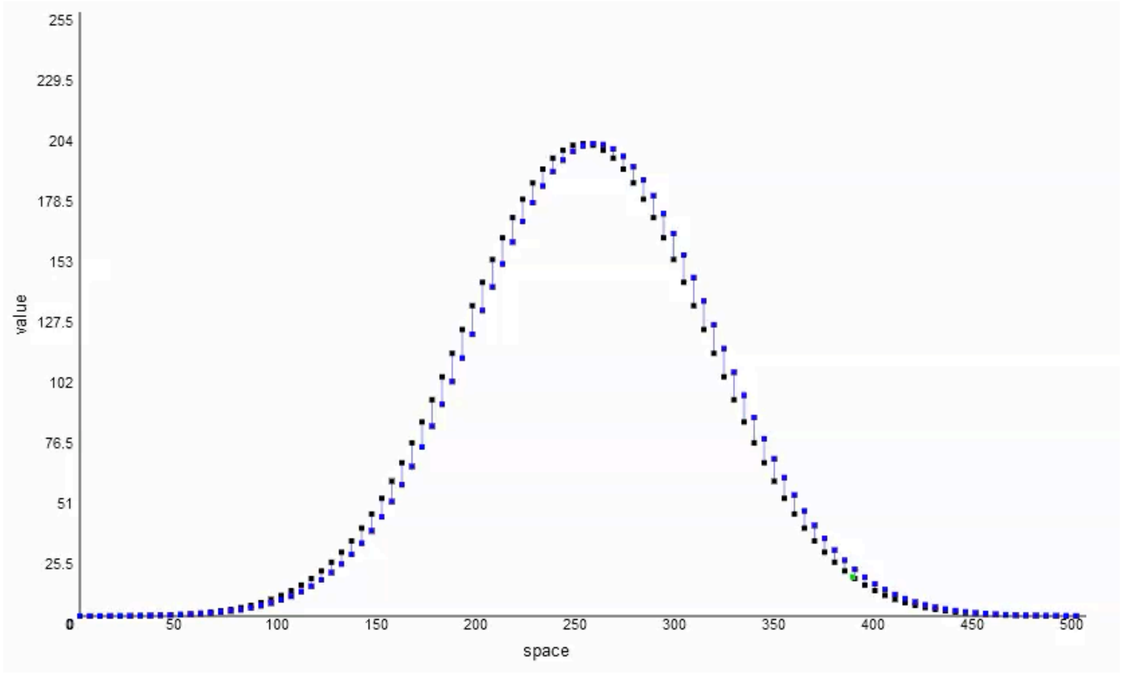


Linearity, Brightness constancy



$f(x)$ $f(x+\delta)$ $f(x)+\delta\frac{\partial f(x)}{\partial x}$ $B(x,t)$ $f(x)+(1+\alpha)B(x,t)$


255
229.5
204
178.5
153
127.5
102
76.5
51
25.5
0
value
50
100
150
200
250
300
350
400
450
500
space


255
229.5
204
178.5
153
127.5
102
76.5
51
25.5
0
value
50
100
150
200
250
300
350
400
450
500
space

In practice: **Laplacian pyramid** motion magnification

- Motivations: handle noise, select which scales can be amplified

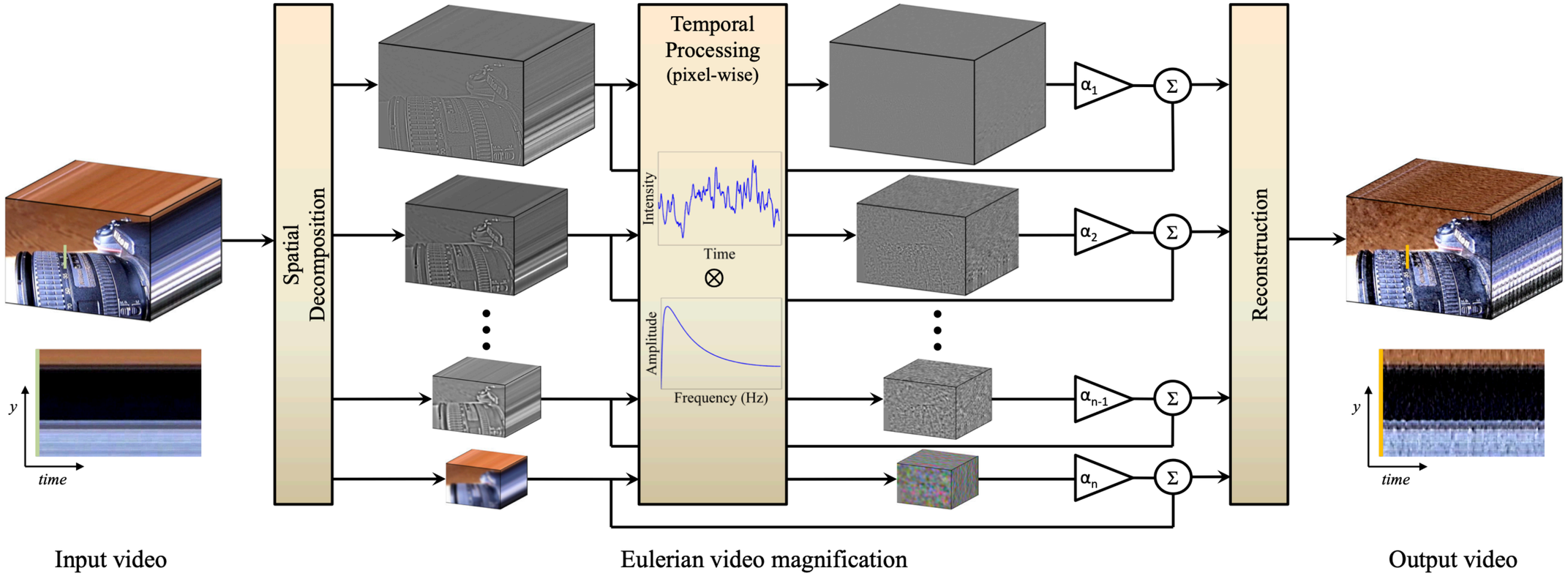


Figure 2: *Overview of the Eulerian video magnification framework. The system first decomposes the input video sequence into different spatial frequency bands, and applies the same temporal filter to all bands. The filtered spatial bands are then amplified by a given factor α, added back to the original signal, and collapsed to generate the output video. The choice of temporal filter and amplification factors can be tuned to support different applications. For example, we use the system to reveal unseen motions of a Digital SLR camera, caused by the flipping mirror during a photo burst (*camera*; full sequences are available in the supplemental video).*

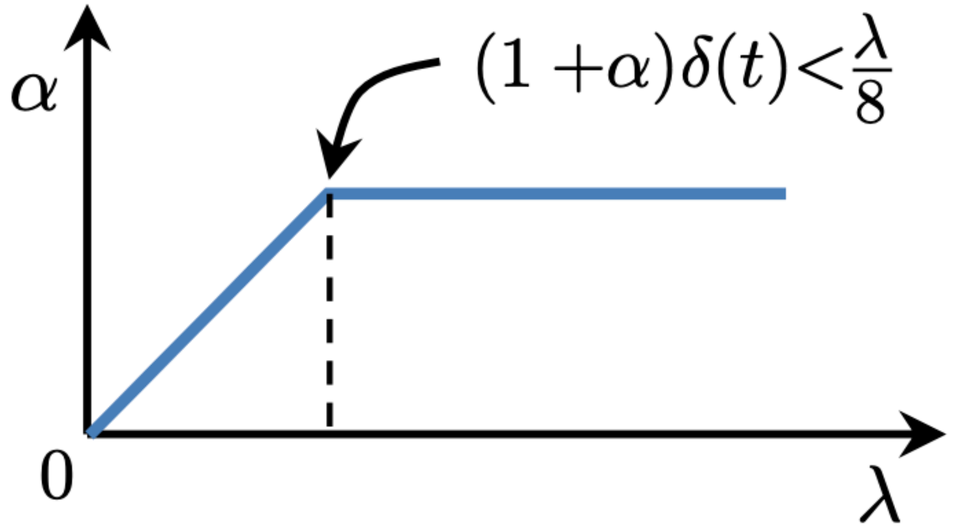


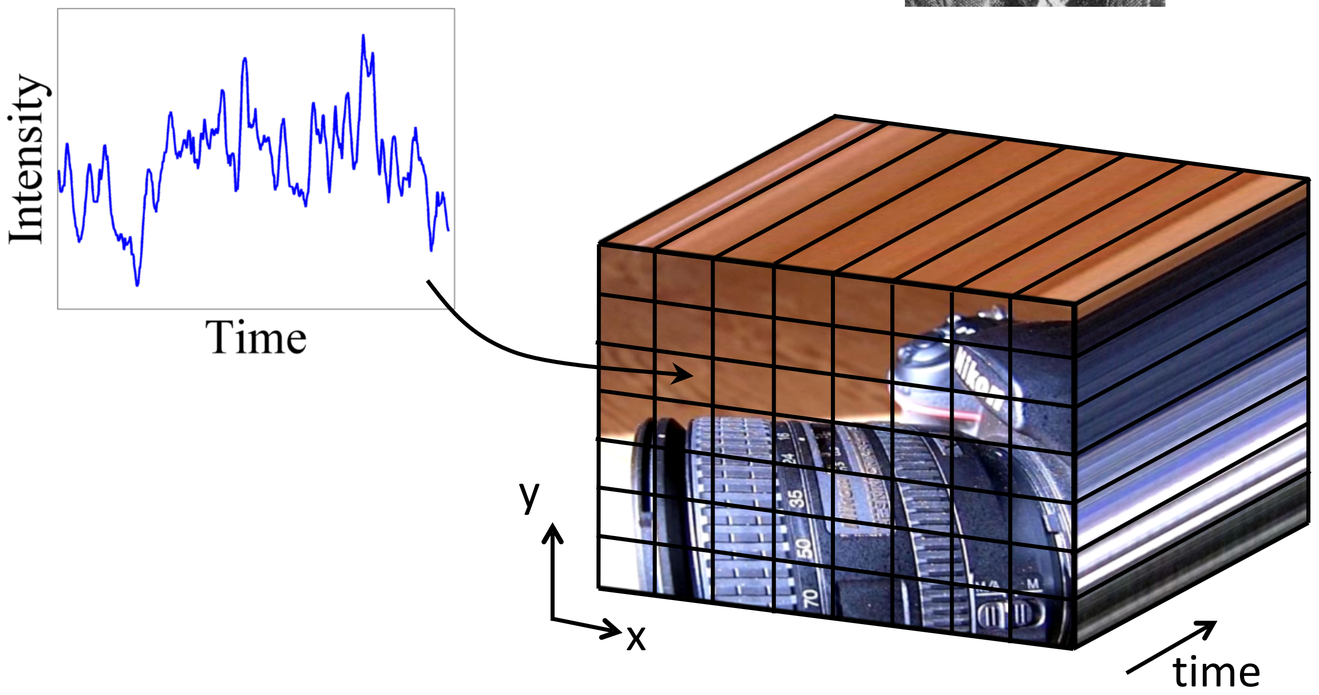
Figure 6: *Amplification factor, α, as function of spatial wavelength λ, for amplifying motion. The amplification factor is fixed to α for spatial bands that are within our derived bound (Eq. 14), and is attenuated linearly for higher spatial frequencies.*

Eulerian Perspective: Videos

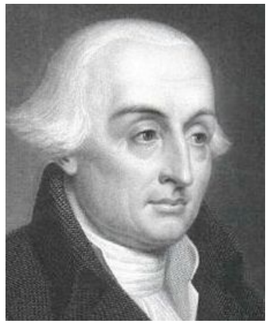
- Each pixel is processed independently
- We treat each pixel as a time series and apply signal processing to it

Eulerian





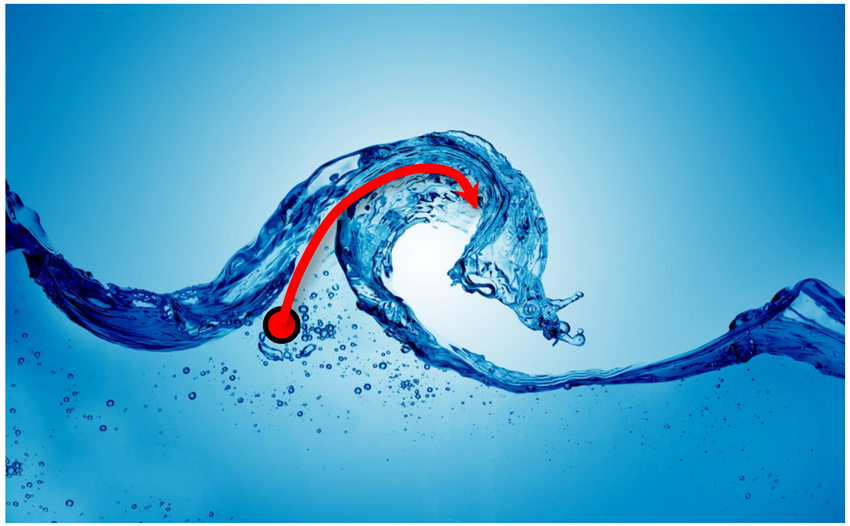
Lagrangian and Eulerian Perspectives: Fluid Dynamics

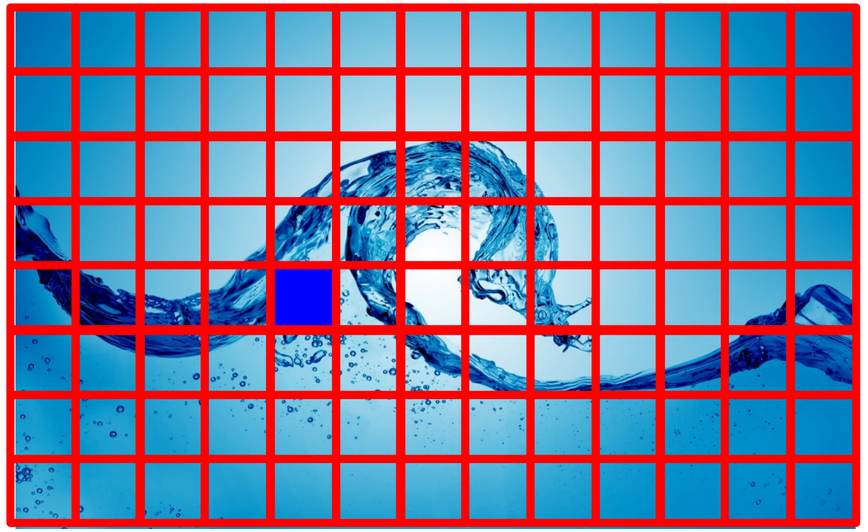


Lagrangian

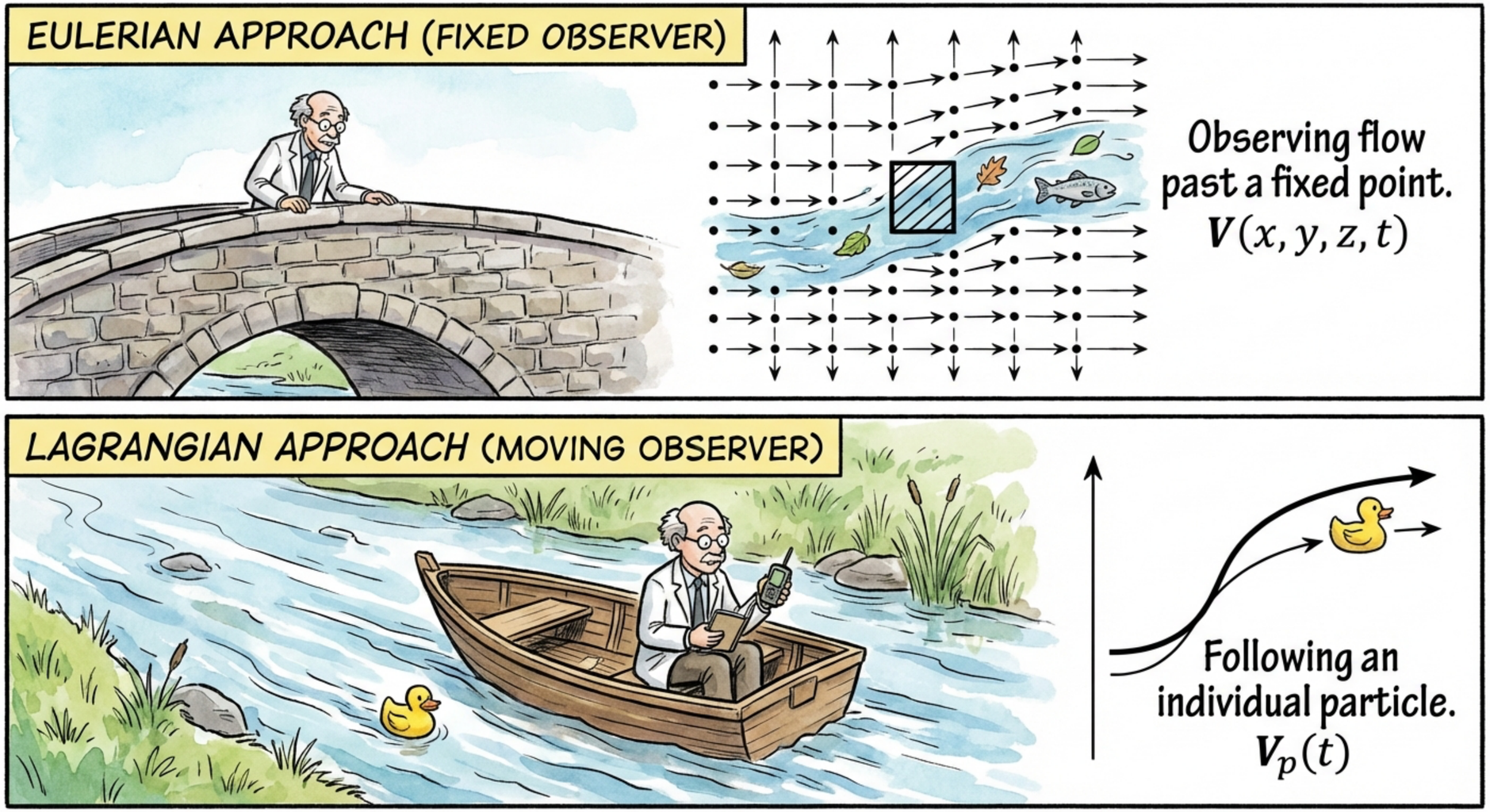
Eulerian







Lagrangian vs. Eulerian



Lagrangian motion magnification

Motion Magnification

Ce Liu Antonio Torralba William T. Freeman Frédo Durand Edward H. Adelson

Computer Science and Artificial Intelligence Lab (CSAIL)
Massachusetts Institute of Technology*





The original version (2005 vs 2012)

More challenging & brittle

But can handle bigger motion and magnification

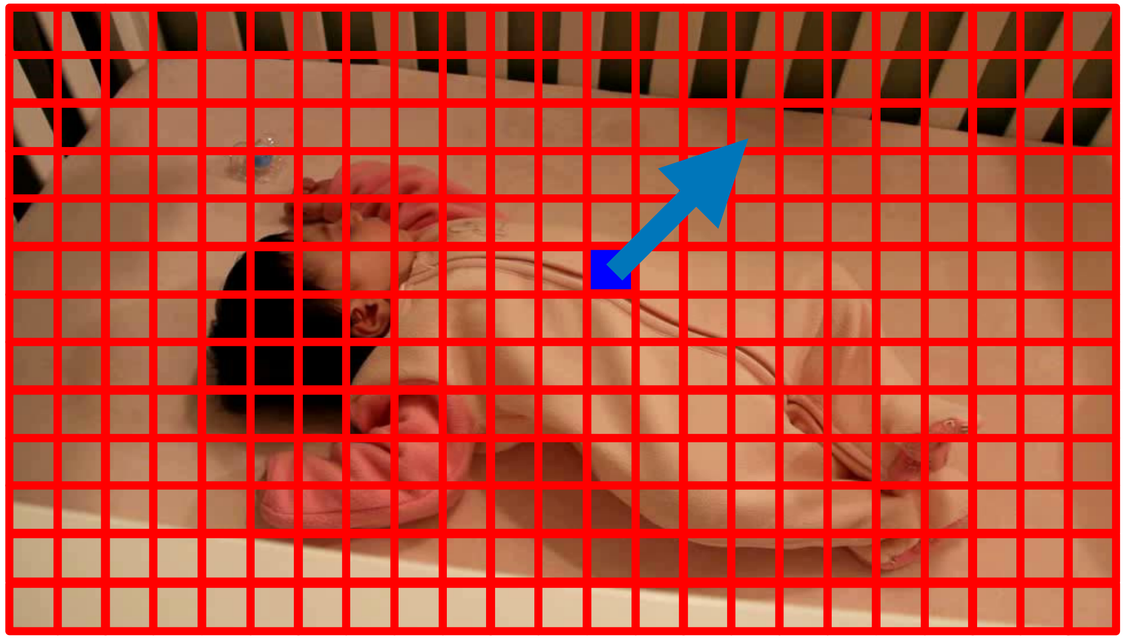
In short

- **Amazing how much Eulerian Video Magnification works given there is no explicit notion of shift or motion**
  - Largely because when things are small enough, Taylor/linearity saves us
- Purely local processing (well, a bit multiscale)
  - But is it “diagonal” or local in the right basis to handle movements such as shifts?

Phase-based video magnification

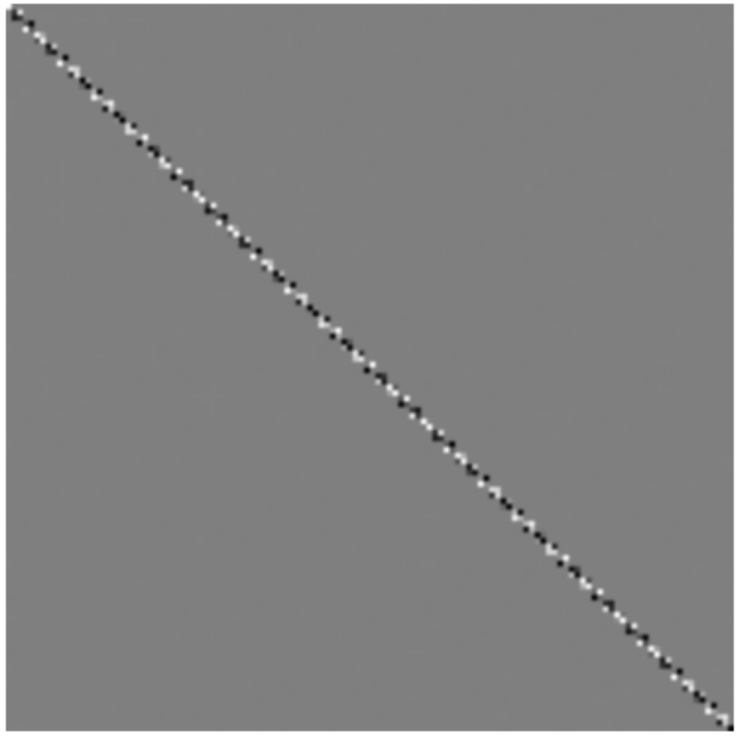
Better handling movement

- Movement is a shift
- If we want a basis/representation where shifts are purely “local”, it’s Fourier !

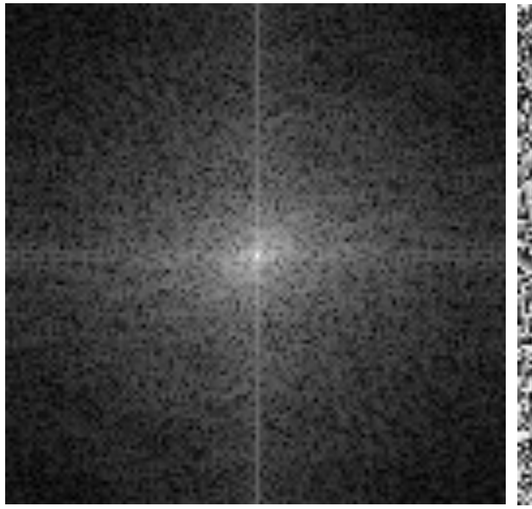


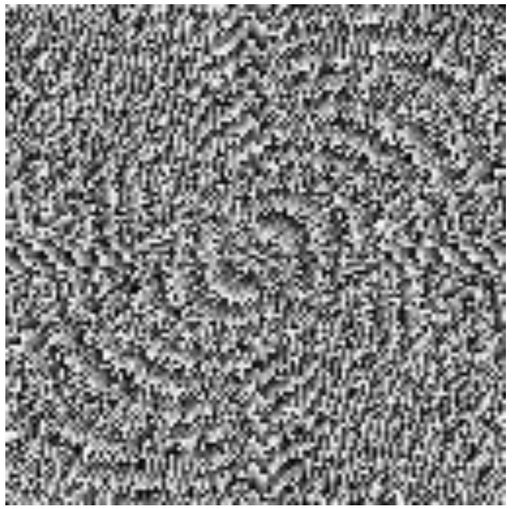
Shifting images

$$[\rho(g)f](x) = f(g^{-1}x)$$

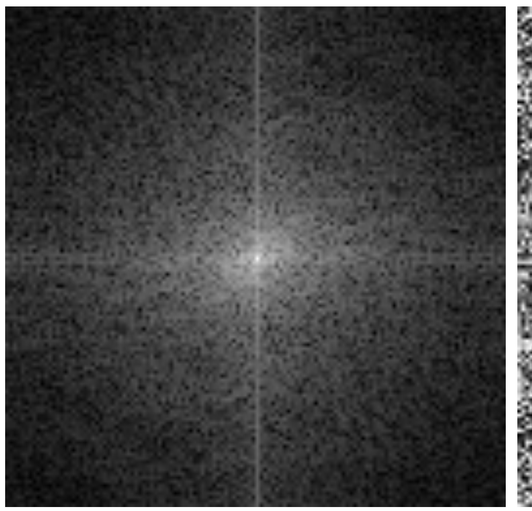


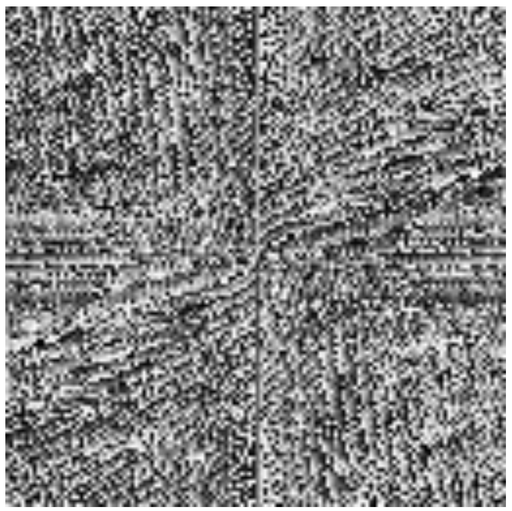










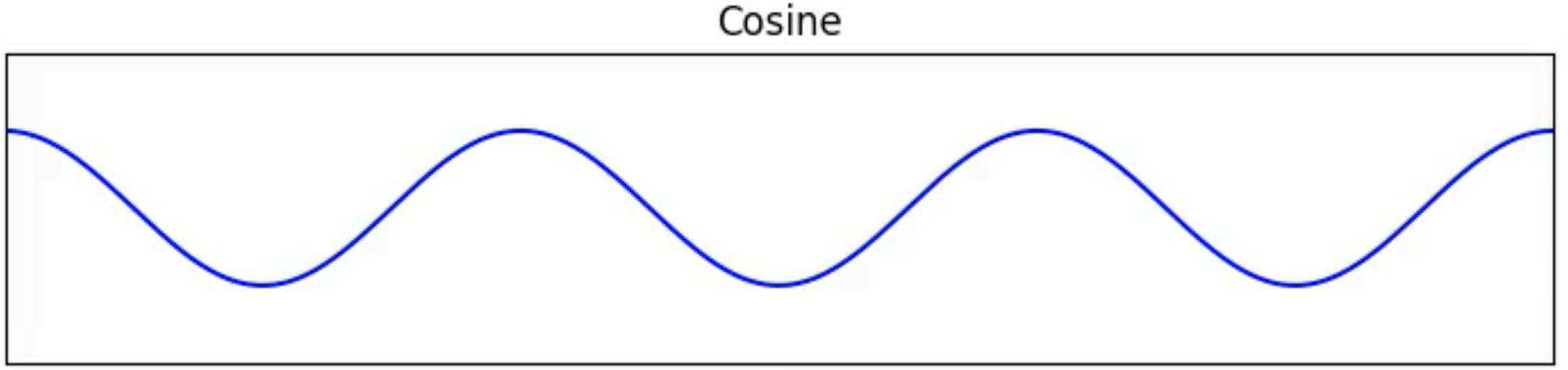


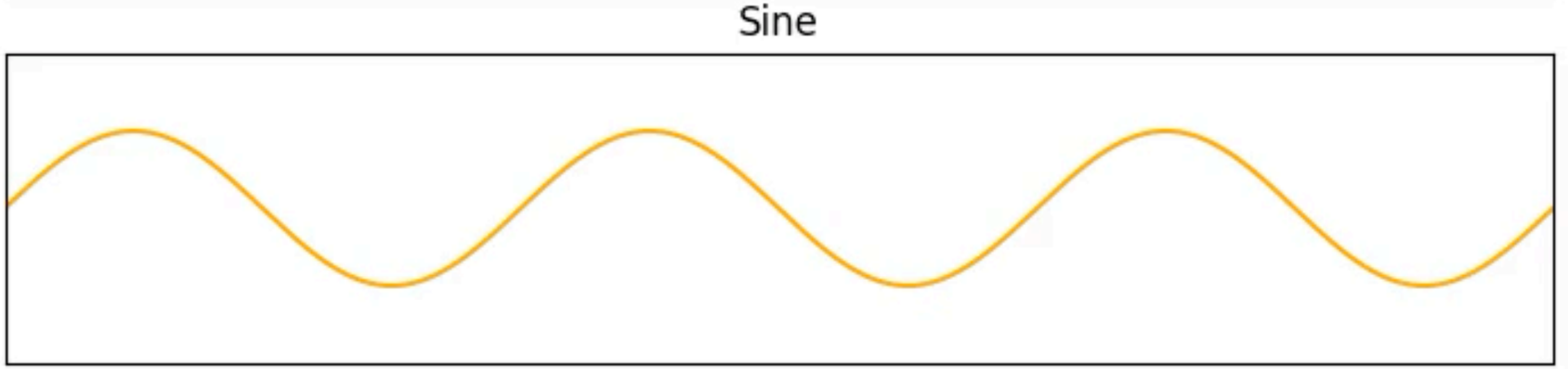
Translating a sine wave in the Fourier Basis

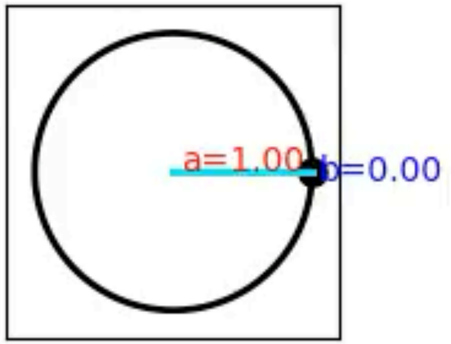
Express in
(cos, sin) basis

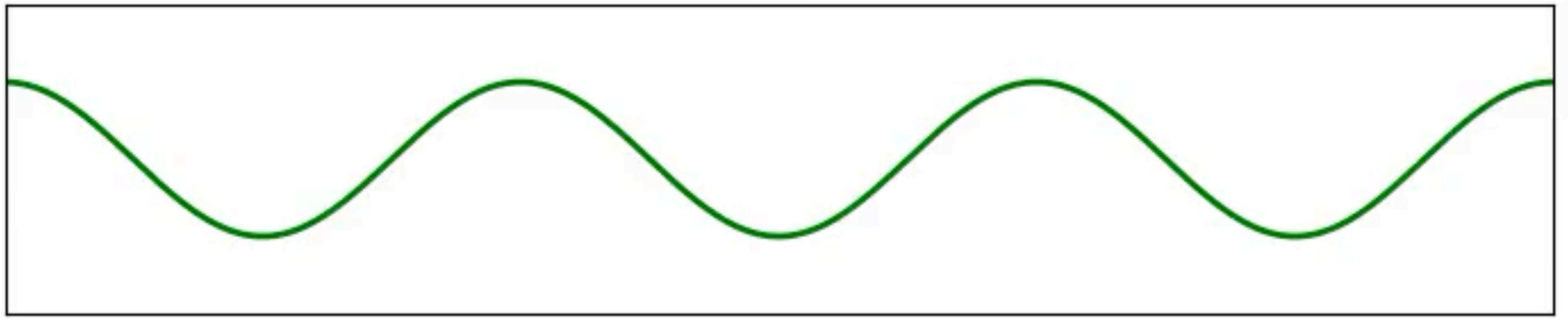
Translation group acts
via *rotation* of
coefficients!

$$< \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \begin{bmatrix} a \\ b \end{bmatrix} >$$









Steering the Fourier Basis

$Y(\alpha - \theta)$

$$\rho(-\theta) = \bigoplus_{l=-L}^{L} \rho_l(-\theta)$$

$Y(\alpha)$

$$\begin{pmatrix} \\ \\ \\ \\ \\ \\ \\ \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 & 0 & 0 & 0 & 0 \\ 0 & \sin\theta_1 & \cos\theta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & 0 & \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cos\theta_3 & -\sin\theta_3 \\ 0 & 0 & 0 & 0 & 0 & \sin\theta_3 & \cos\theta_3 \end{pmatrix} \begin{pmatrix} \\ \\ \\ \\ \\ \\ \\ \end{pmatrix}$$

Fourier video magnification simplest version

- Fourier transform of two images
- For each Fourier coefficient, consider phase difference
  - Or rotation angle of real-valued pair of coefficients
  - Amplify!
- Reconstruct with inverse Fourier transform

Fourier video magnification?

- Good news:
  compact point wise (if complex) blockwise (if real) handling of translational motion
- Problem:
  - Fourier handles only global shifts
  - Assumes cyclic images

Steerable pyramid goal

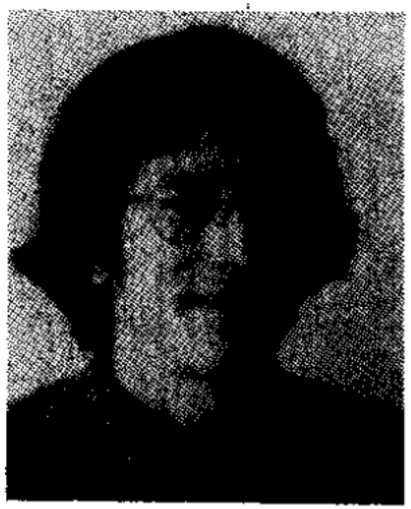
- **Sweet spot between Fourier and canonical**
  - Limited boundary issues
  - Local enough
- **Steerable in all sorts of ways**
  - Rotation invariance
  - Local shifts
  - Scale
- **At the cost of overcomplete**
  - (also careful: bunch of different flavors)

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 13, NO. 9, SEPTEMBER 1991

The Design and Use of Steerable Filters

William T. Freeman and Edward H. Adelson

1991





More recently :)





The Design and Use of Steerable Filters

William T. Freeman and Edward H. Adelson

I. Introduction

ORIENTED filters are used in many vision and image processing tasks, such as texture analysis, edge detection, image data compression, motion analysis, and image enhancement. In many of these tasks, it is useful to apply filters of arbitrary orientation under adaptive control and to examine the filter output as a function of both orientation and phase.

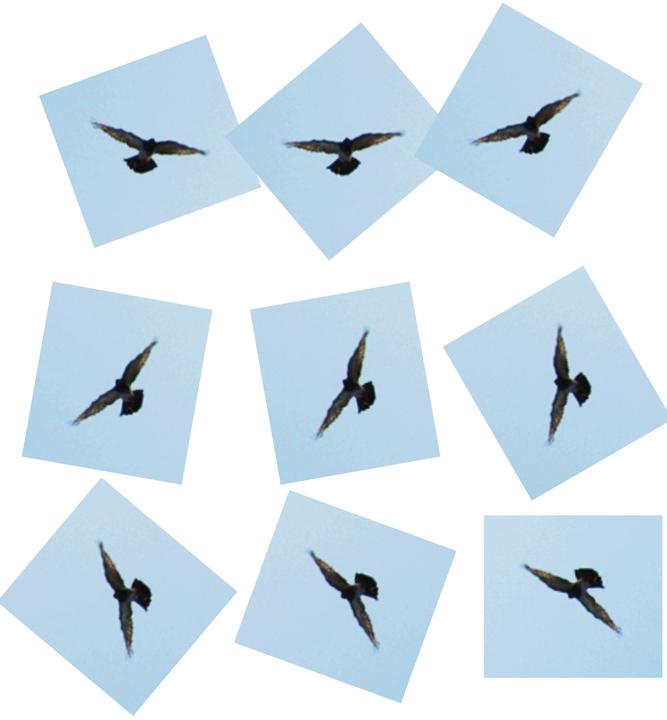
Goal: filters where the response for the filter rotated by an arbitrary angle can be obtained with a simple combination of a finite set of filters

e.g. Detection across shift & scale & orientation



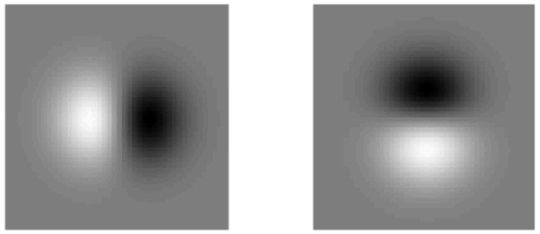


Template



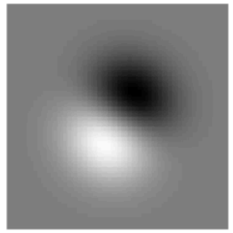
Steerability in orientation

2 basis filters



a b

Filter at arbitrary orientation



Input



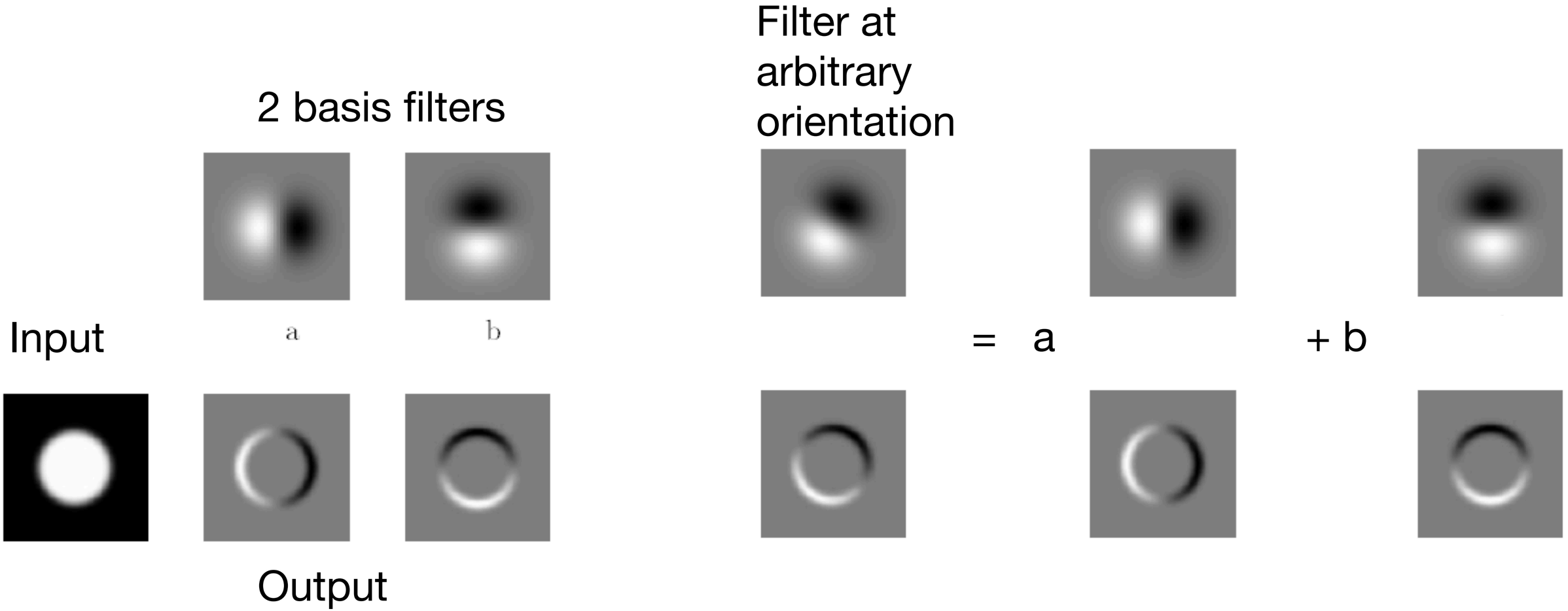




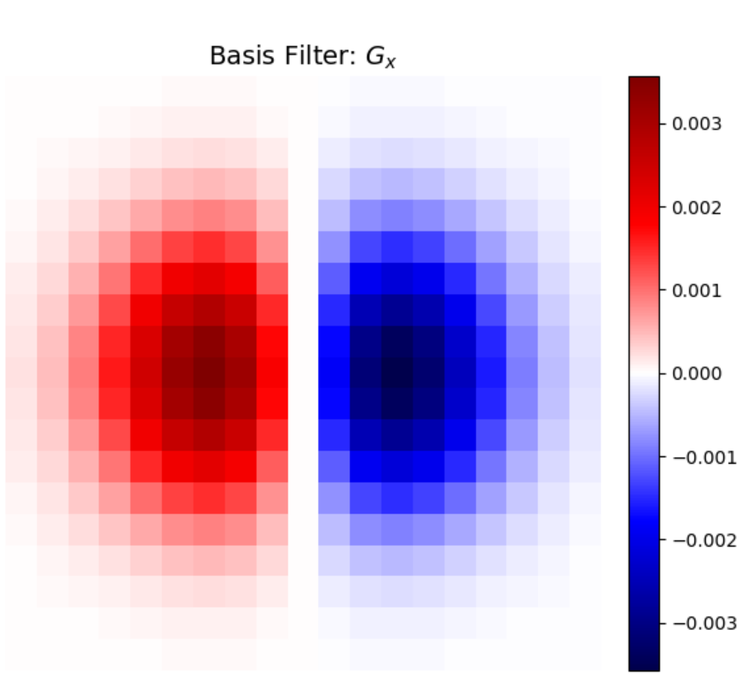


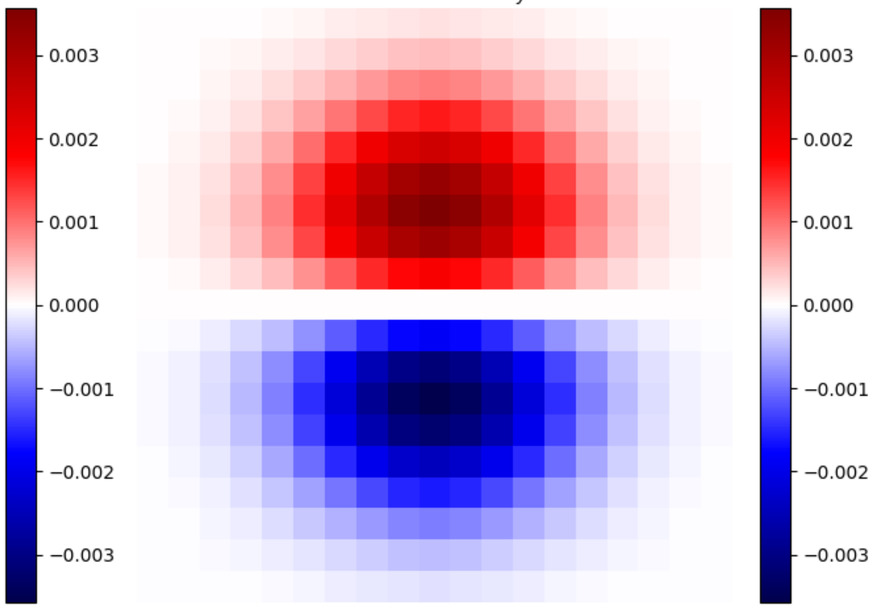
Output

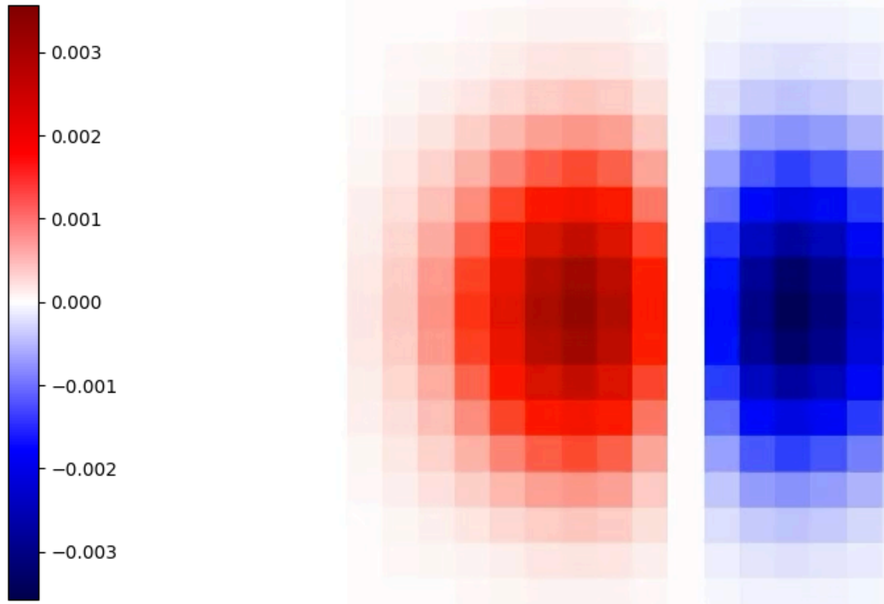
Steerability in orientation



Steerable Filters

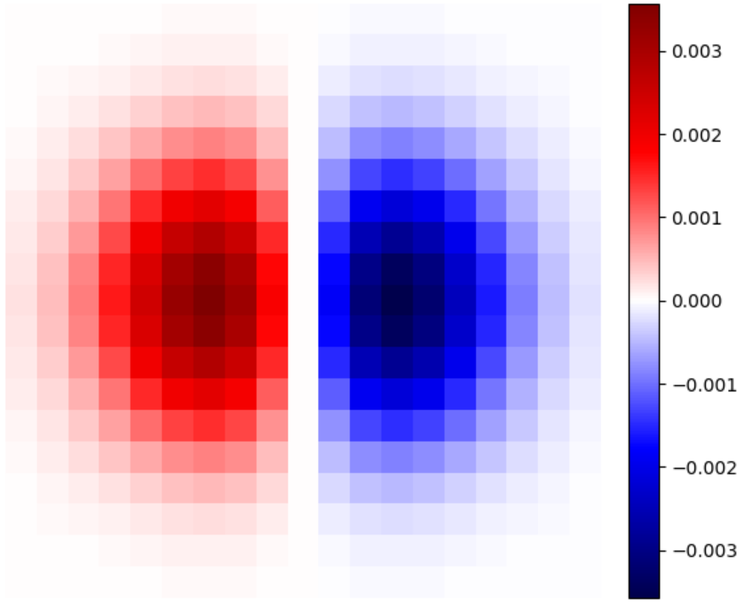




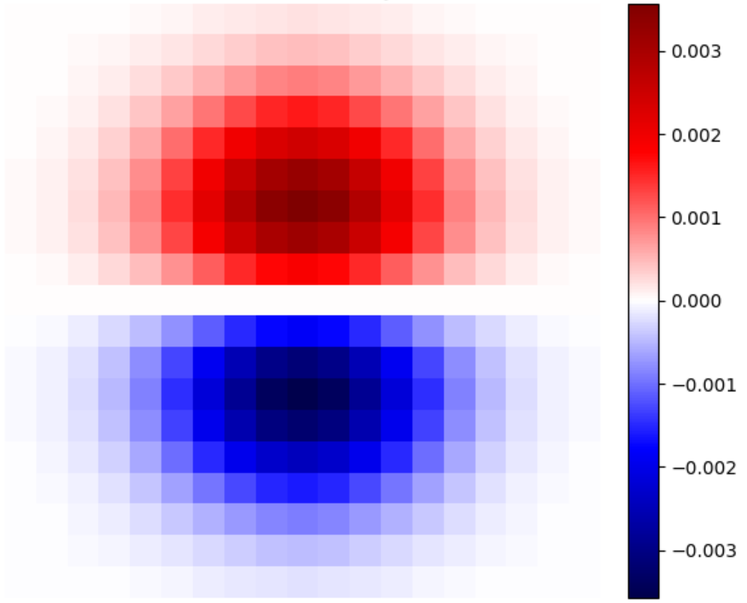


Steerable Filters

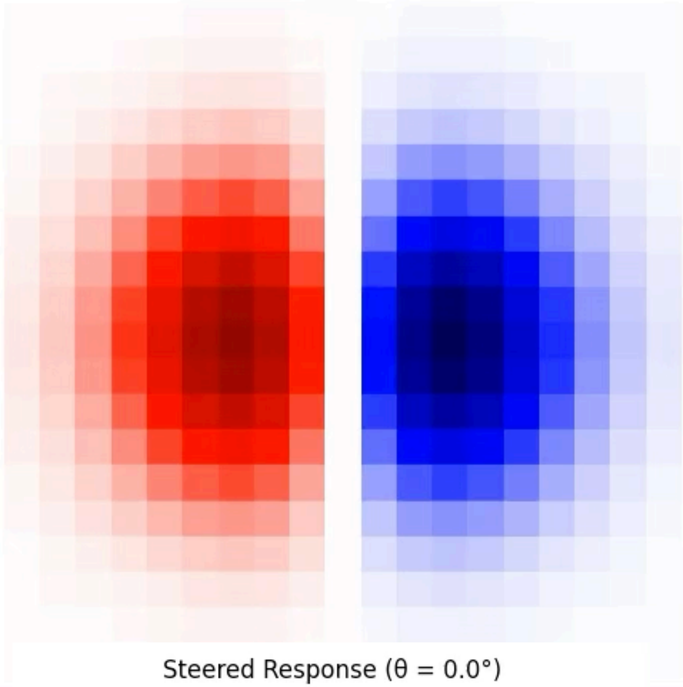
Basis Filter: G_x



Basis Filter: G_y



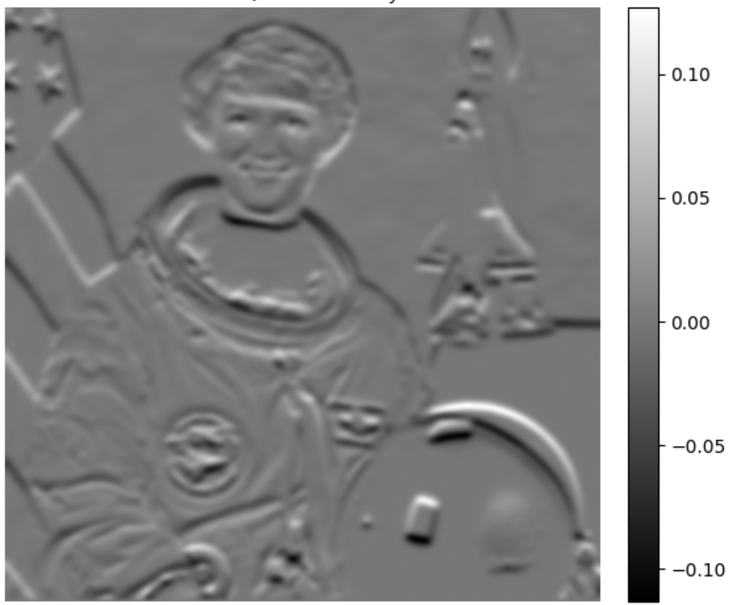
Steered Filter (θ = 0.0°)



Response to G_x



Response to G_y

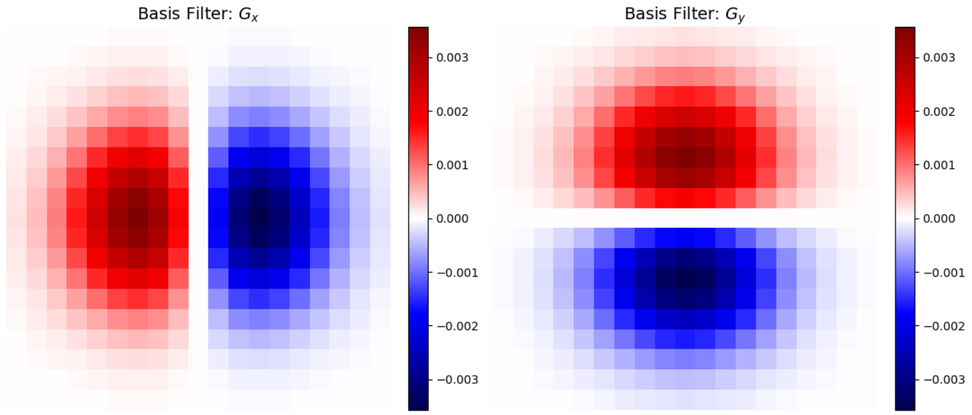


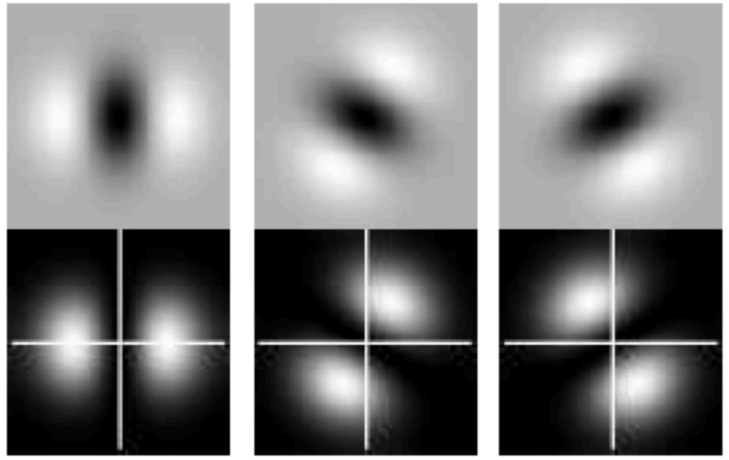
Steered Response (θ = 0.0°)



Steerable filters

- Related to spatial sampling/reconstruction/Shannon/Nyquist
  - Discrete sampling of direction
  - What can we home reconstruct given discrete samples?
  - Need to band limit in angle
- The higher the number of orientation samples, the “sharper” (more selective) in orientations





Steerable pyramids

THE STEERABLE PYRAMID: A FLEXIBLE ARCHITECTURE FOR MULTI-SCALE DERIVATIVE COMPUTATION

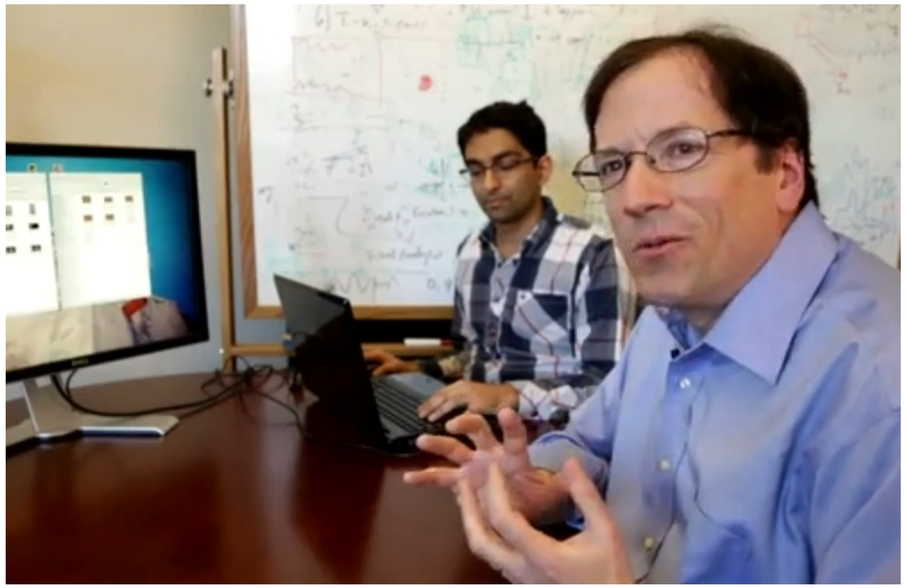
Eero P Simoncelli

GRASP Laboratory, Room 335C
3401 Walnut St
Philadelphia, PA 19104-6228

William T Freeman

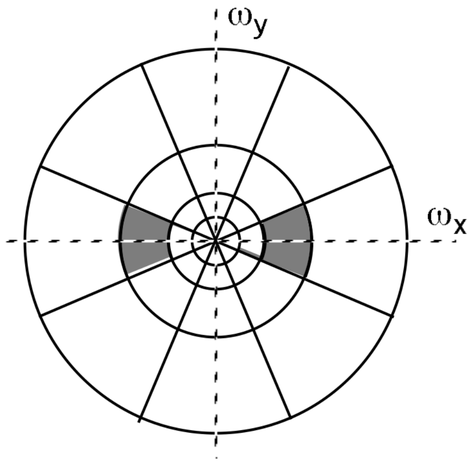
Mitsubishi Electric Research Laboratories
201 Broadway
Cambridge, MA 02139





Steerable pyramids

- Similar to Laplacian: coarse subdivision of frequency domain
- The rest is encoded spatially



Actual boundaries are “soft” (Important for steerability)

Figure 1. Idealized illustration of the spectral decomposition performed by a steerable pyramid with $k = 4$. Frequency axes range from $-\pi$ to π. The basis functions are related by translations, dilations and *rotations* (except for the initial highpass subband and the final lowpass subband). For example, the shaded region corresponds to the spectral support of a single (vertically-oriented) subband.

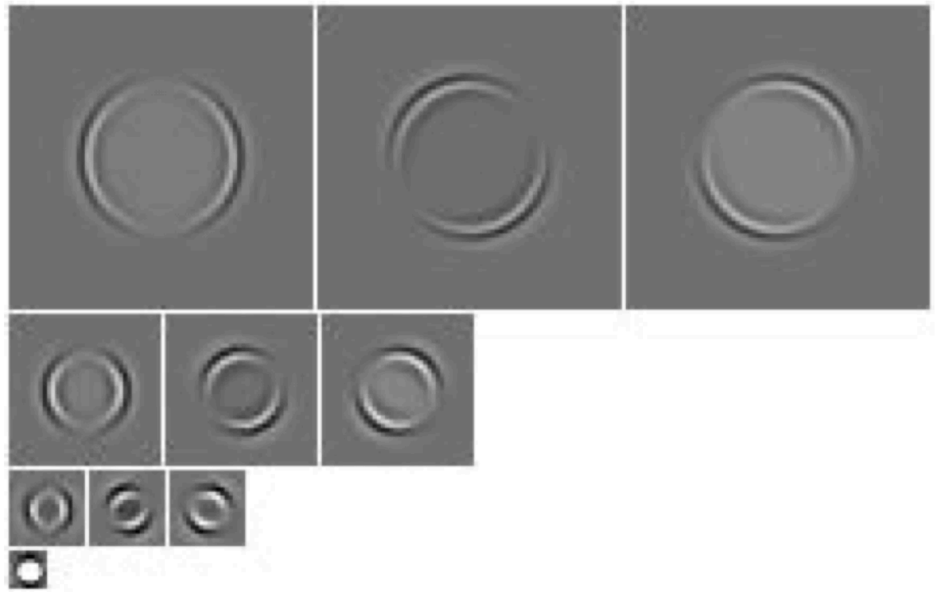


Figure 5. A 3-level $k = 3$ (second derivative) steerable pyramid. Shown are the three bandpass images at each scale and the final lowpass image.

- Can be constructed in spatial of frequency domain

Soft mask

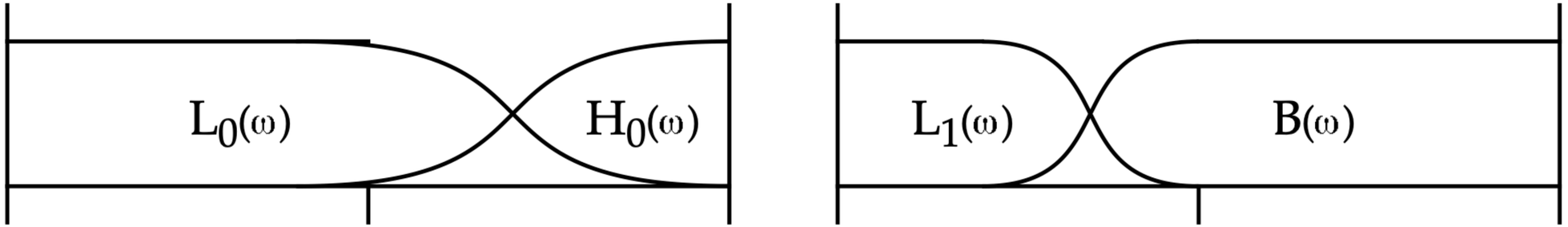


Figure 3. Idealized depiction of filters satisfying the constraints of the block diagram in figure 2. Plots show Fourier spectra over the range $[0, \pi]$.

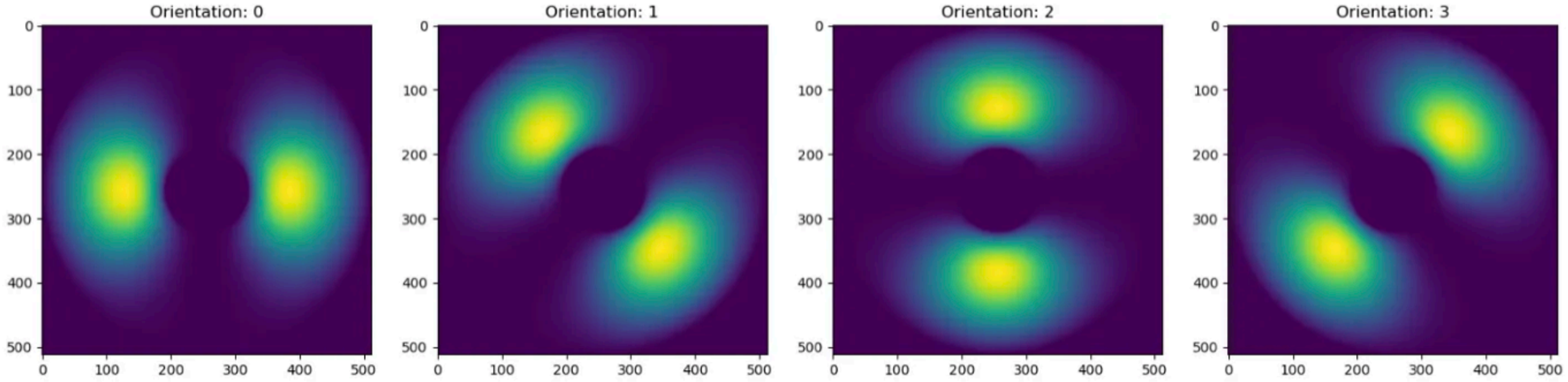
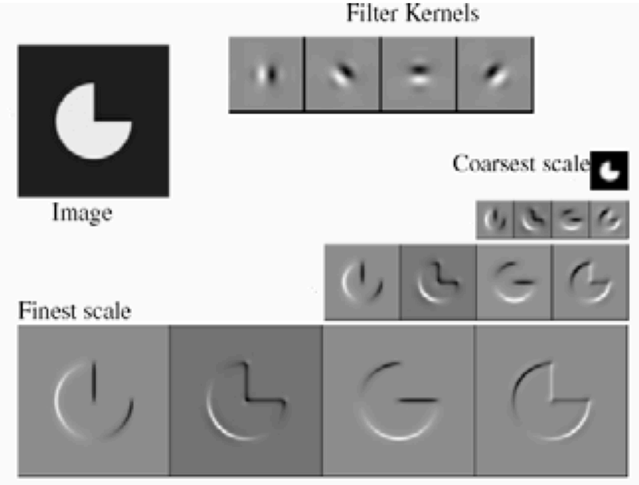


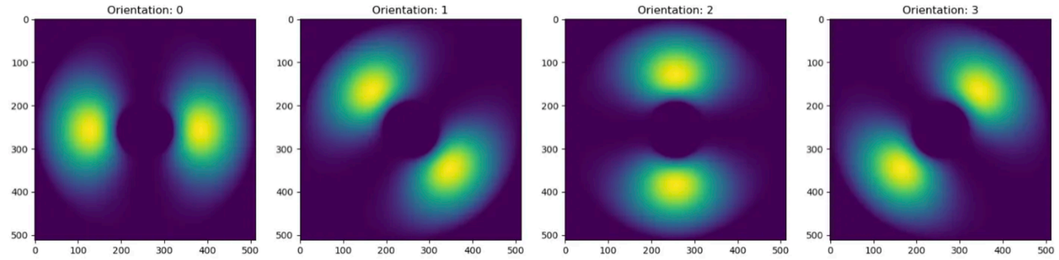
Figure 10. First level of Steerable Pyramid Filters with 4 orientations. Source: Author.

Different constructions

- Spatial domain construction
  - Bank of multi scale oriented filters



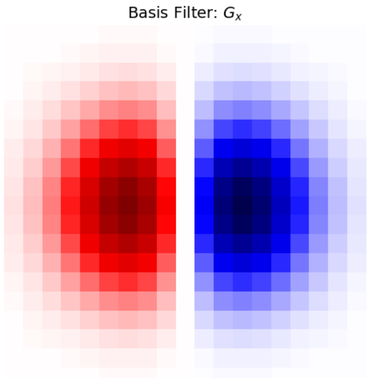
- Frequency domain construction
  - Fourier transform, Soft Masks, inverse Fourier



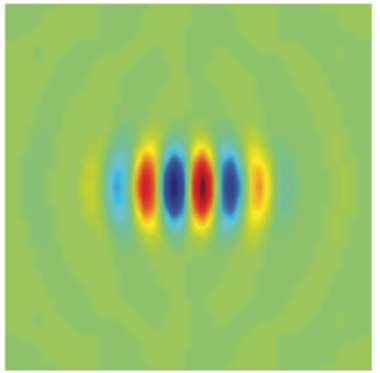
- In all cases, balance between different objectives (compactness in space, frequency, selectivity, reconstruction quality, etc.)
  - Try to get close enough to idealized math, but practical

Complex : Phase and magnitude

- Original steerable pyramid is real-valued and looks more like a local band-limited derivative



- But we can define a complex-valued one that is more like a sine wave locally
  - Local phase, local magnitude



A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients

JAVIER PORTILLA AND EERO P. SIMONCELLI
Center for Neural Science, and Courant Institute of Mathematical Sciences, New York University, New York, NY 10003, USA

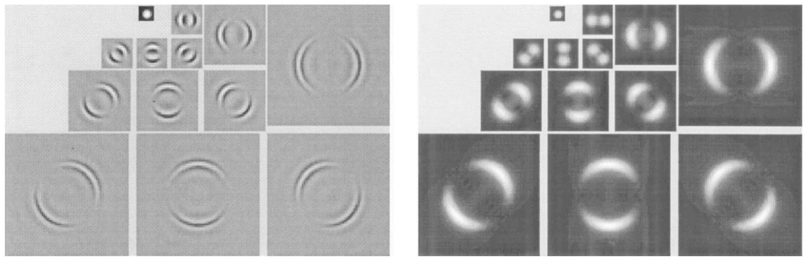
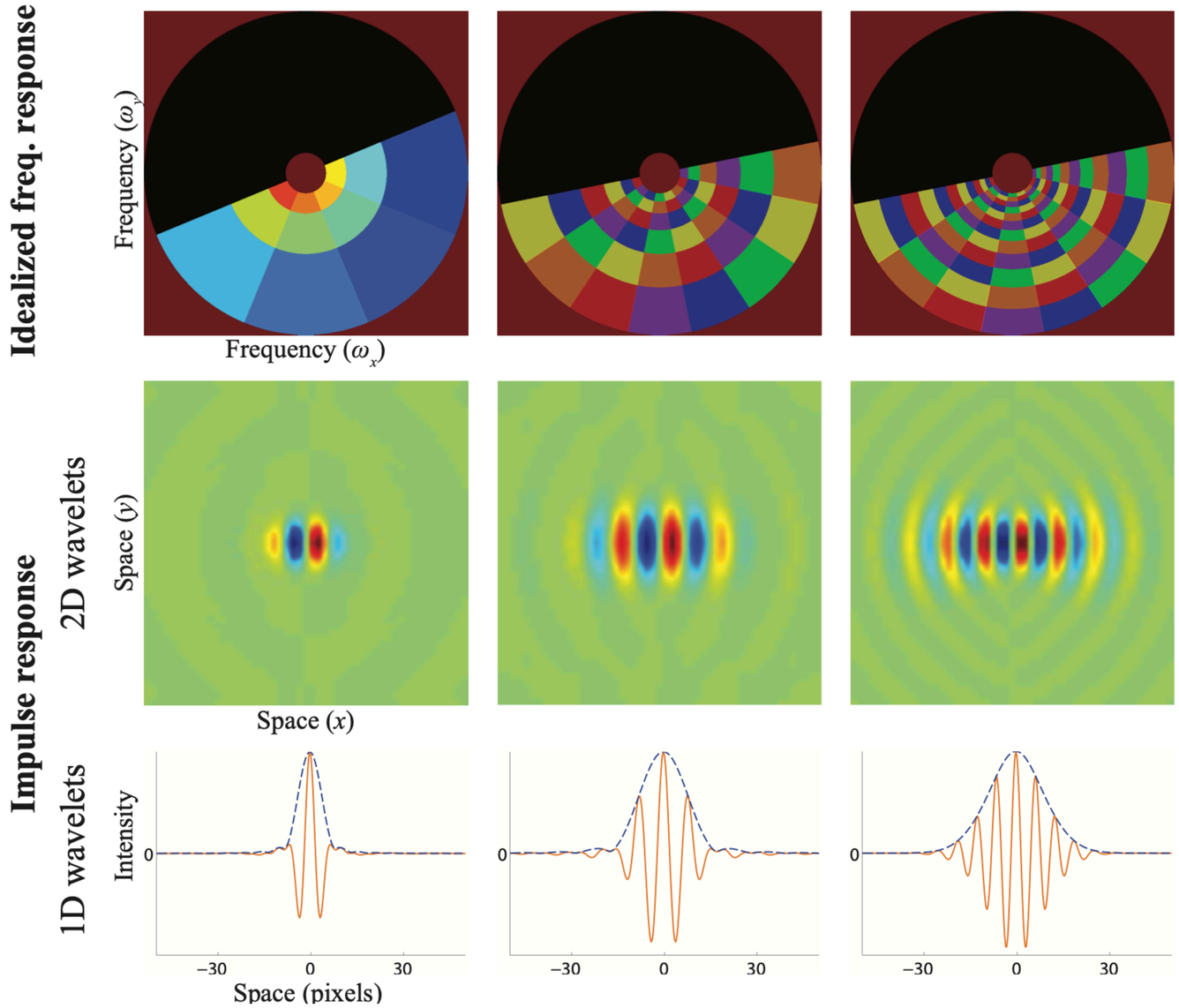
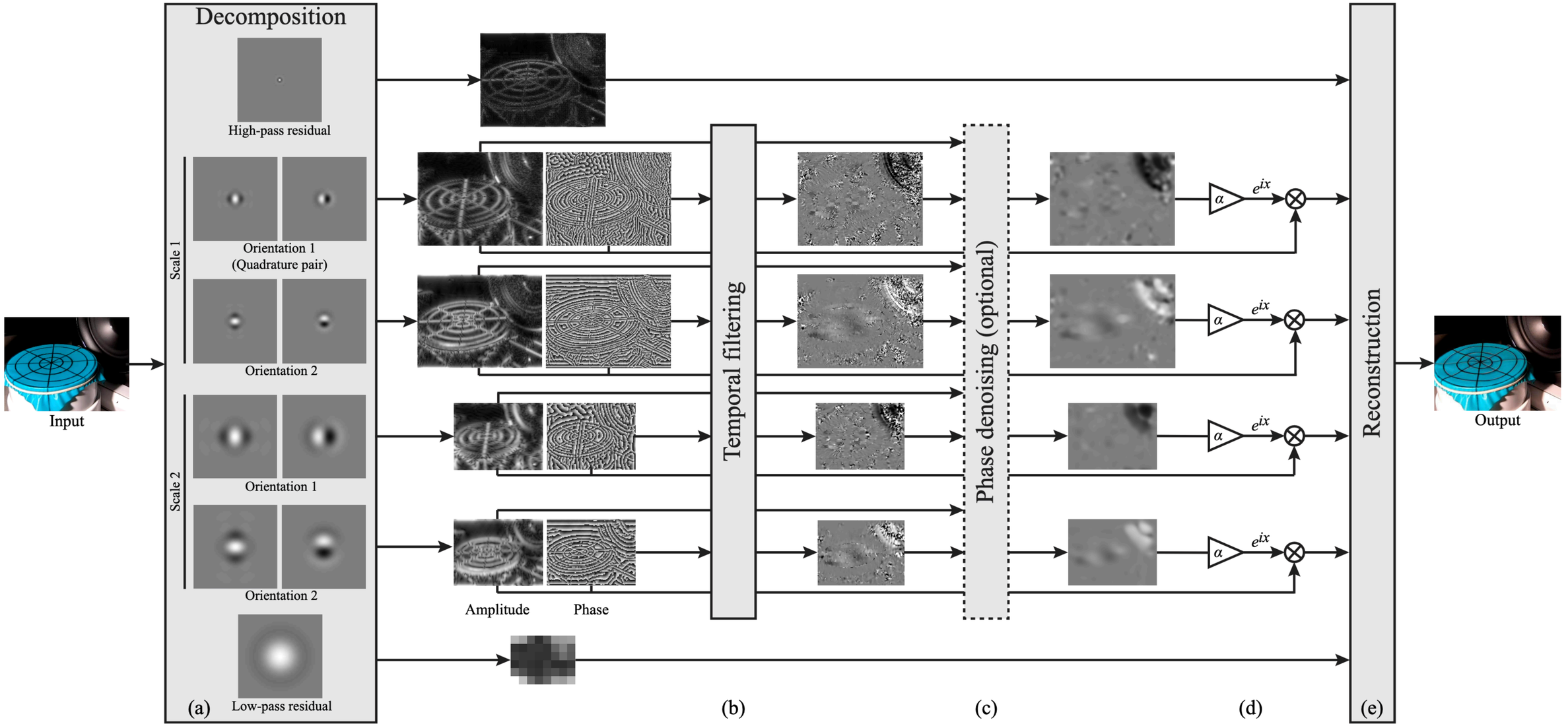


Figure 2. A 3-scale, 4-orientation complex steerable pyramid representation of a disk image. Left: real parts of oriented bandpass images at each scale and the final lowpass image. Right: magnitude (complex modulus) of the same subbands. Note that the highpass residual band is not shown.

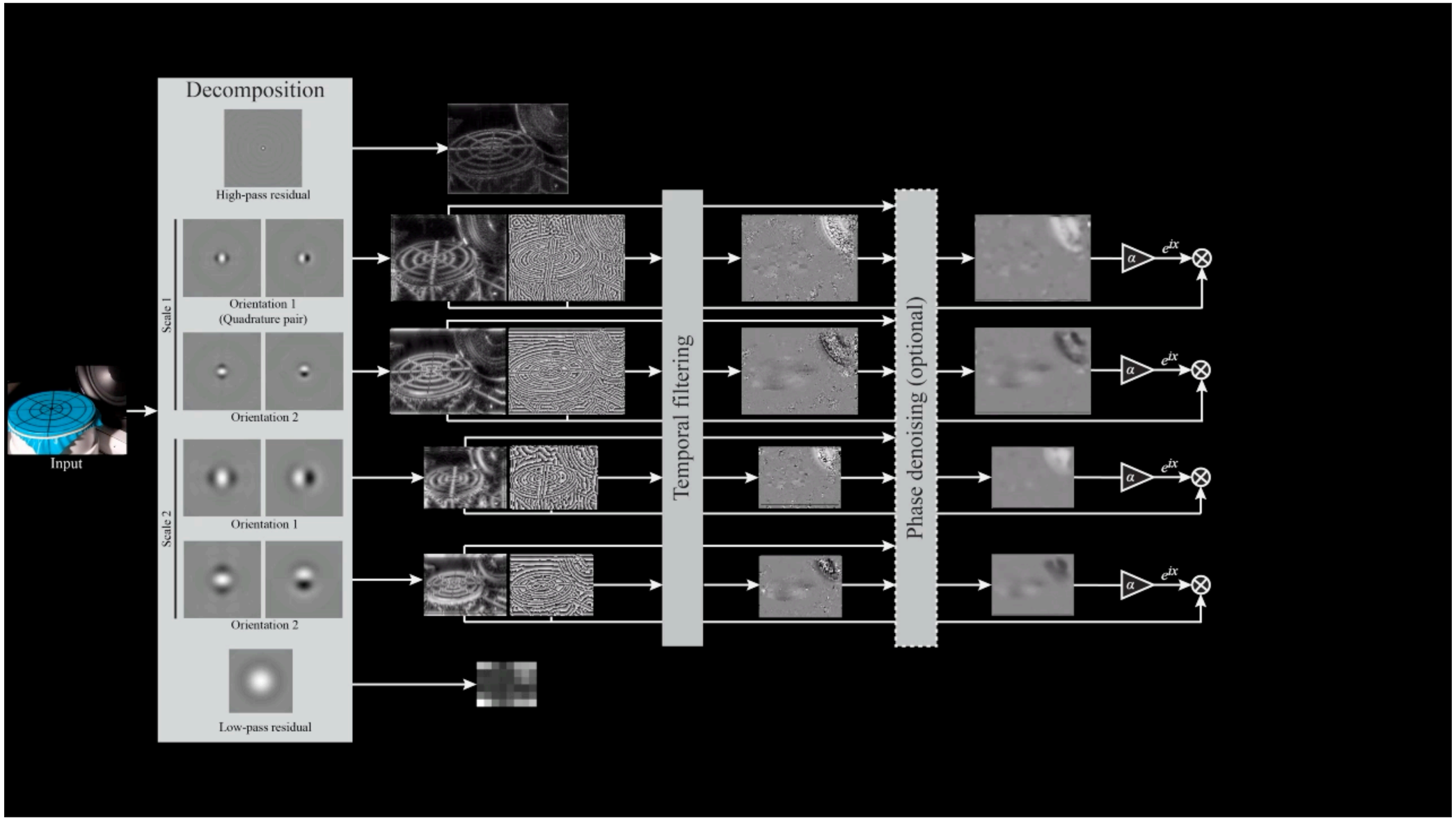
Different flavors (number of bands in scale and orientation)



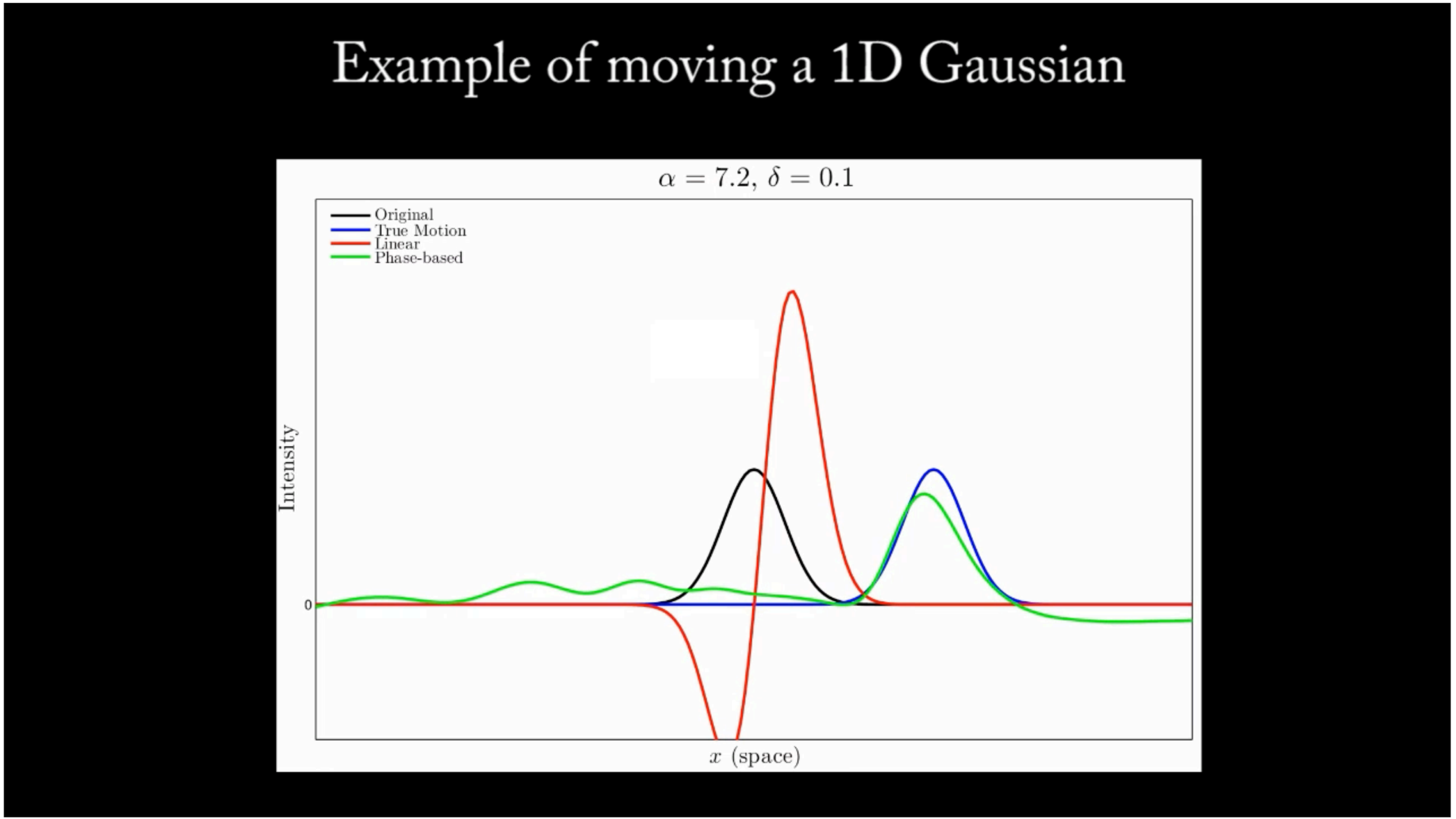
Application (and your homework): motion magnification



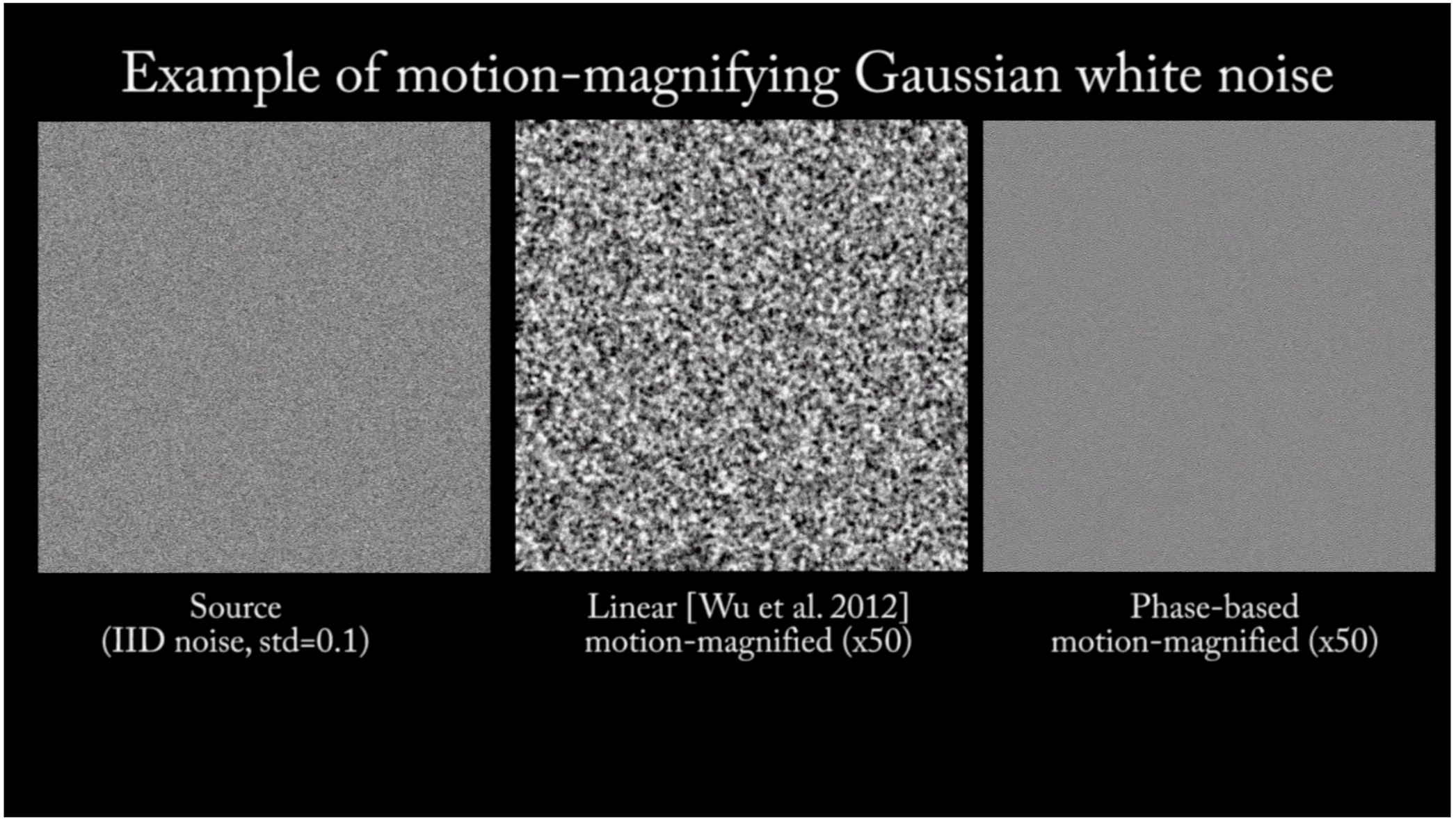
Application: Phase-based video processing



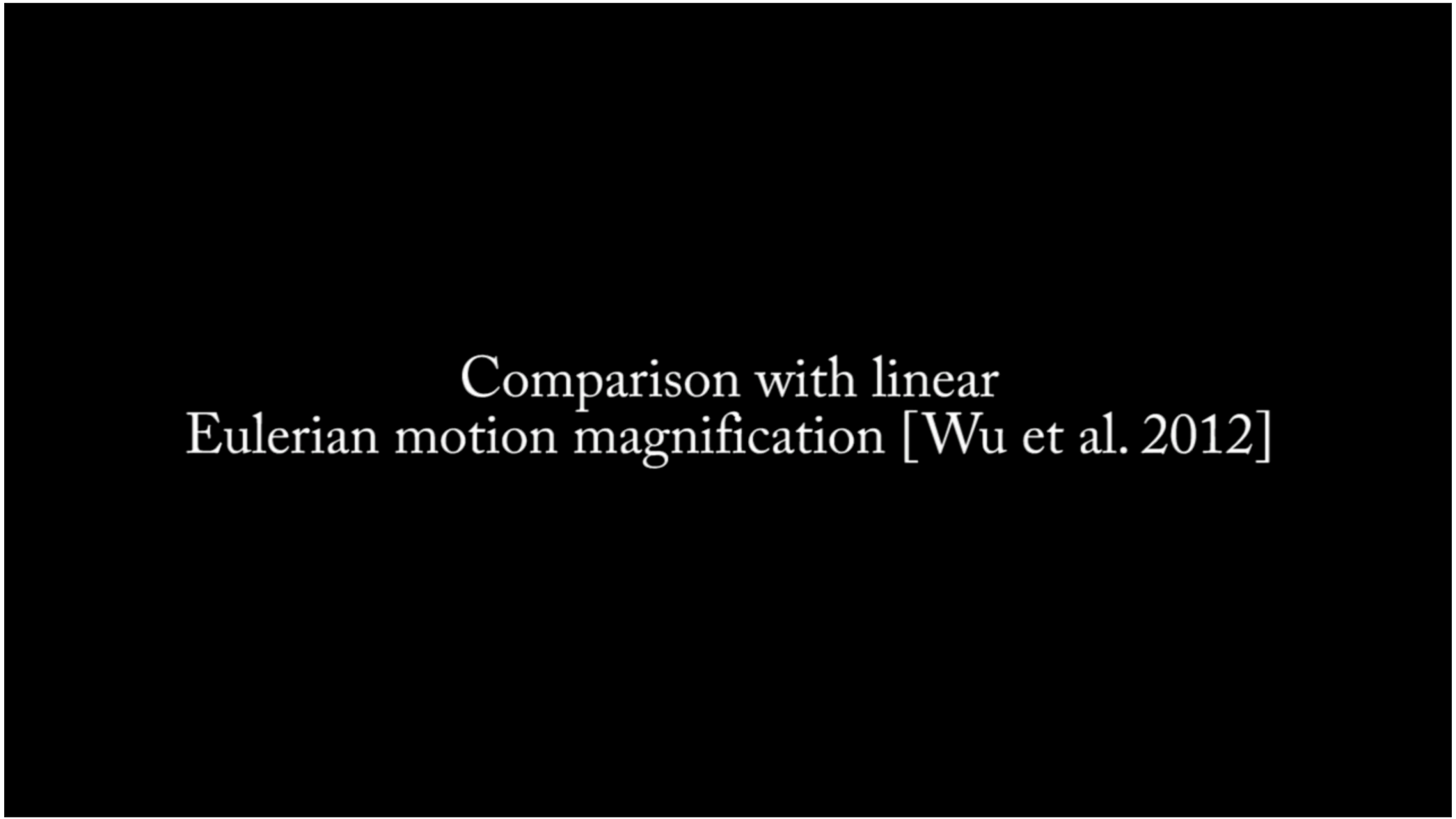
Application: Phase-based video processing



Application: Phase-based video processing

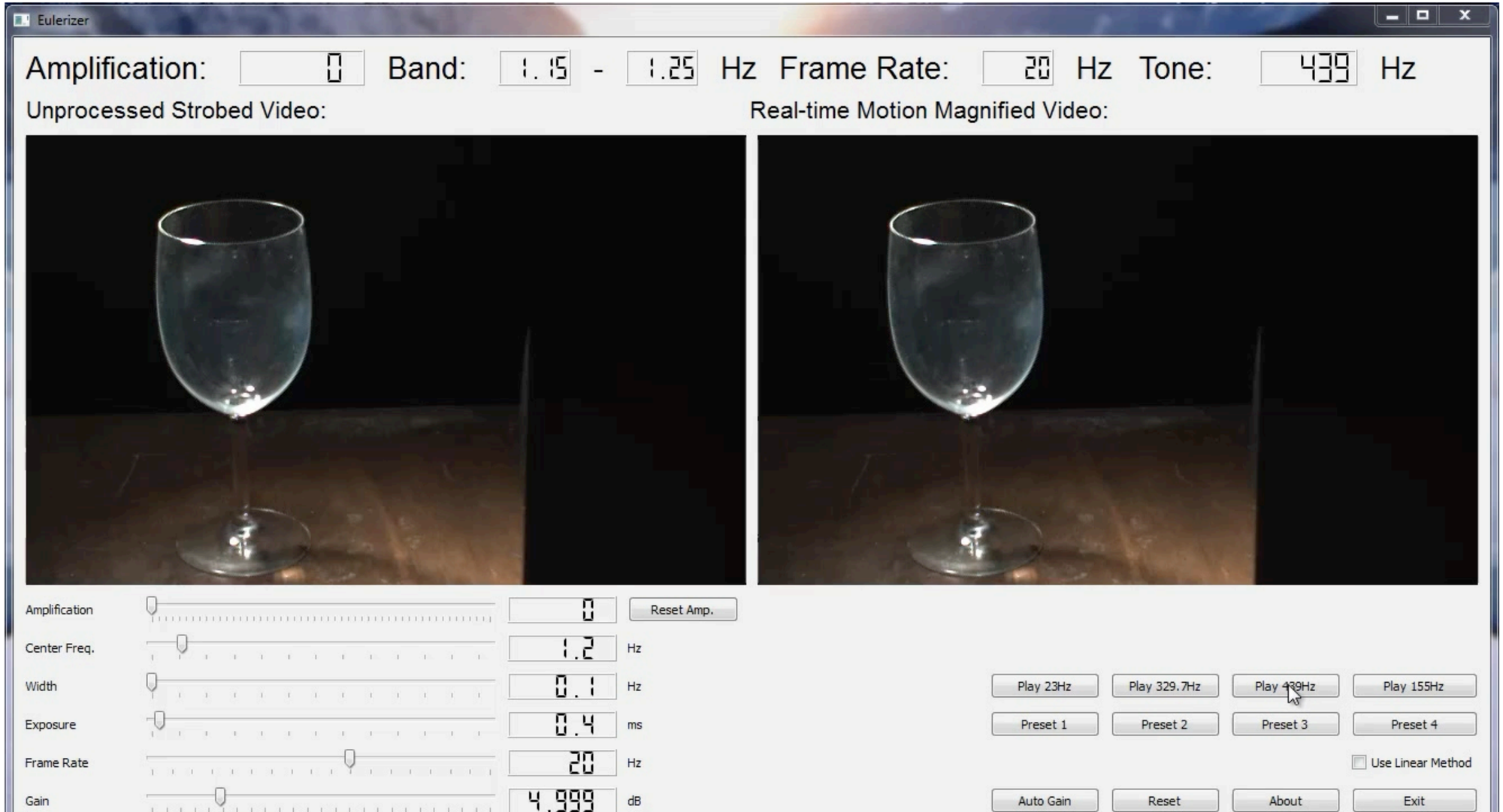


Application: Phase-based video processing



Glass demo

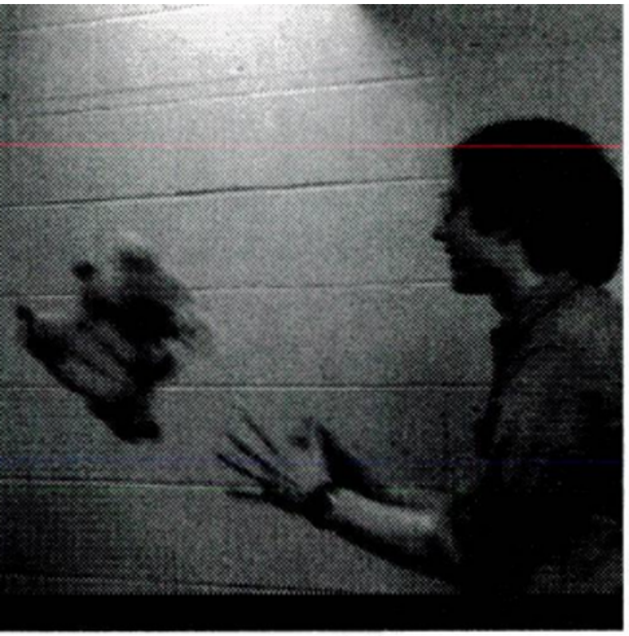
- Phase-based magnification + aliasing (strobe effect)



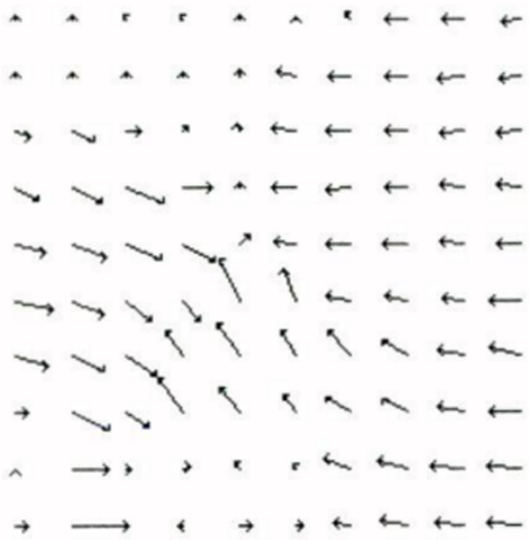
Other application: Motion without movement

Motion Without Movement

William T. Freeman†, Edward H. Adelson†, and David J. Heeger§



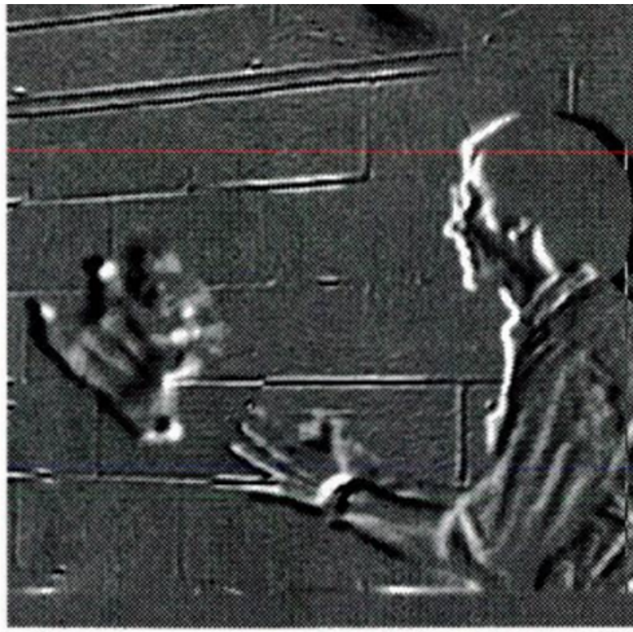
(a)



(b)



(c)

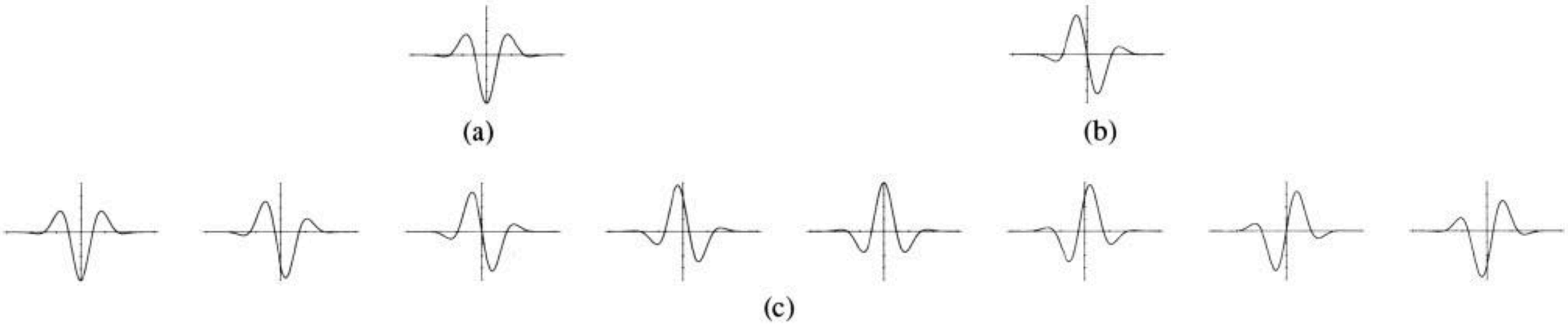


(d)



Motion without movement

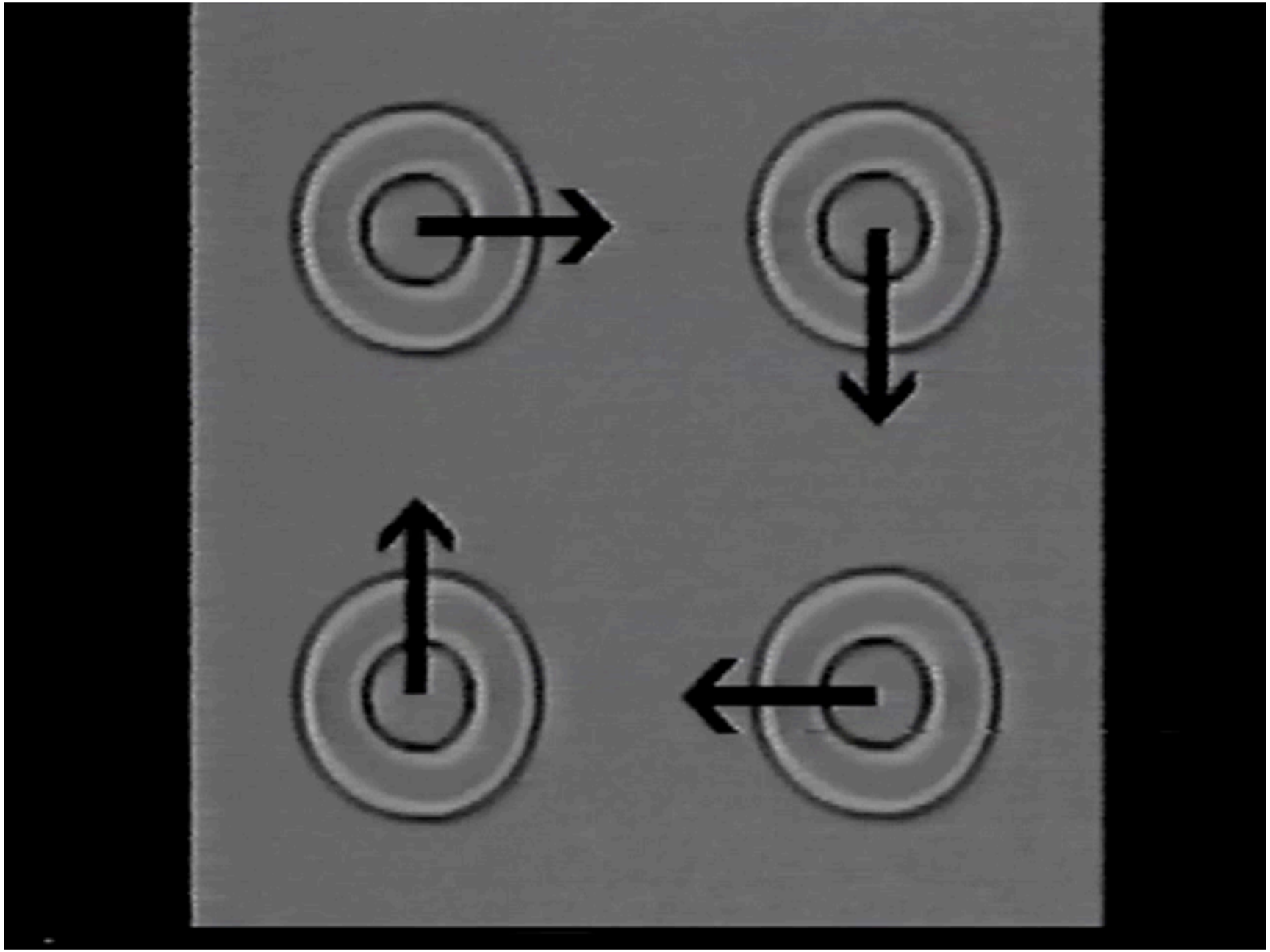
- Decompose image with steerable filters
- Change phase but mot magnitude to imply local motion
  - Steer according to some local motion field direction/magnitude



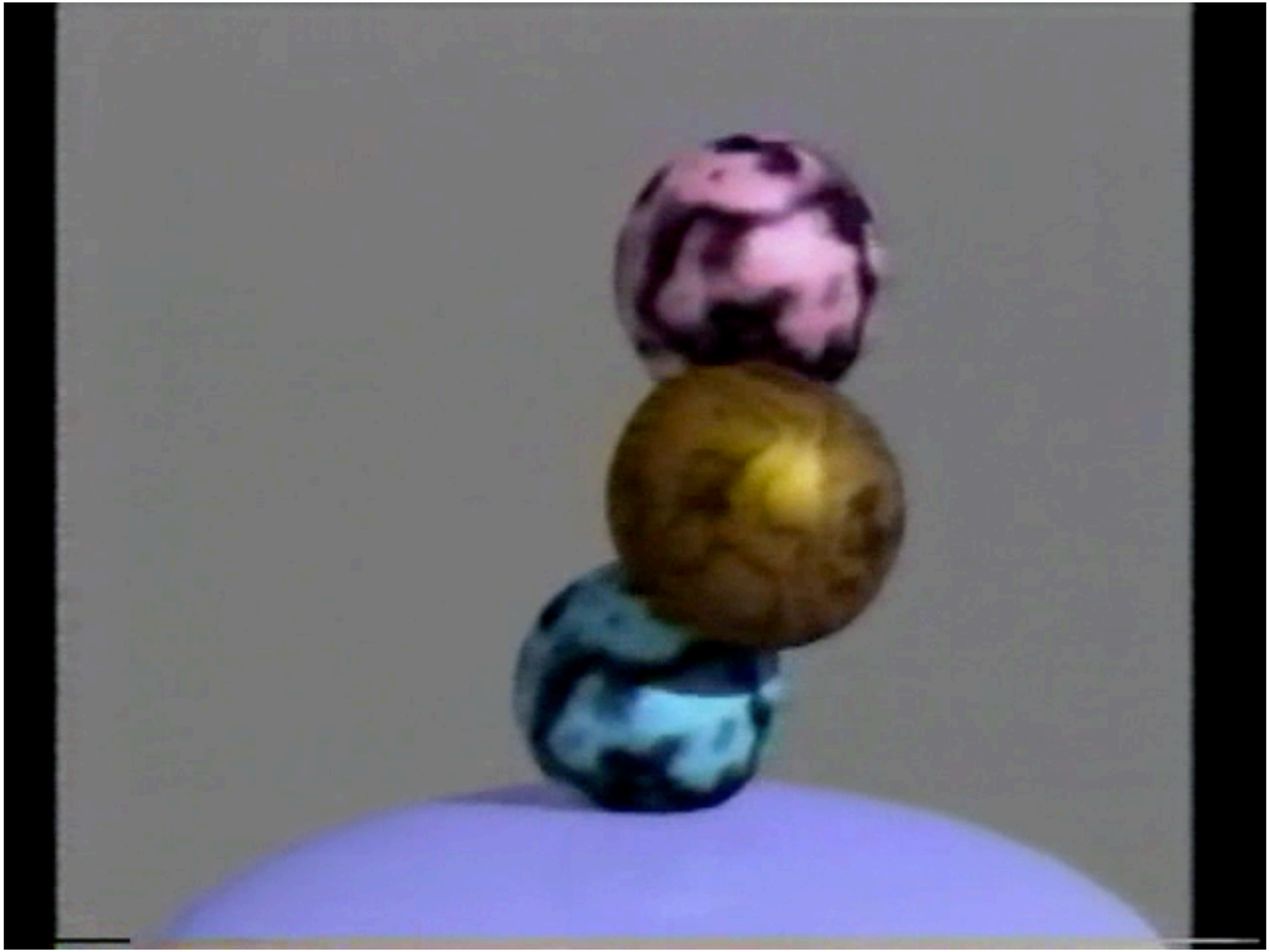


http://www.cse.yorku.ca/~kosta/Motion_Without_Movement/Motion_Without_Movement.html

Motion without movement



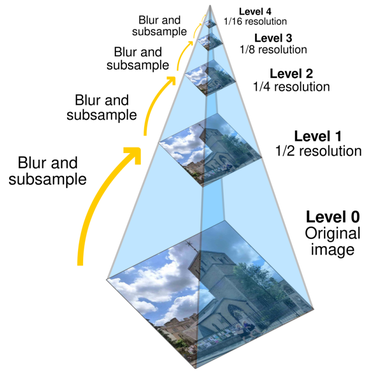
Motion without movement

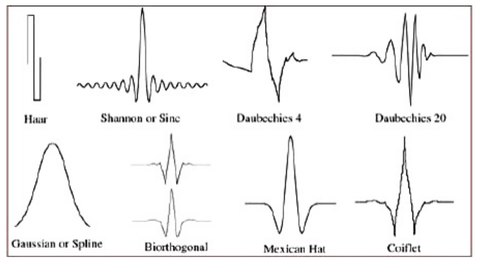


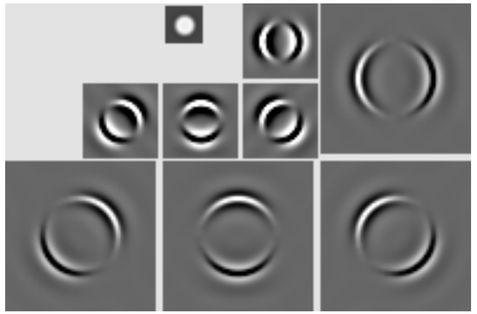
Other multi resolution representatio

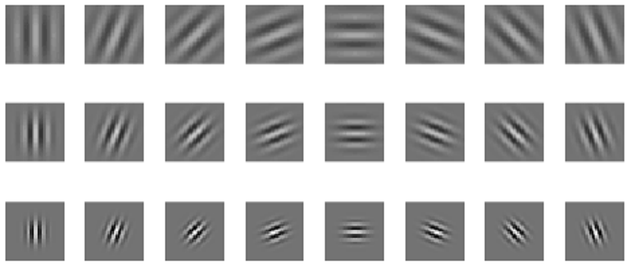
Other mutliresolution tools

- **Laplacian pyramids**
- **Wavelets**
  - Tighter (not over complete)
  - Less Fourier-friendly
  - Not shift invariant, not rotation invariant
  - Used a bunch for compression (JPEG 2000)
- **Steerable pyramids**
  - Add a notion of direction
  - Even more over complete
  - Very nicely shift/rotation invariant
- **Gabor functions/transform**







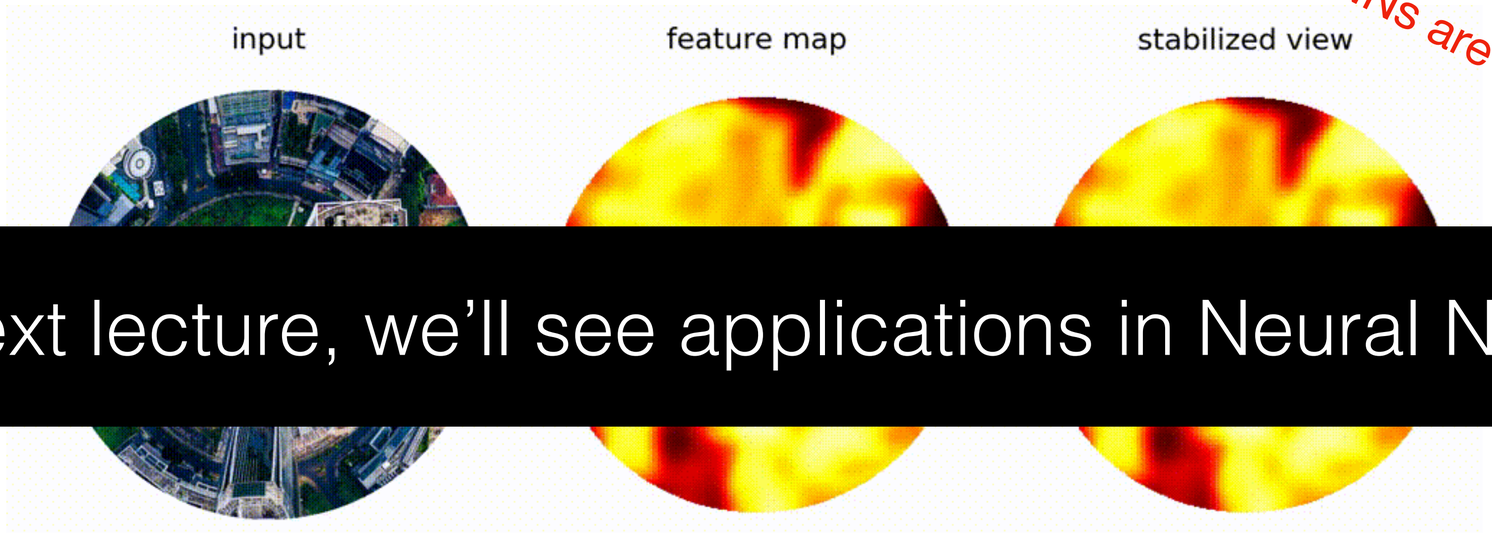


	Steerable Pyramid	Separable Orthog. Wavelet	Laplacian Pyramid	Gabor (octave)	Block DCT
jointly-localized (space/frequency)	yes	yes (can be)	yes	not inverse	not in frequency
translation-equivariant (no aliasing)	yes (approx)	no	yes (approx)	no	no
oriented kernels	yes	no (not diagonals)	N/A	yes	no
rotation-equivariant (steerable)	yes (approx)	no	N/A	no	no
tight frame (self-inverting)	yes (approx)	yes	no	no	yes
overcompleteness	4k/3	1	4/3	1	1

https://www.cns.nyu.edu/~eero/steerpyr/

Geometric guarantees (equivariance)

CNNs are not rotation equivariant

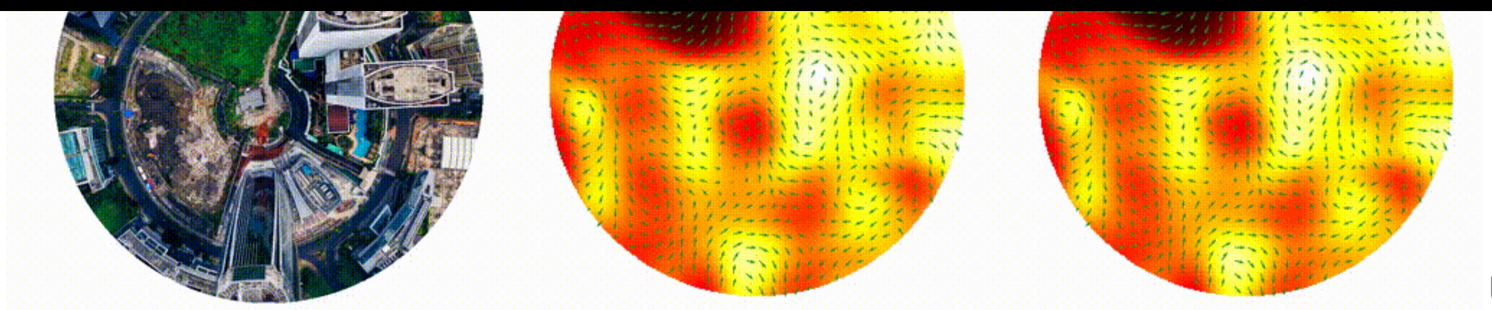


CNN

In the next lecture, we'll see applications in Neural Nets!

E

How can we *find* a steerable basis for a given transformation?





Figures source:
https://github.com/QUVA-Lab/e2cnn