Pyramids

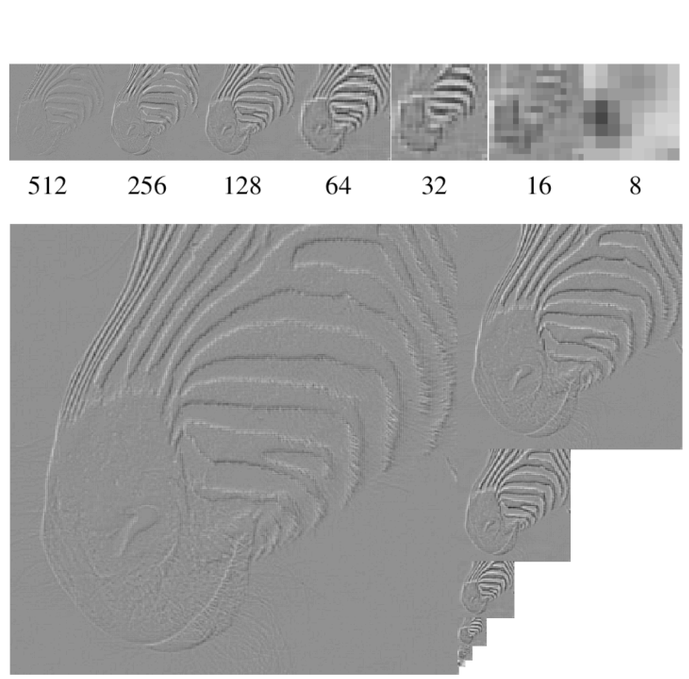
Server issues, extensions

- Sorry about the problems
- Both panorama stitching problems sets will be delayed by two days
  (due on Fridays 10/24 and 11/7)

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
  - Key points
  - CNN & UNet





Original papers from the 80s

- Burt & Adelson ; Williams

532 IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-3l, 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

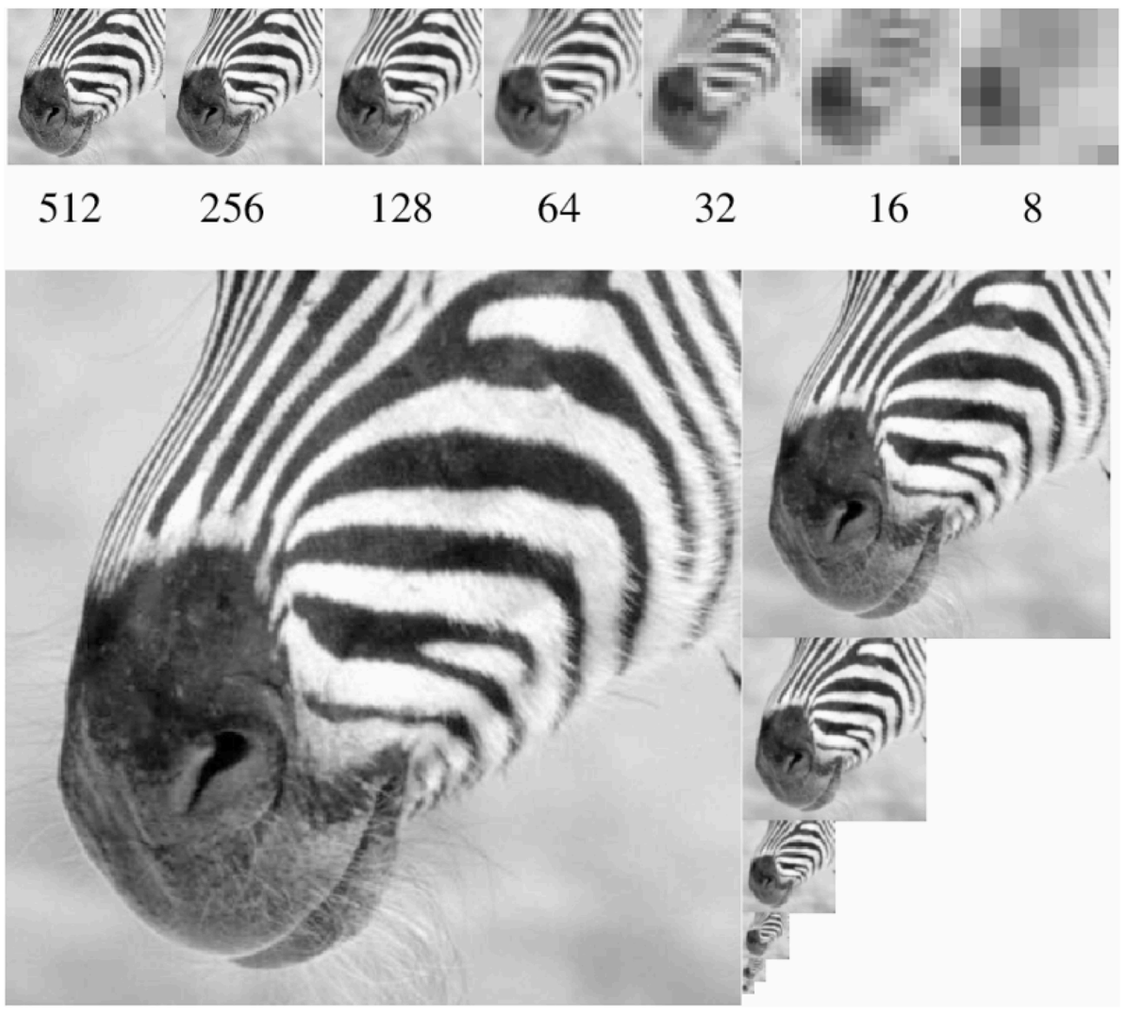
Wim Sweldens

Katholieke Universiteit Leuven Belgium, Department of Computer Science

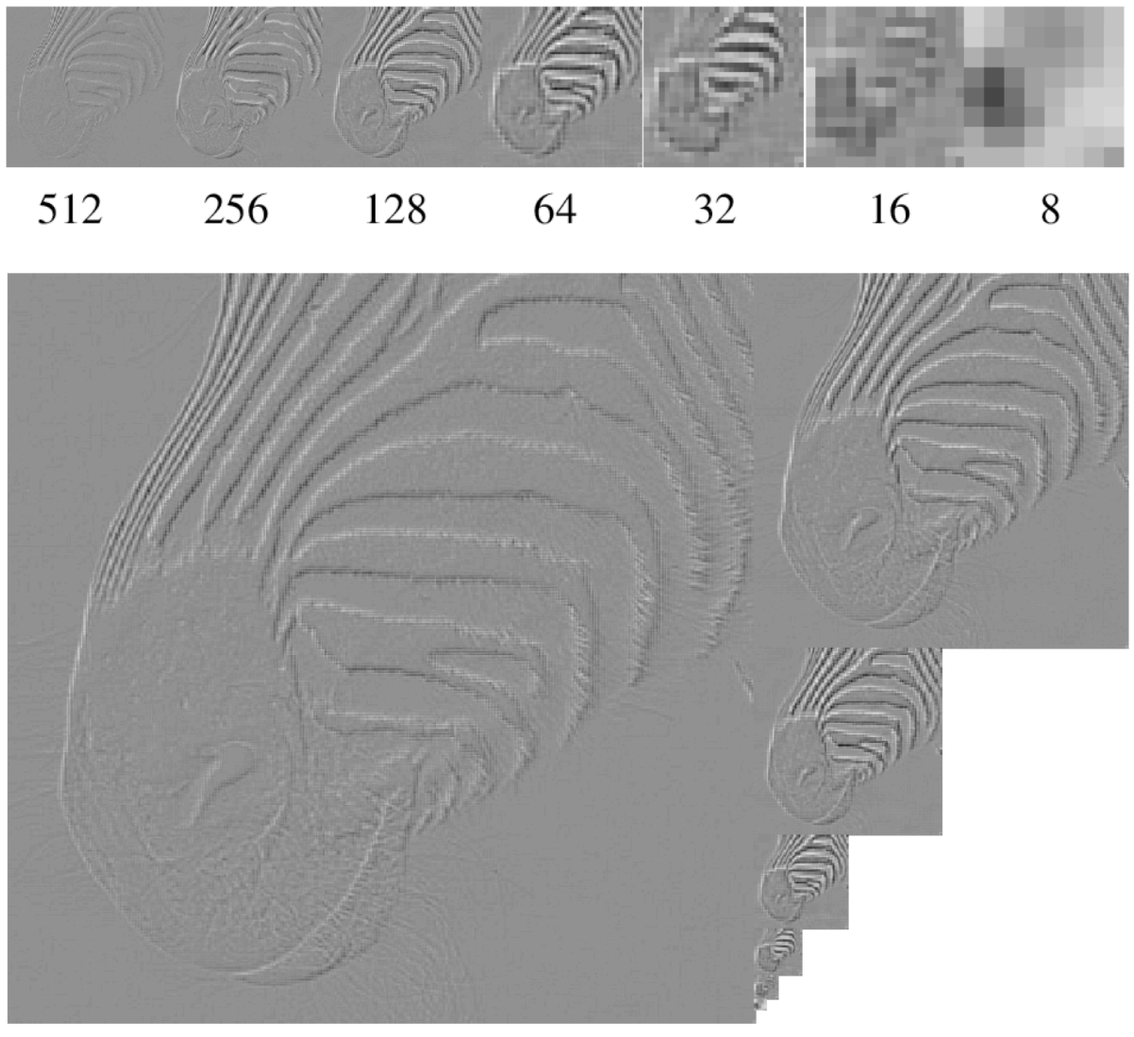
Wim.Sweldens@cs.kuleuven.ac.be

Two main families

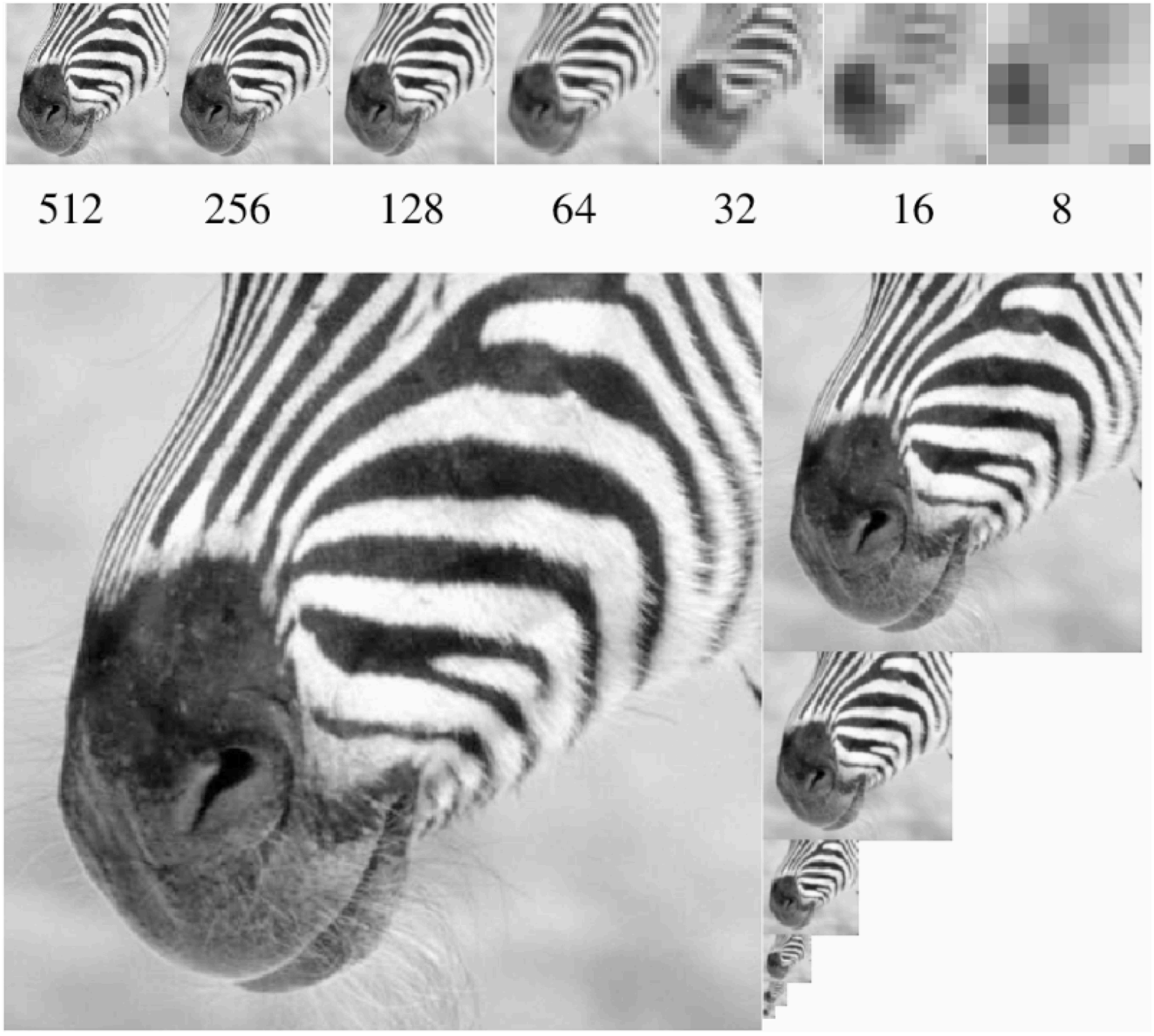
Low pass
Each level is a lower resolution of the original



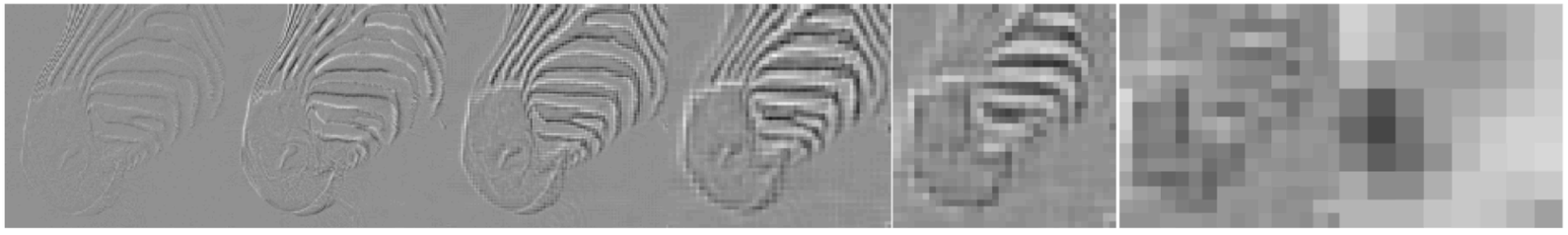
Band pass:
Only narrow frequencies per level



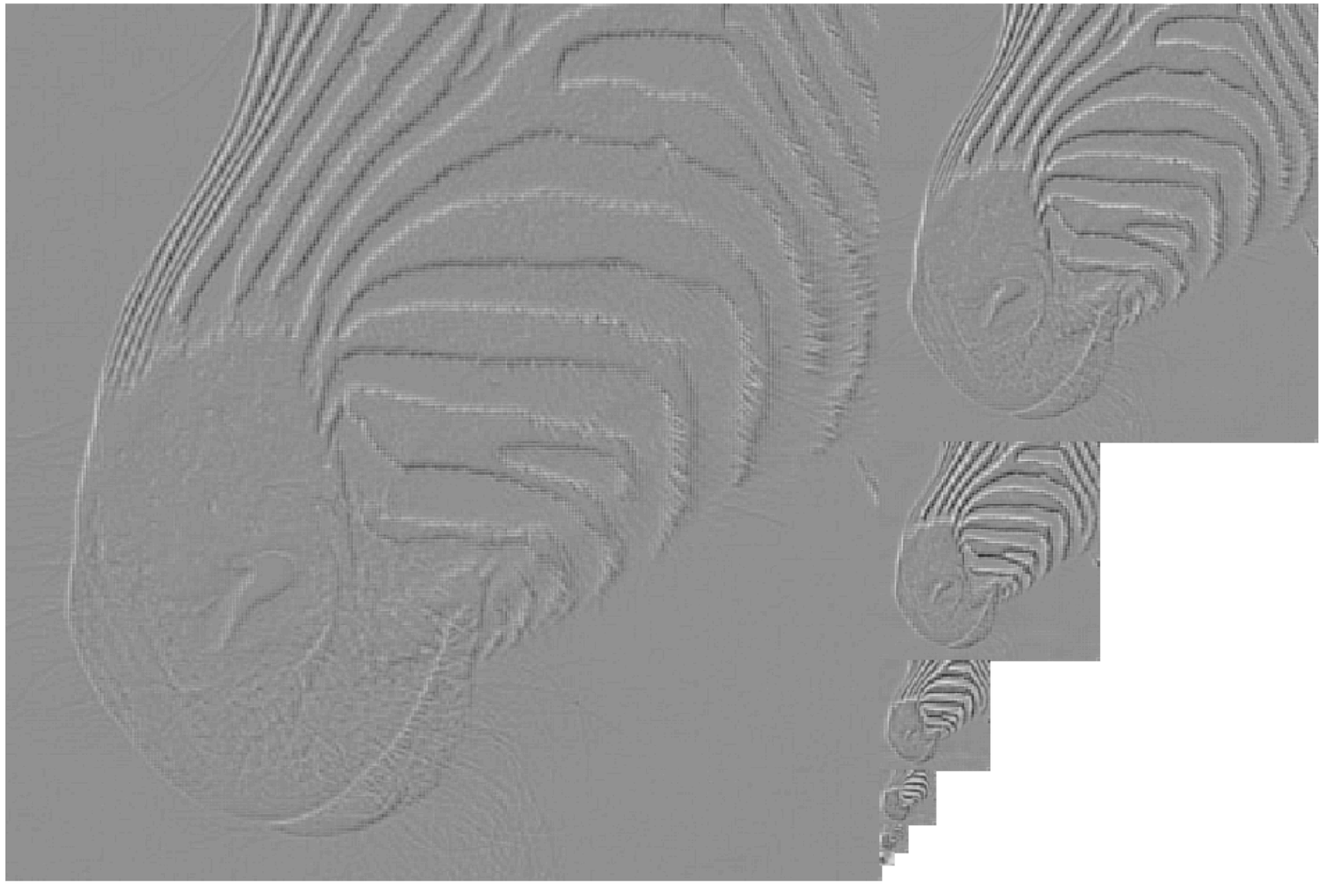
Gaussian pyramid



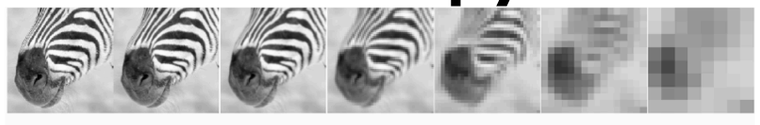
Laplacian pyramid



512 256 128 64 32 16 8



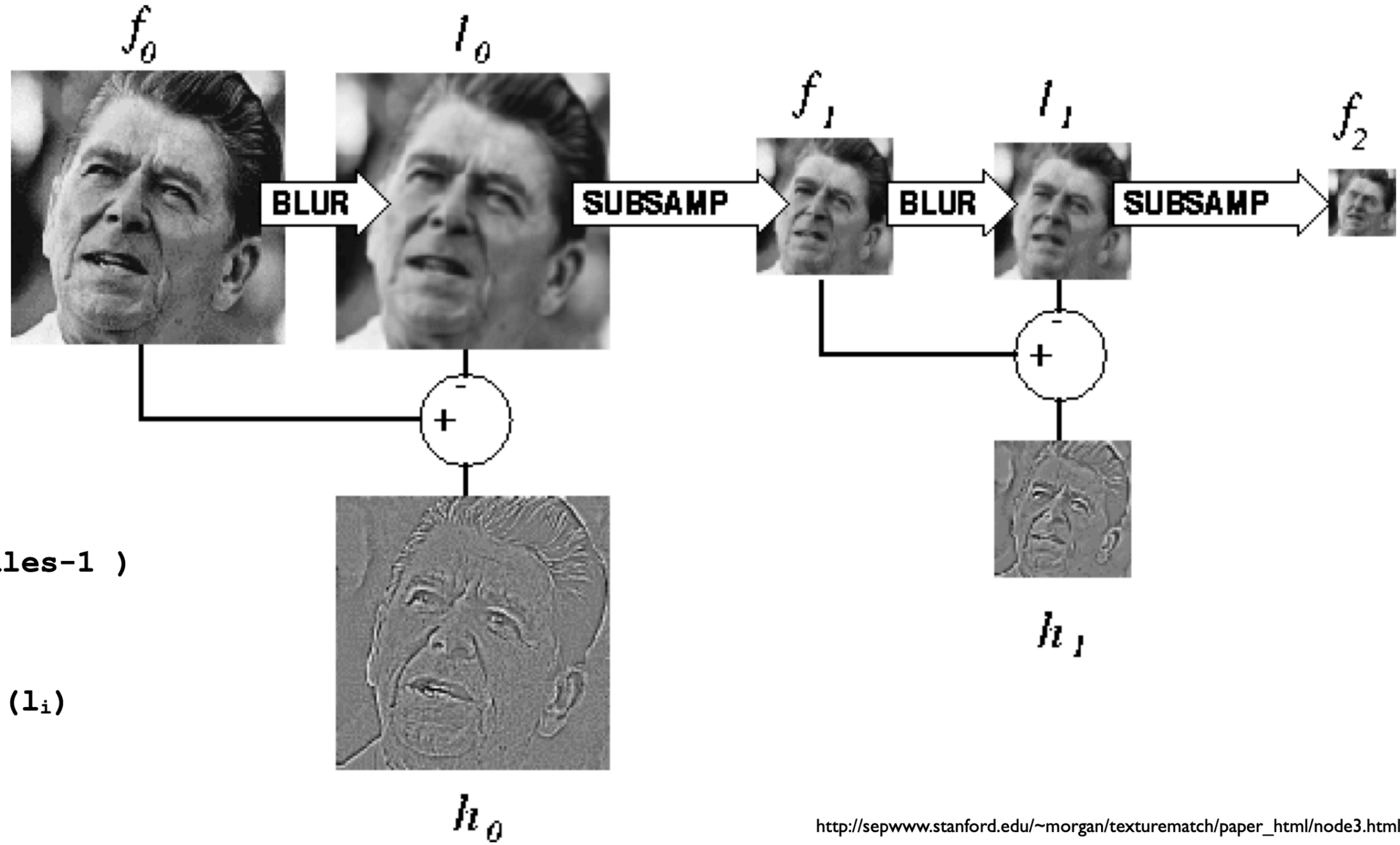
Essentially difference between levels of th Gaussian pyramid



512 256 128 64 32 16 8



Constructing a Laplacian Pyramid




```
do( i = 0 : nScales-1 )
{
  l_i = blur(f_i)
  h_i = l_i - f_i
  f_i+1 = subSamp2(l_i)
}
```

http://sepwww.stanford.edu/~morgan/texturematch/paper_html/node3.html

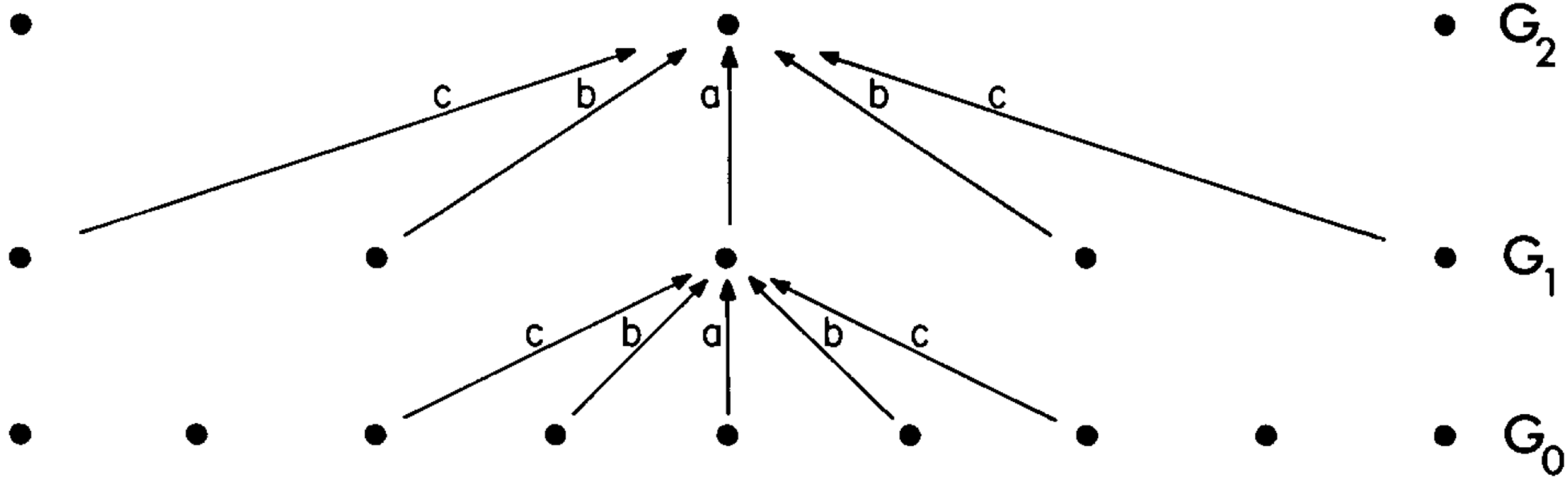
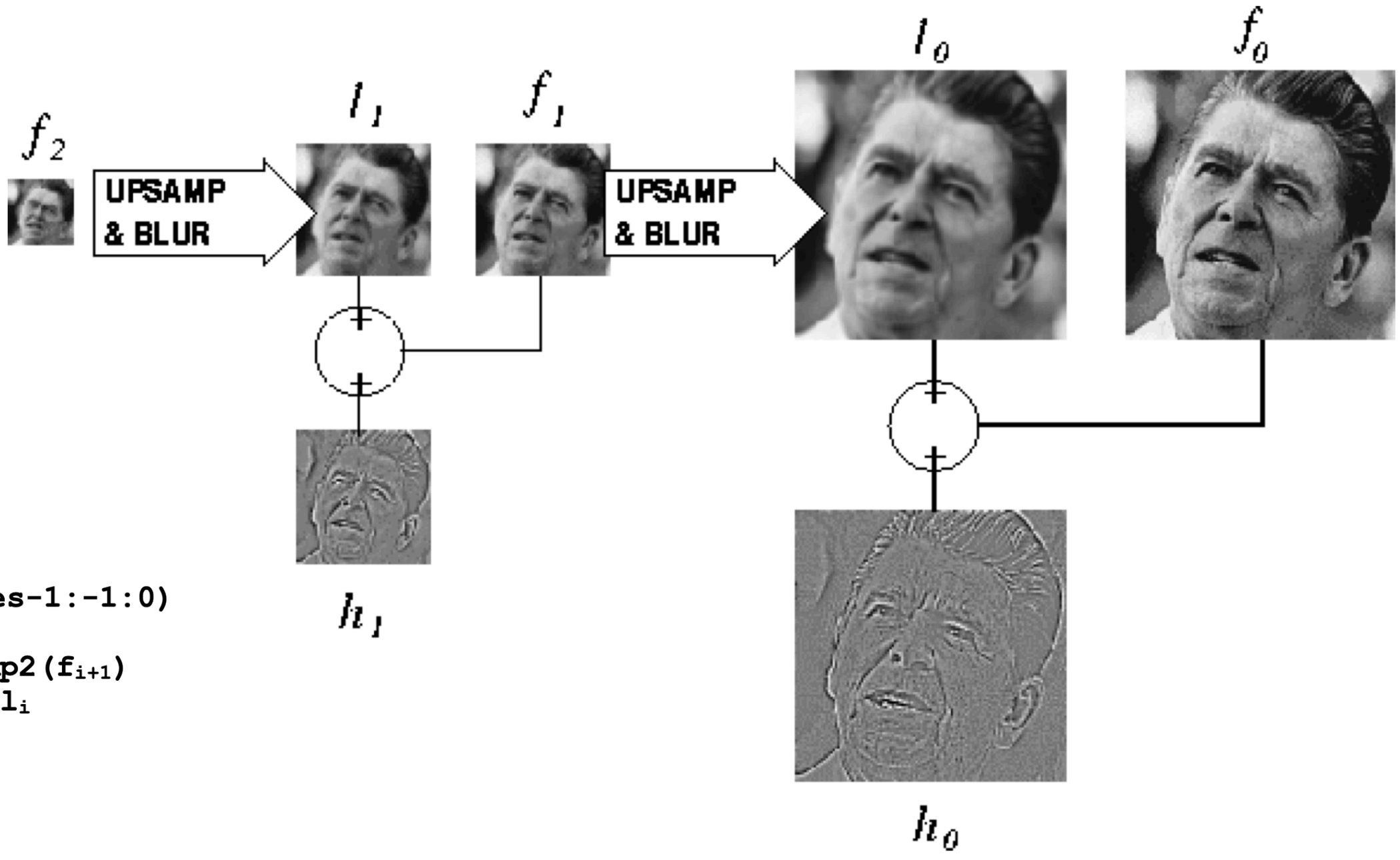


Fig. 4. A one-dimensional graphical representation of the iterative REDUCE operation used in pyramid construction.

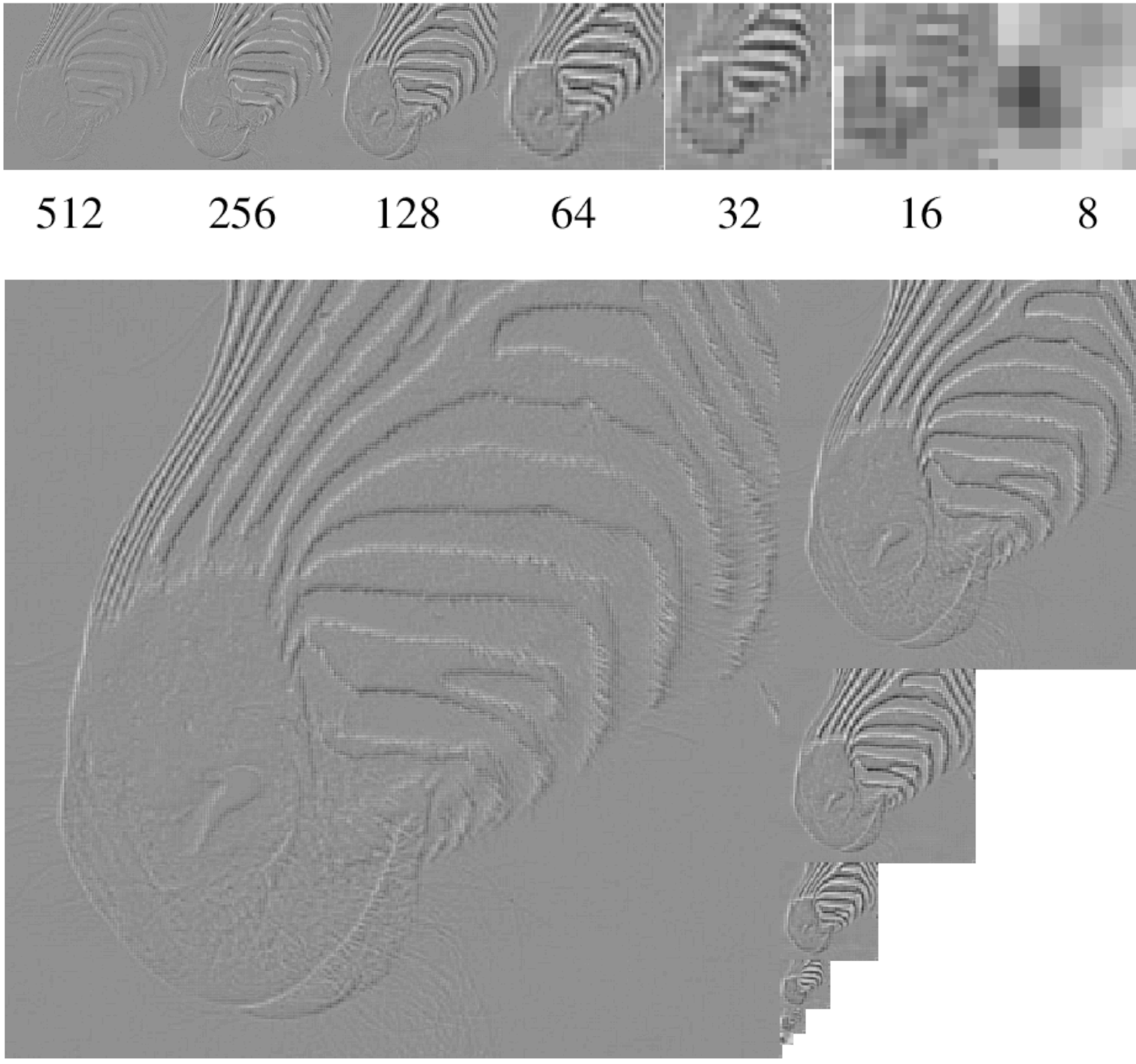
Reconstructing the image




```
do( i = nScales-1:-1:0)
{
     li = upSamp2(fi+1)
     fi = hi + li
}

output: f0
```

Questions?

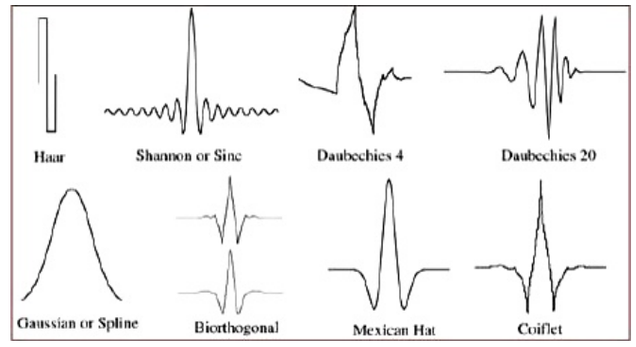


Properties of Laplacian Pyramid

- Reasonably shift invariant
- Overcomplete
  (more numbers than original image)
- Close to Fourier decomposition but more local **spatially**
  - And thus less local in Fourier
- **Isotropic**
  - Reasonably rotation invariant
  - No notion of direction
  - No preferred direction

Other mutliresolution tools

- **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

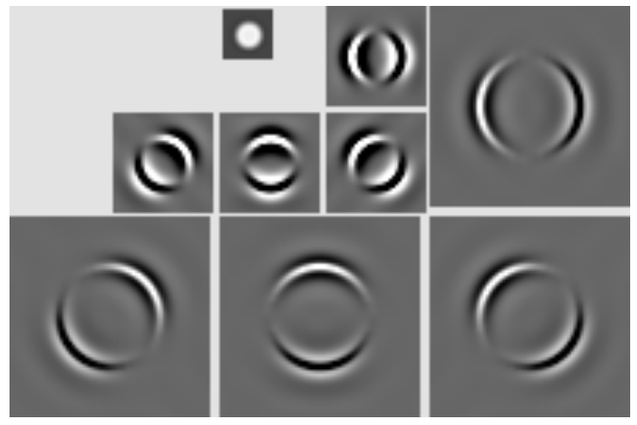
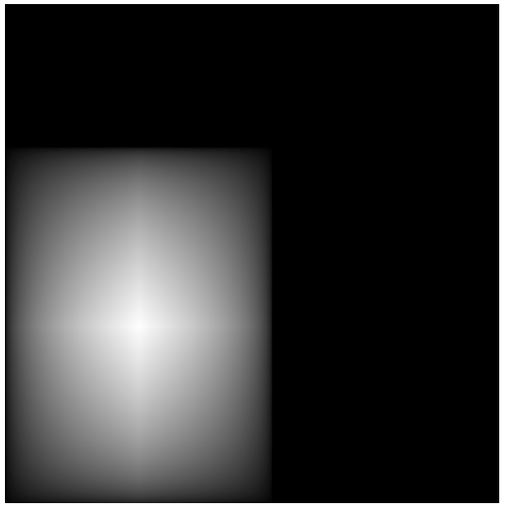
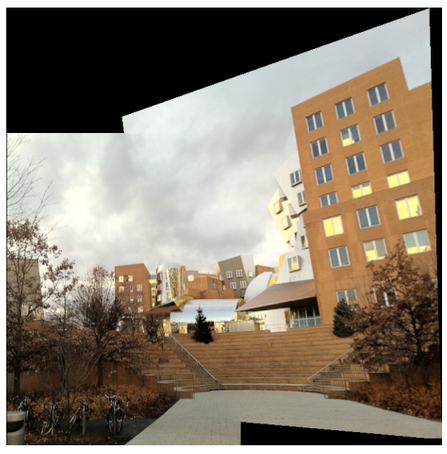
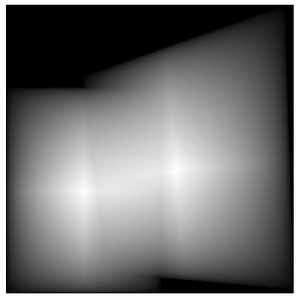
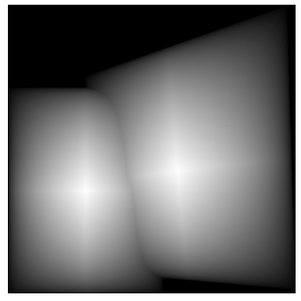
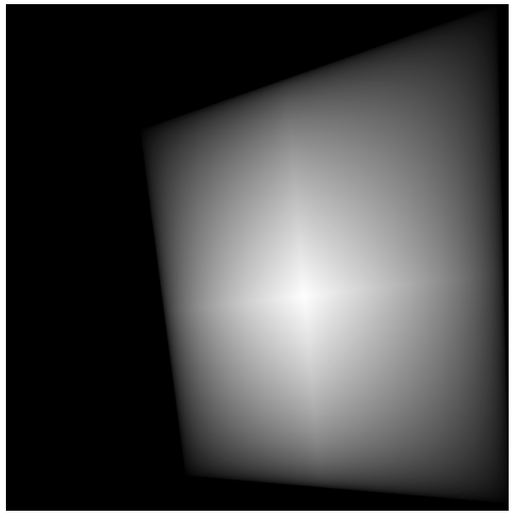


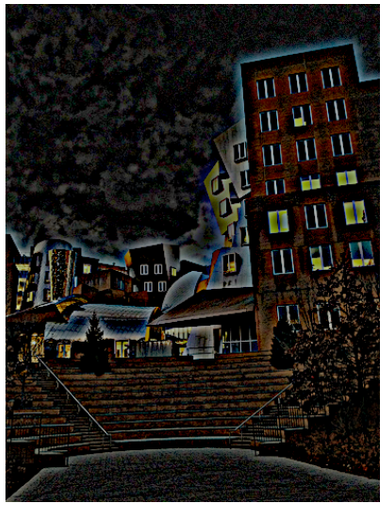
Image blending

2-scale separation





Multiscale blending

- Proposed by Adelson and Burt
  - http://persci.mit.edu/pub_pdfs/spline83.pdf
- Simplified to two scales by Brown and Lowe
  - http://cs.bath.ac.uk/brown/papers/iccv2003.pdf

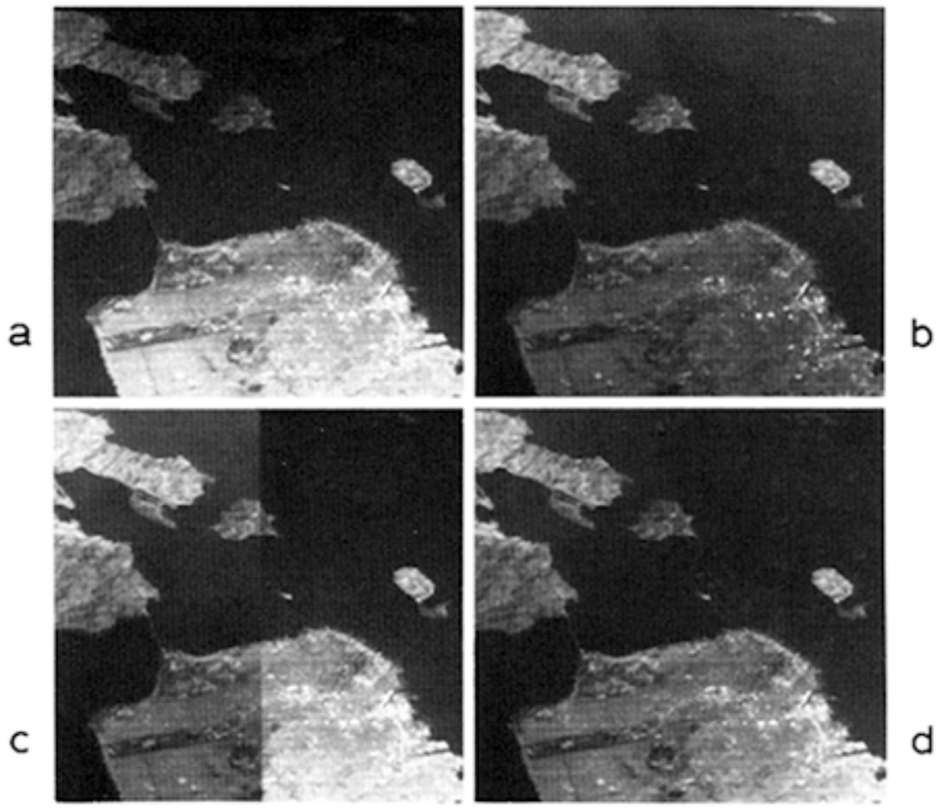


Fig. 6. The spline applied to Landsat images of San Francisco. When the left half of Figure 6a is joined to the right half of Figure 6b without a spline, the boundary is clearly visible (Figure 6c). No boundary is visible when the multiresolution spline is used (Figure 6d).



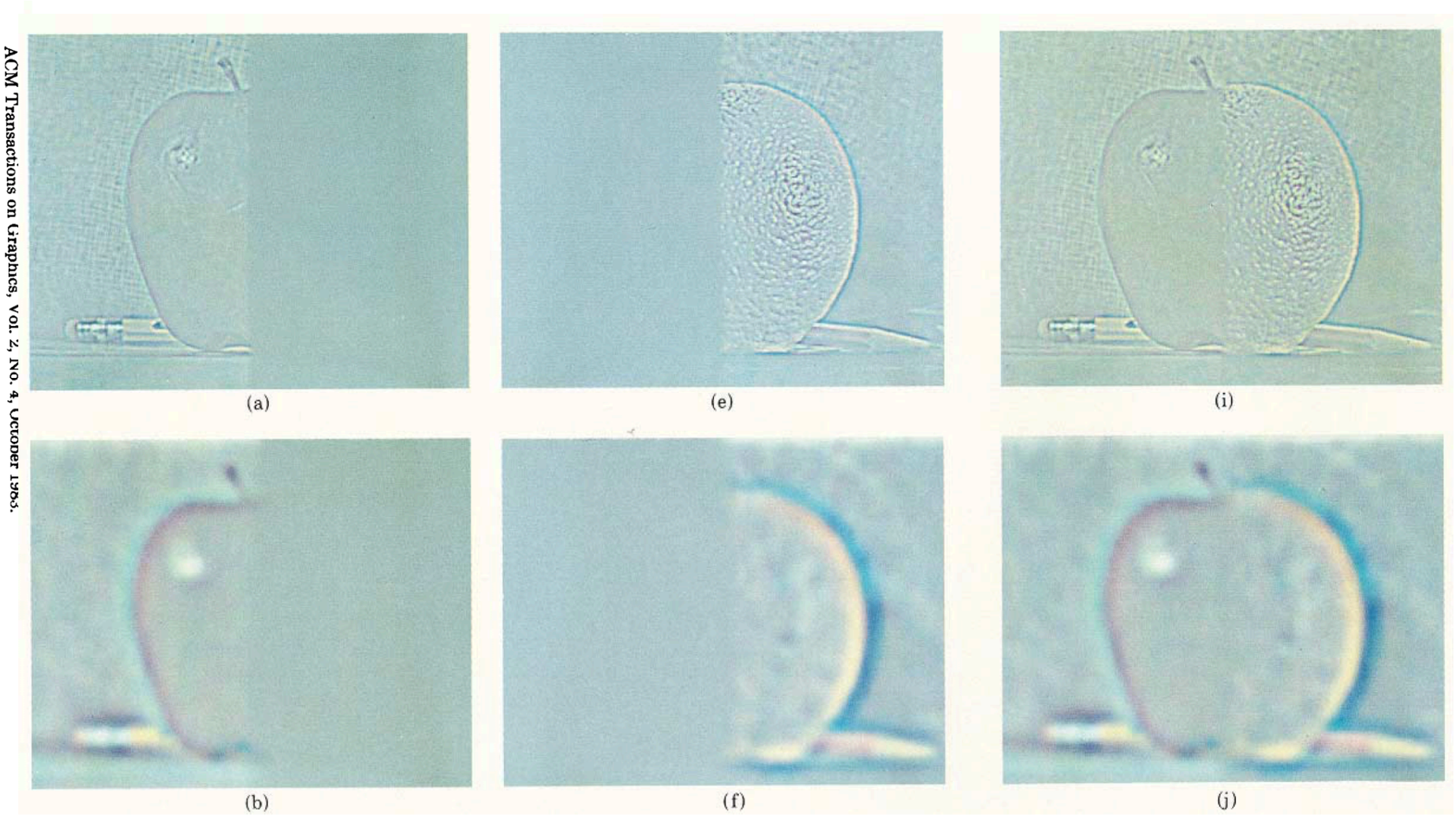
(b) All 80 images registered



(c) Rendered with multi-band blending

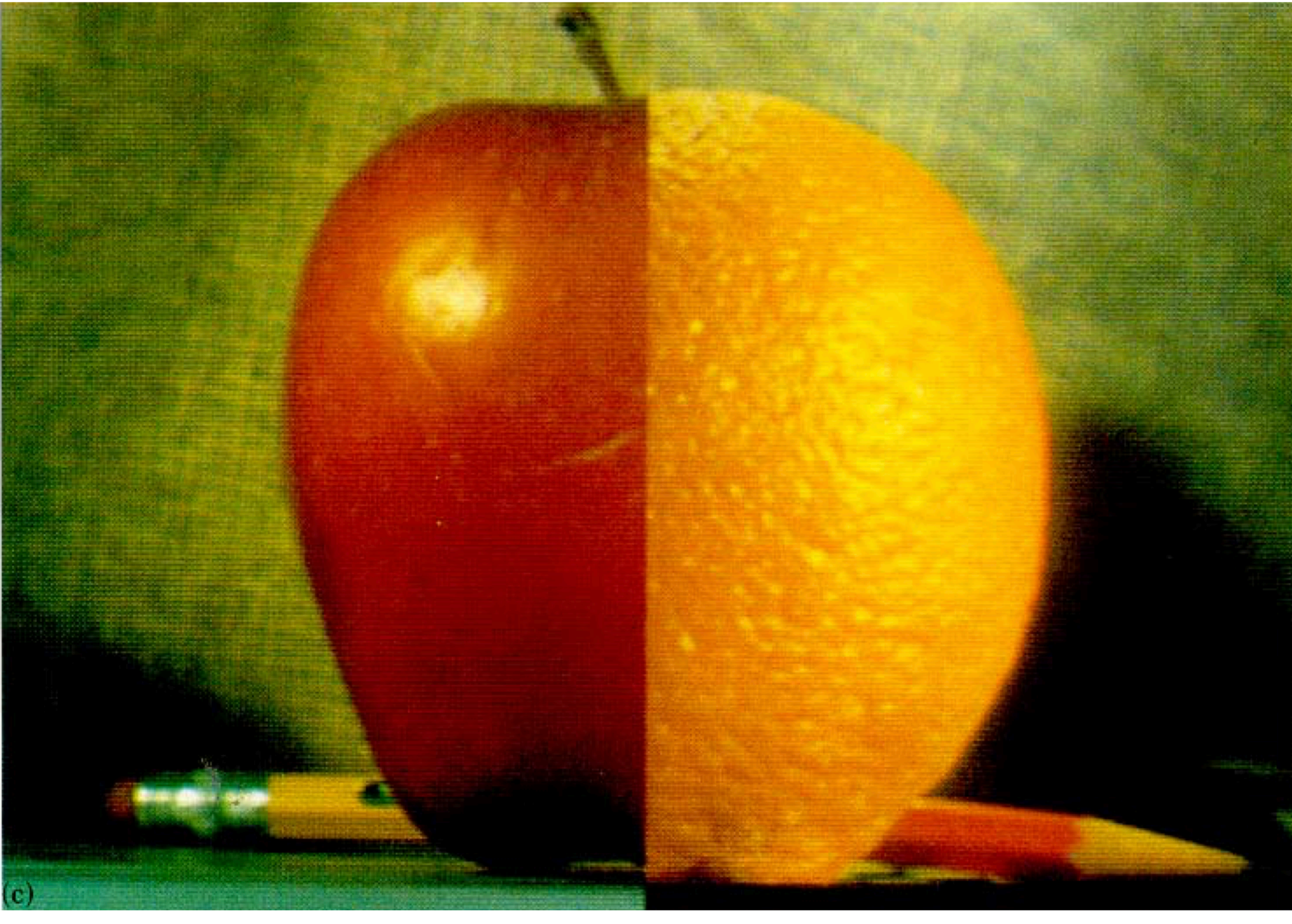
Pyramid blending

- 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


(c)

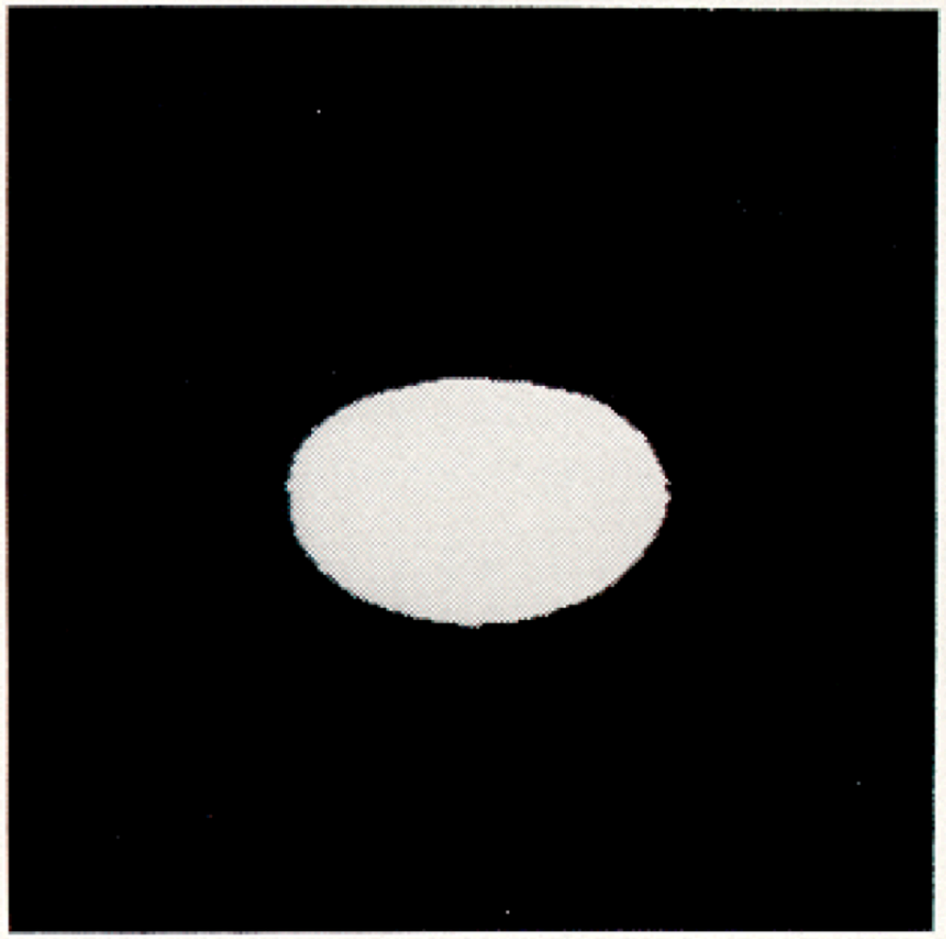
(d)



a



b



c



d

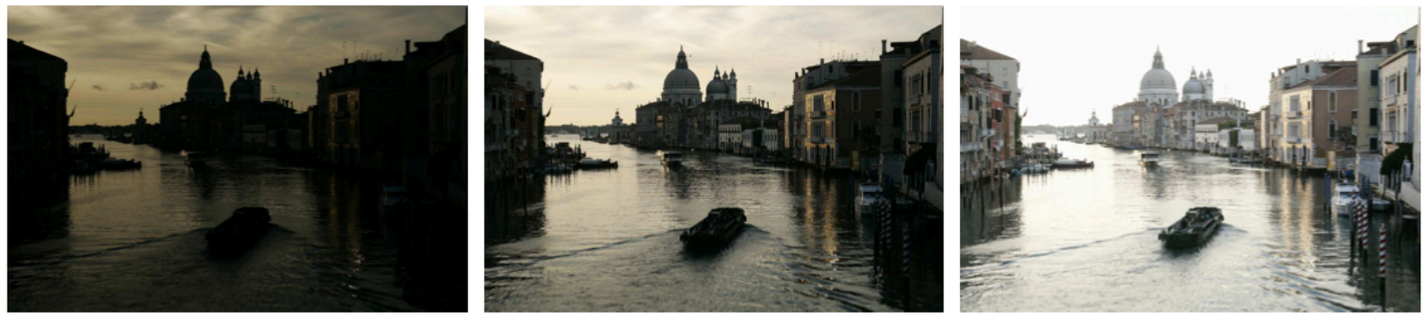
Questions?

Exposure

Fusion

Back to HDR tone mapping

- Kautz & Mertens
- Idea: combine HDR merging & tone mapping
- Similar to panorama blending
- Just use previous technique

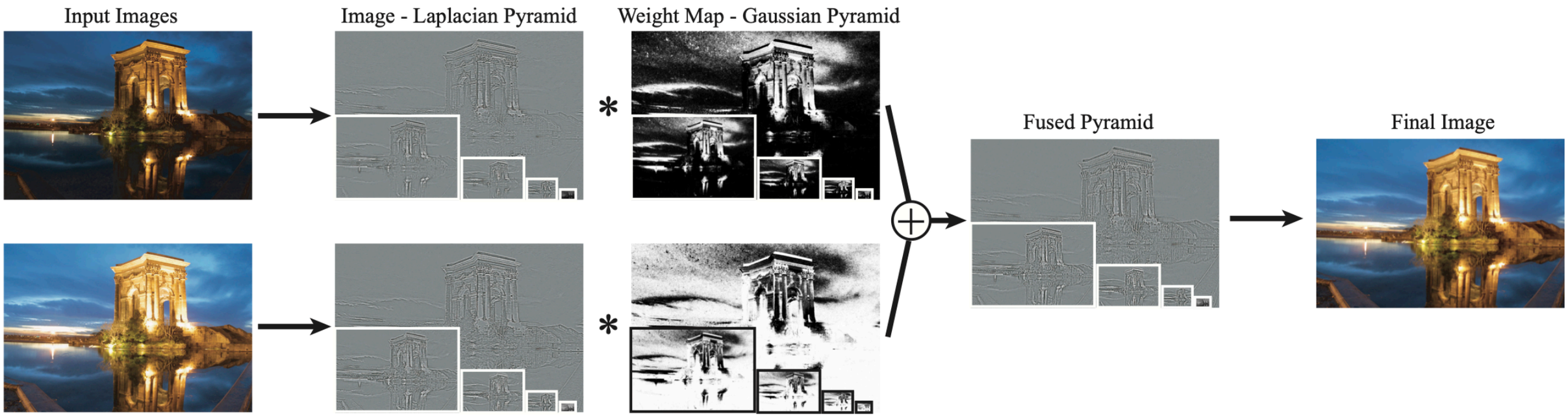


(a) Exposure bracketed sequence



(b) Fused result

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


Input Images
Image - Laplacian Pyramid
Weight Map - Gaussian Pyramid
Fused Pyramid
Final Image



(a) Input images with corresponding weight maps

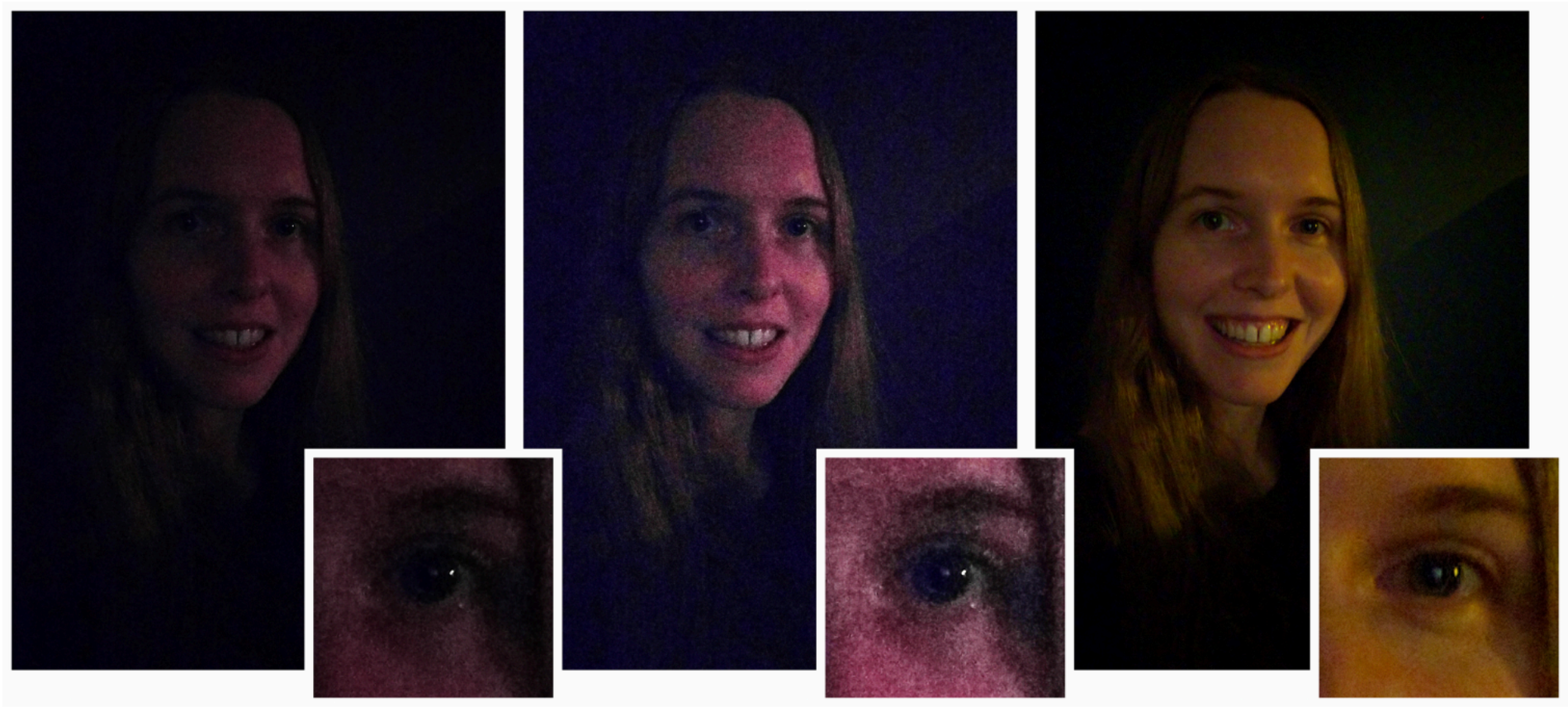
(b) Fused result

HDR+ on Android

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

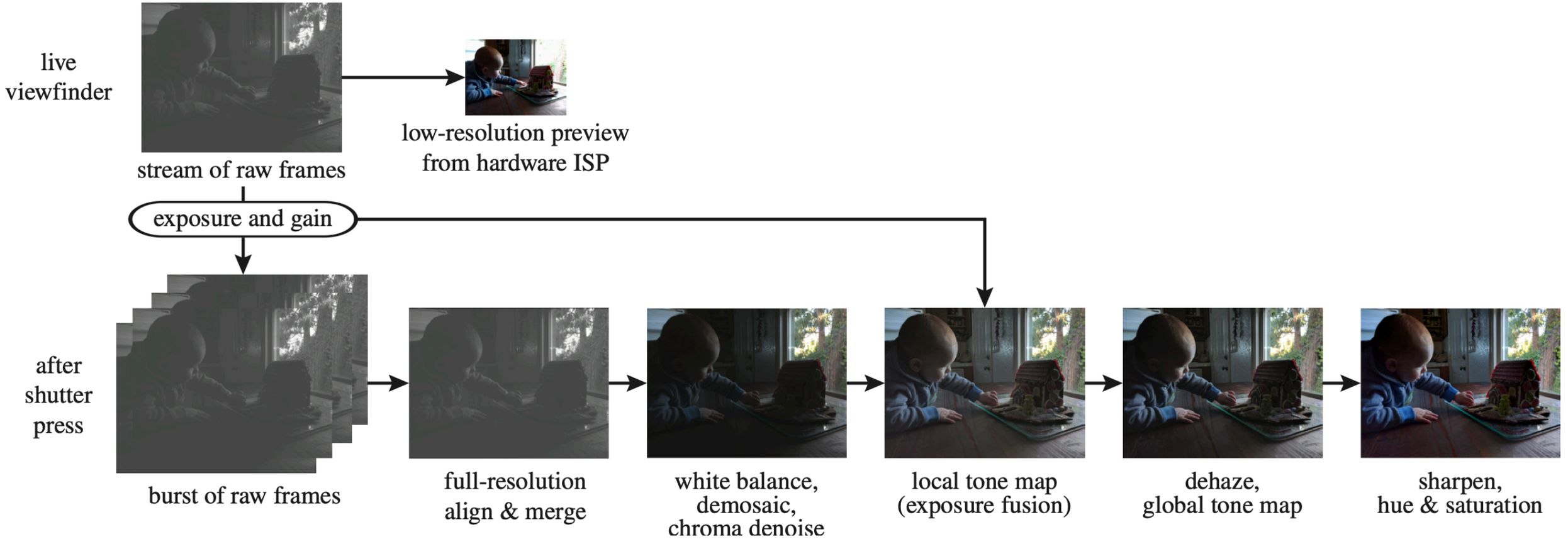
Burst photography for high dynamic range and low-light imaging on mobile cameras

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



Tone mapping in HDR+

- **Two virtual exposures**
- **Only done on brightness (and then re-apply chroma ratios)**



Pyramid Denoising

Pyramid coring/denoising

- 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

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**

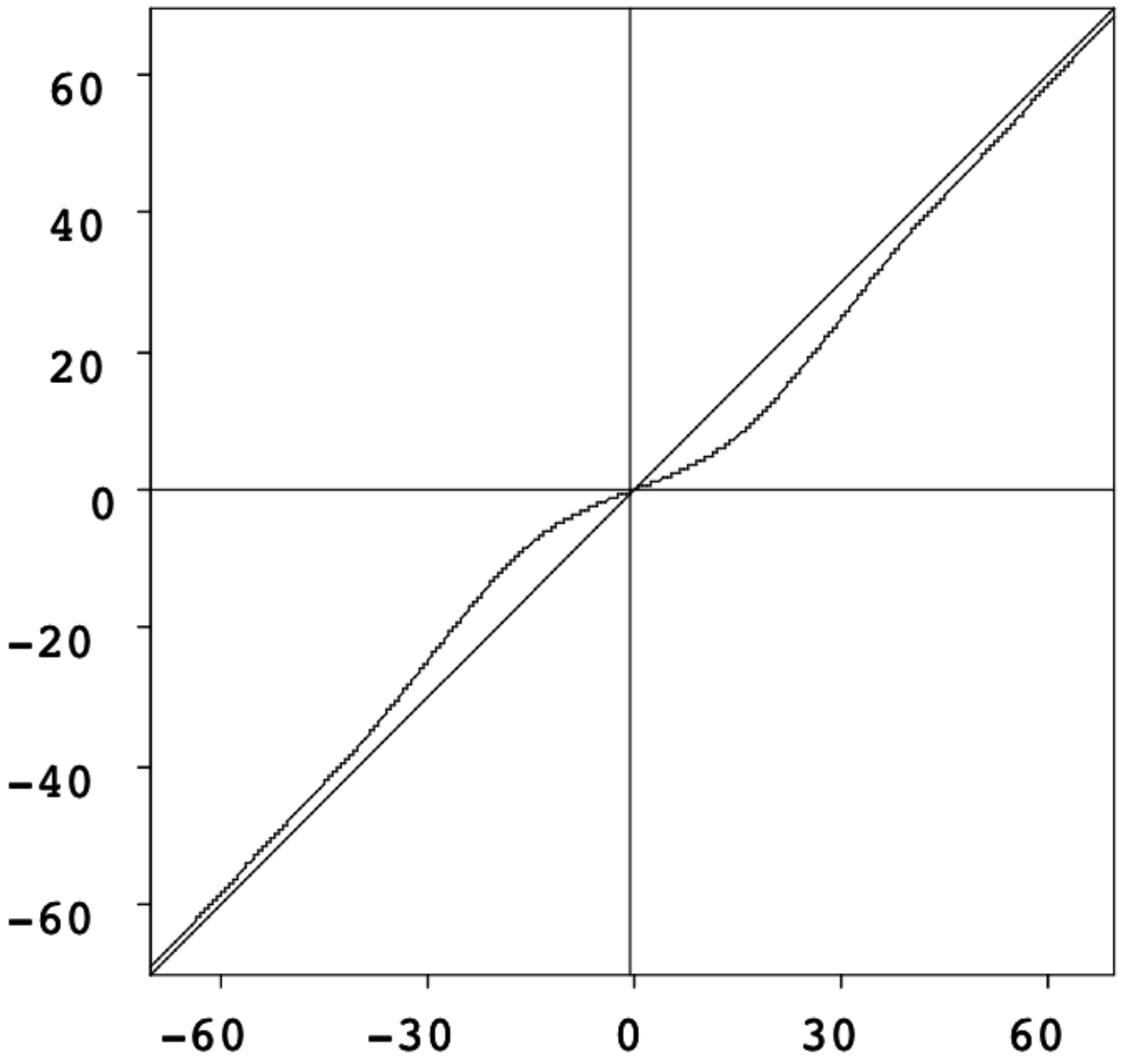
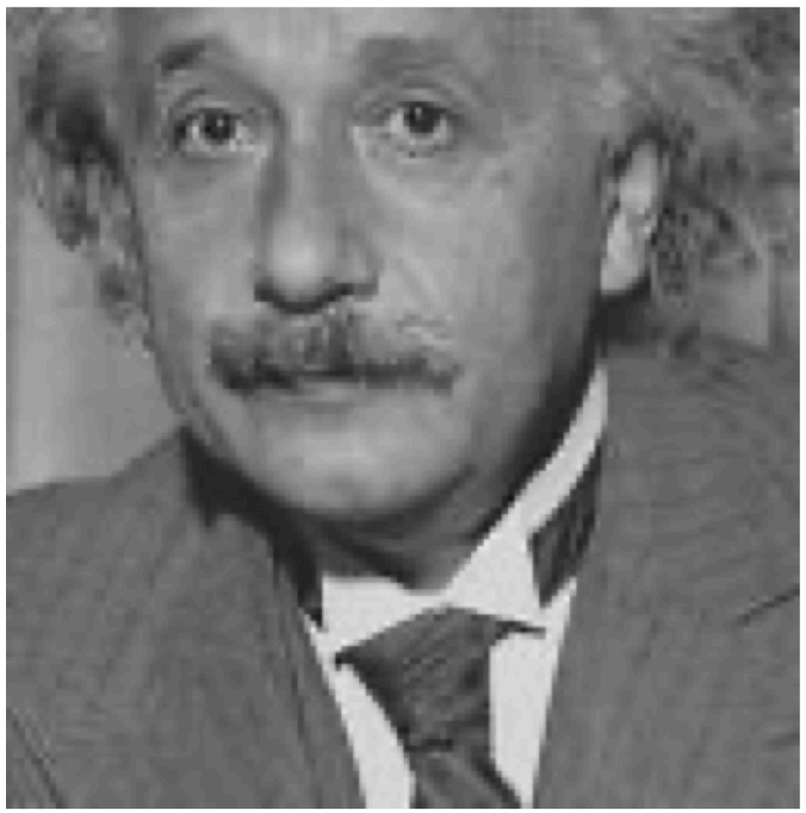
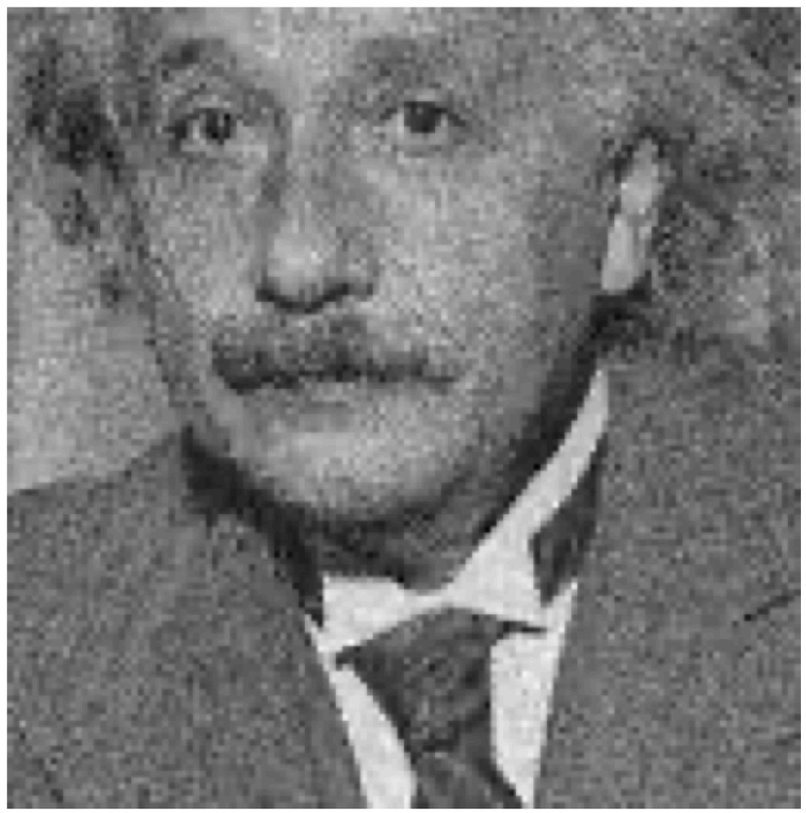


Figure 2. Bayesian estimator (symmetrized) for the signal and noise histograms shown in figure 1. Superimposed on the plot is a straight line indicating the identity function.

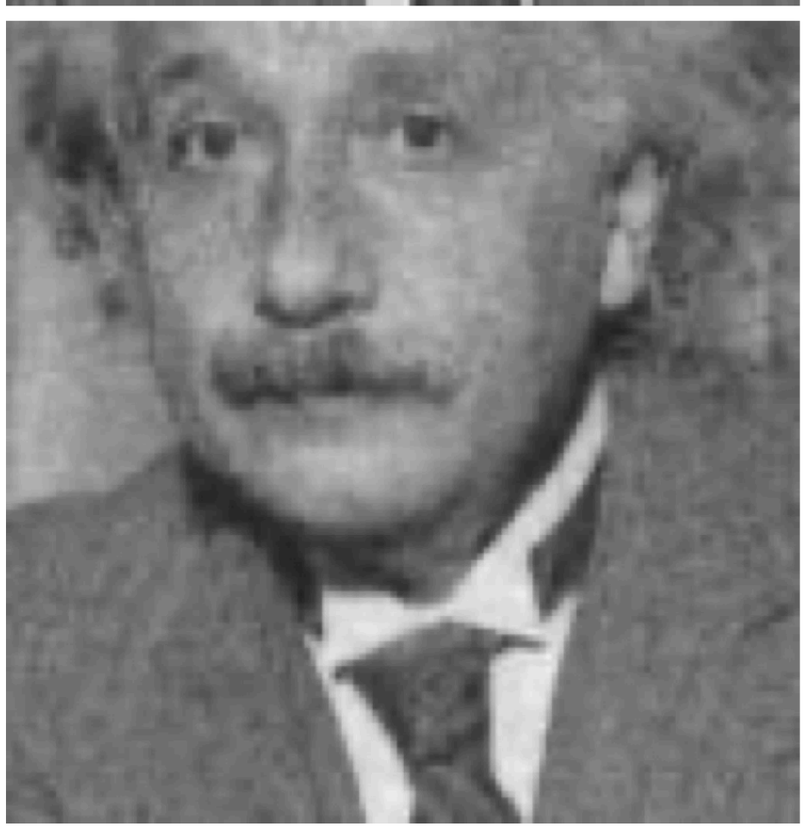
(a)



(b)



(c)



(d)

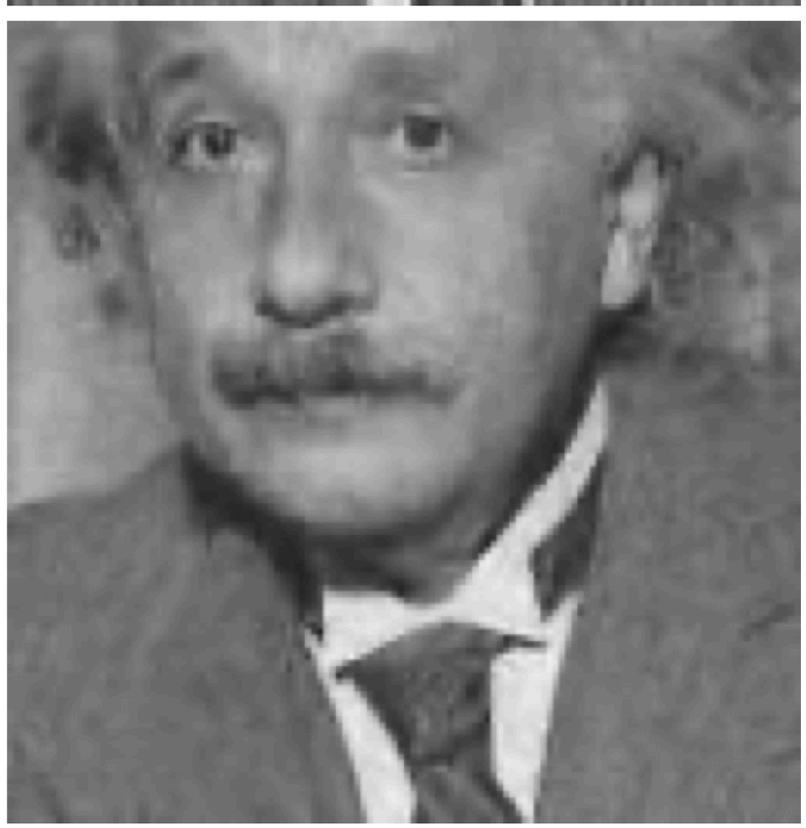


Figure 4. Noise reduction example. (a) Original image (cropped). (b) Image contaminated with additive Gaussian white noise (SNR = 9.00dB). (c) Image restored using (semi-blind) Wiener filter (SNR = 11.88dB). (d) Image restored using (semi-blind) Bayesian estimator (SNR = 13.82dB).

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

Local Laplacian Filtering

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

Local Laplacian Filters:

Edge-aware Image Processing with a Laplacian Pyramid

Sylvain Paris, Adobe

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

Jan Kautz, University College London

Example Application: Increase Level of Detail



input

Naïve Solution: Boost High Frequencies



unsharp ma

Taking Edges into Account for Halo-free Result



our result

Our Strategy: Local Adaptation

1. Generate an image that looks good for a small neighborhood
   - It may look bad elsewhere

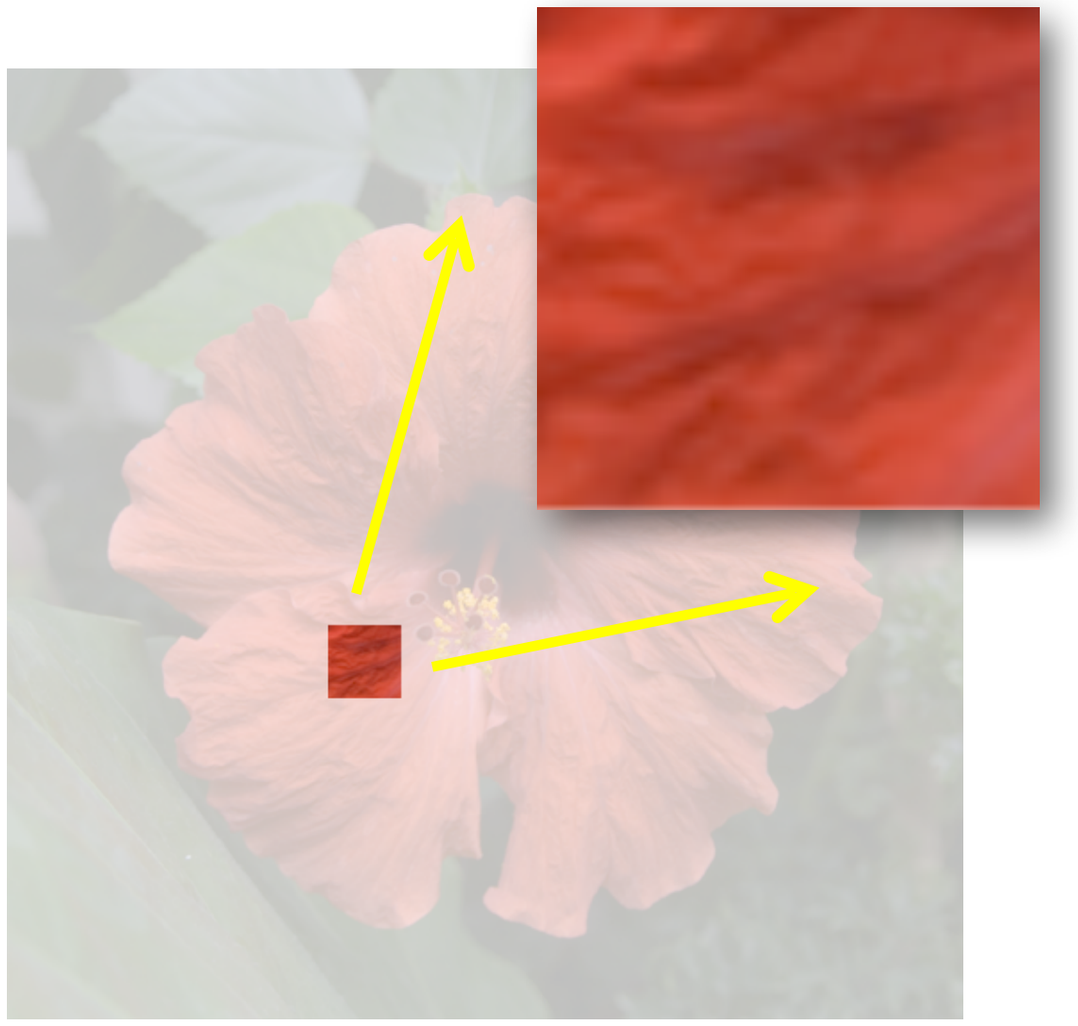
2. "Combine data from all neighborhoods"

Example: Local Contrast Increase



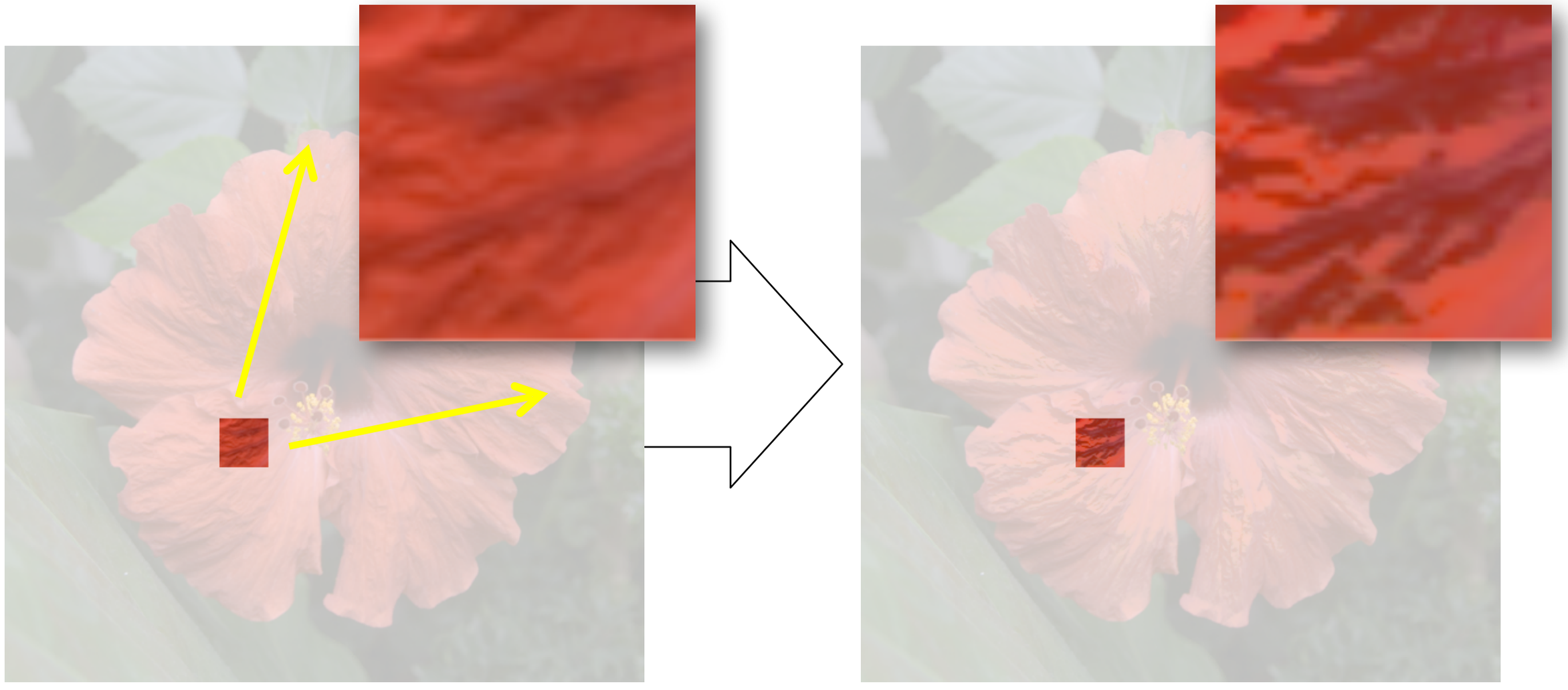
input

Example: Local Contrast Increase



input

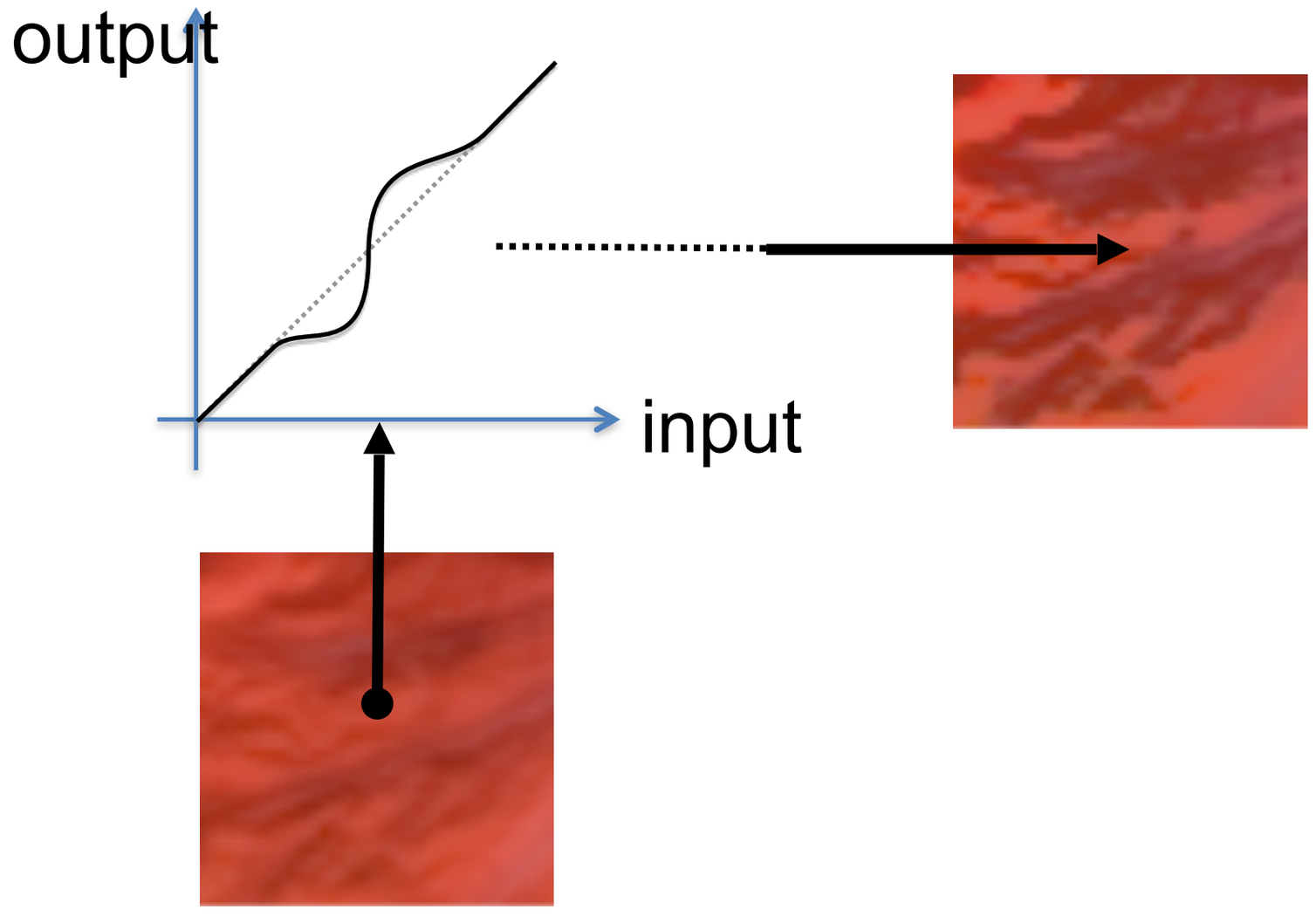
Example: Local Contrast Increase



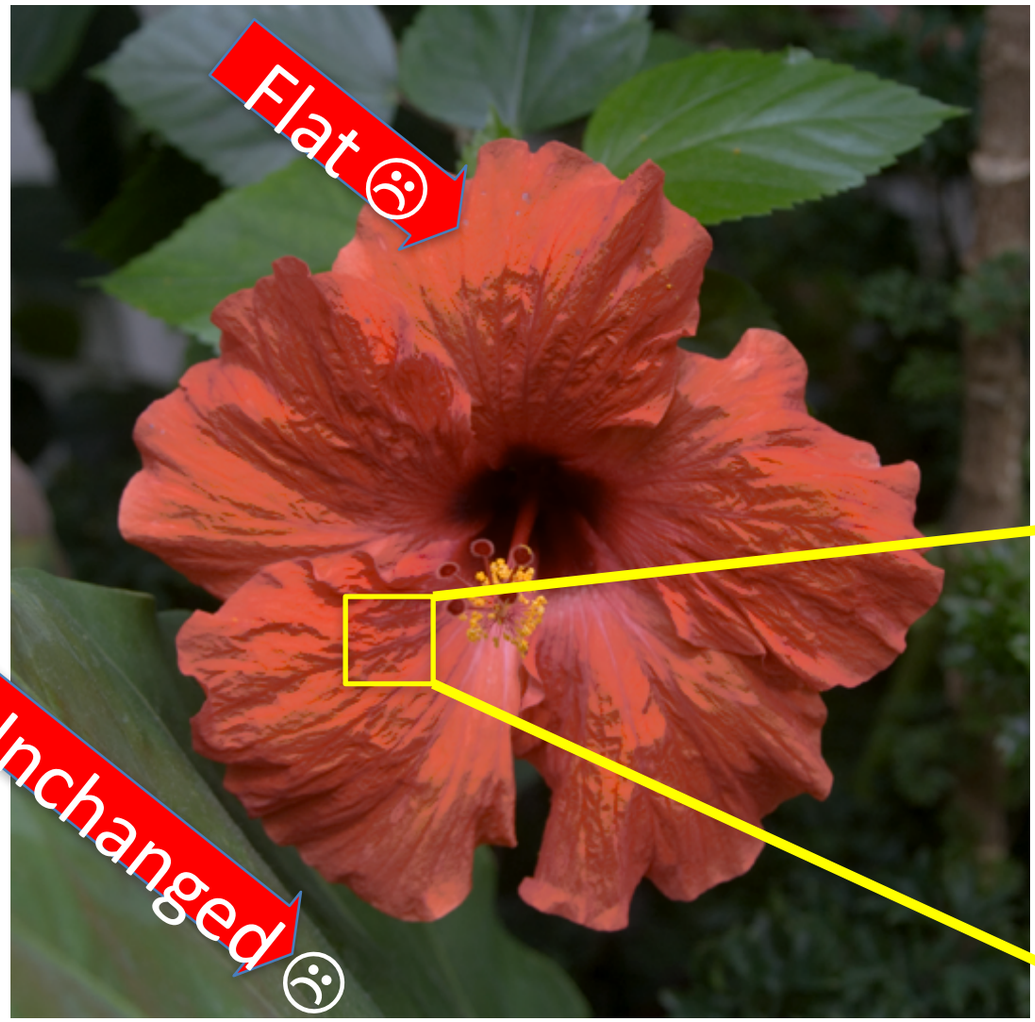
input

output with local contrast increas

Simple Local S-shaped Curve



Only Local Result Matters



output with local contrast increased

- Artifacts appear elsewhere
- Not a problem, we use only local data



The processed image only needs to look good locally

Naïve Stitching



- Paste local results side by side?
  - Visible seams, artifacts…
  - Possible heuristics: vary patch size, smooth seams…
  - But *ad hoc* and brittle



Our Multi-scale Approach

- We build the Laplacian pyramid of the output coefficient by coefficient

```
For each coefficient
  1. Generate “locally good” image
  2. Compute Laplacian pyramid of that image
  3. Copy coefficient to output pyramid
```

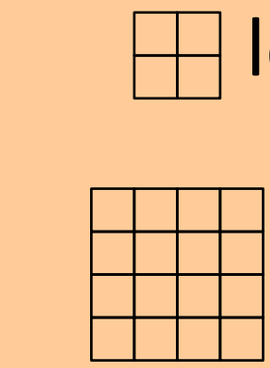
Illustration



input image

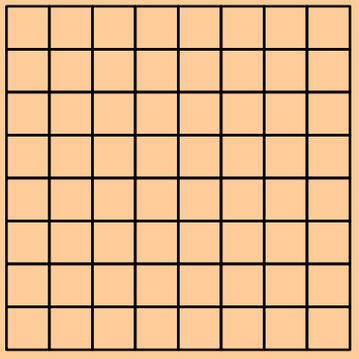
Output

Laplacian pyramid



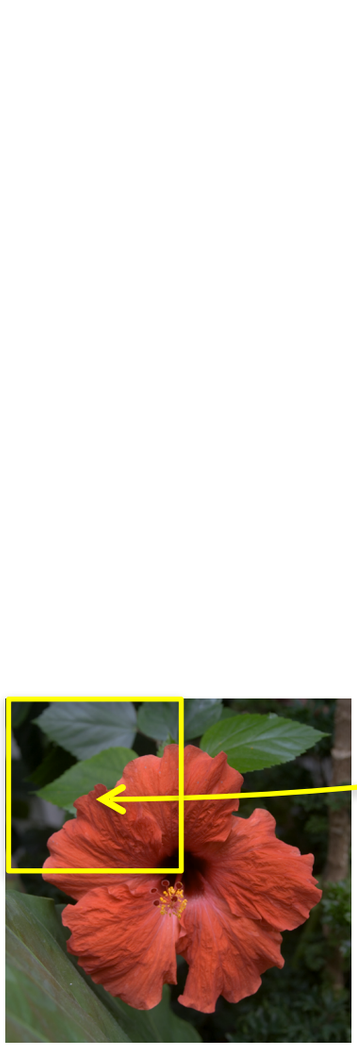
level 2

level 1



level 0

Illustration



input image

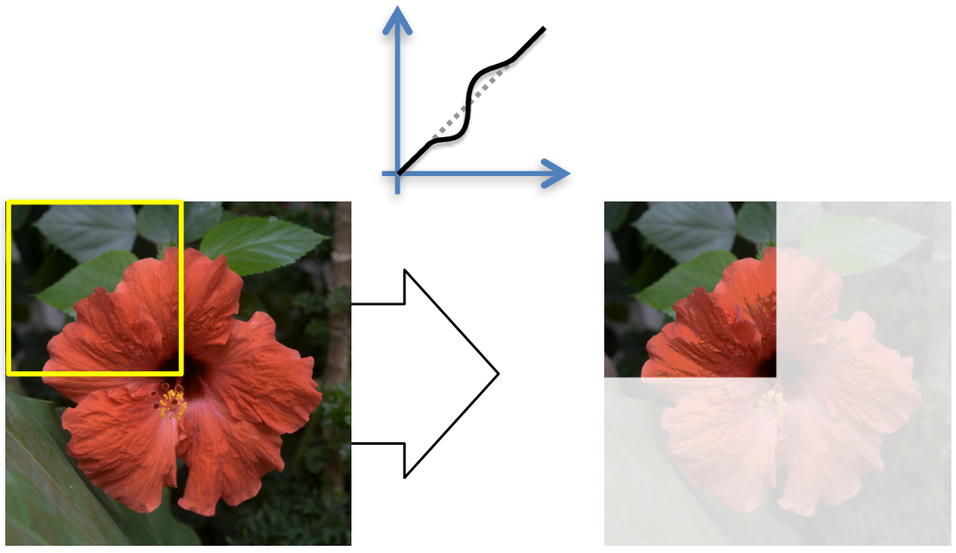
Output
Laplacian pyramid

level 2

level 1

level 0

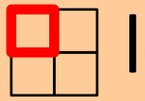
Illustration



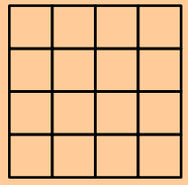
input image

“locally good” image

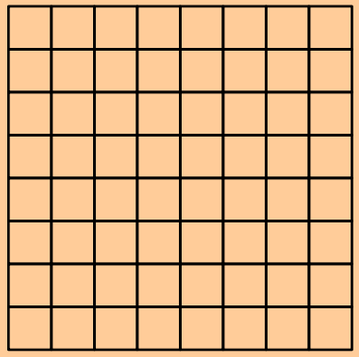
Output
Laplacian pyramid



level 2



level 1



level 0

Illustration

input image



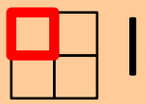
“locally good” image



partial pyramid

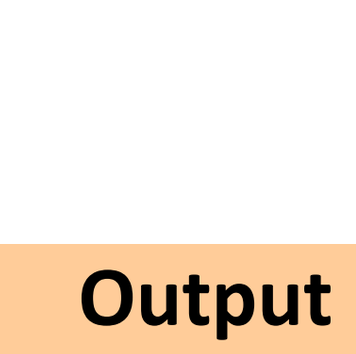
Output

Laplacian pyramid



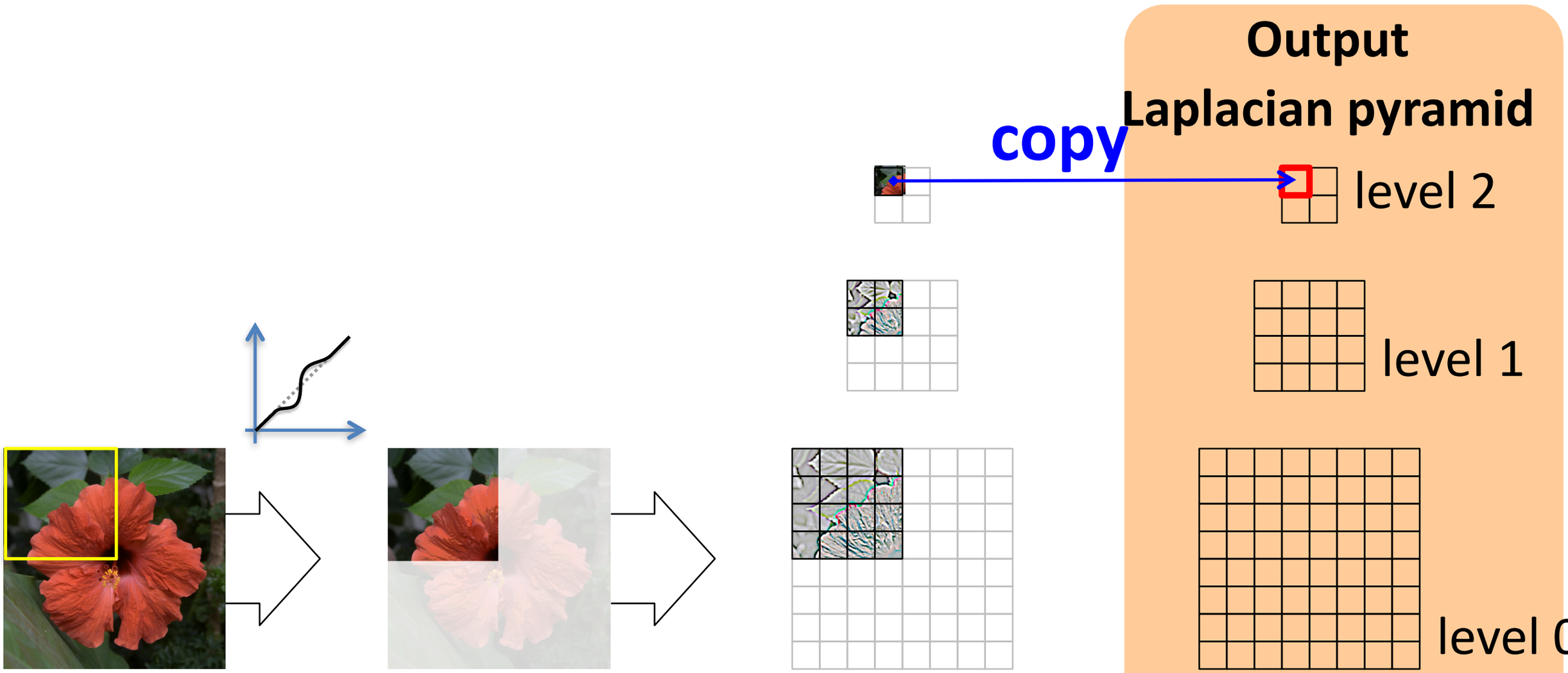
level 2

level 1



level 0

Illustration



Illustration

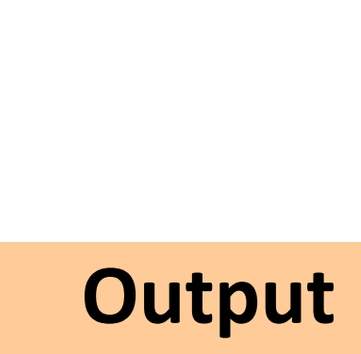
input image

Output

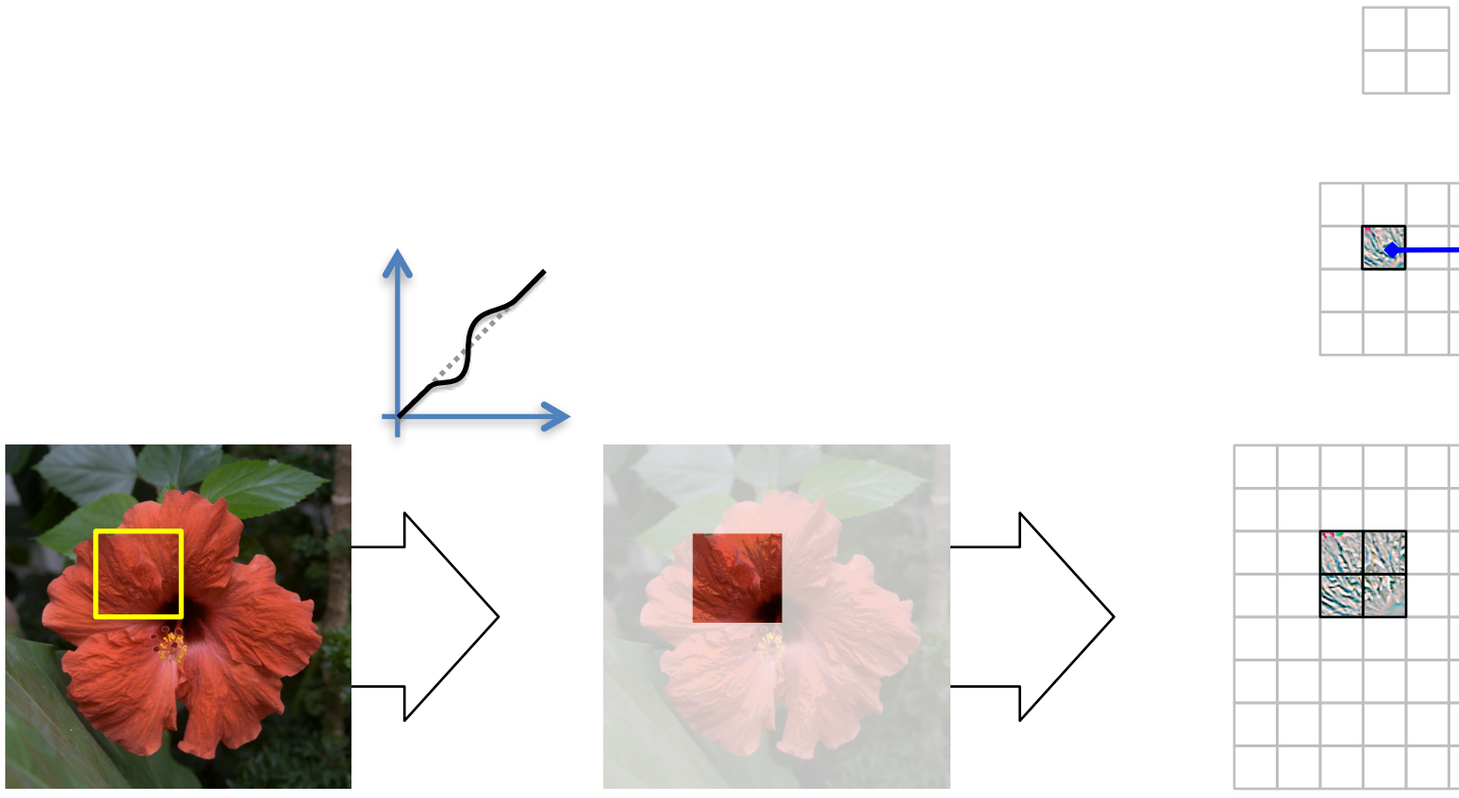
Laplacian pyramid

 level 2

 level 1

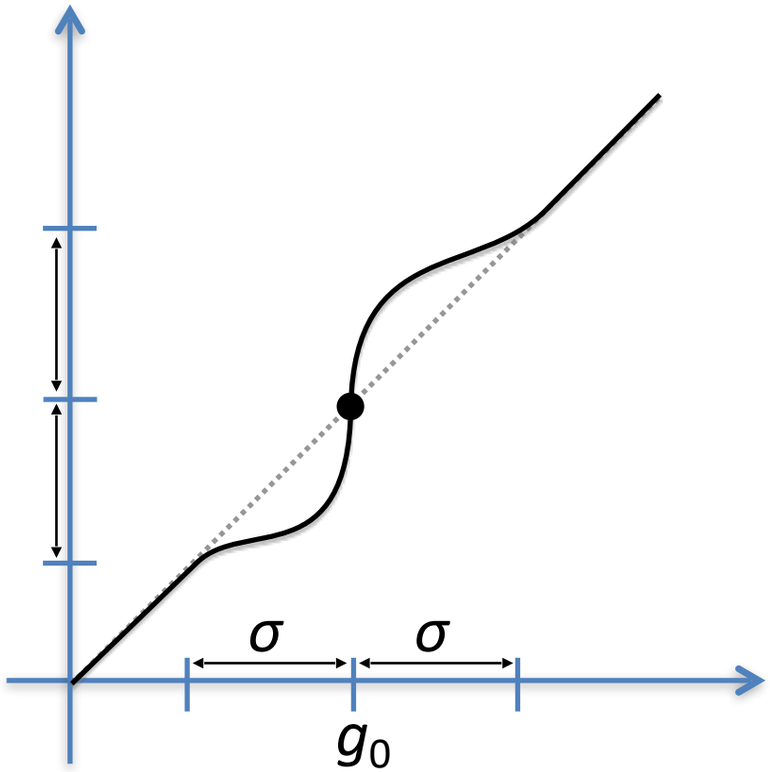
 level 0

Illustration



Possible Nonlinearities

- Detail manipulation





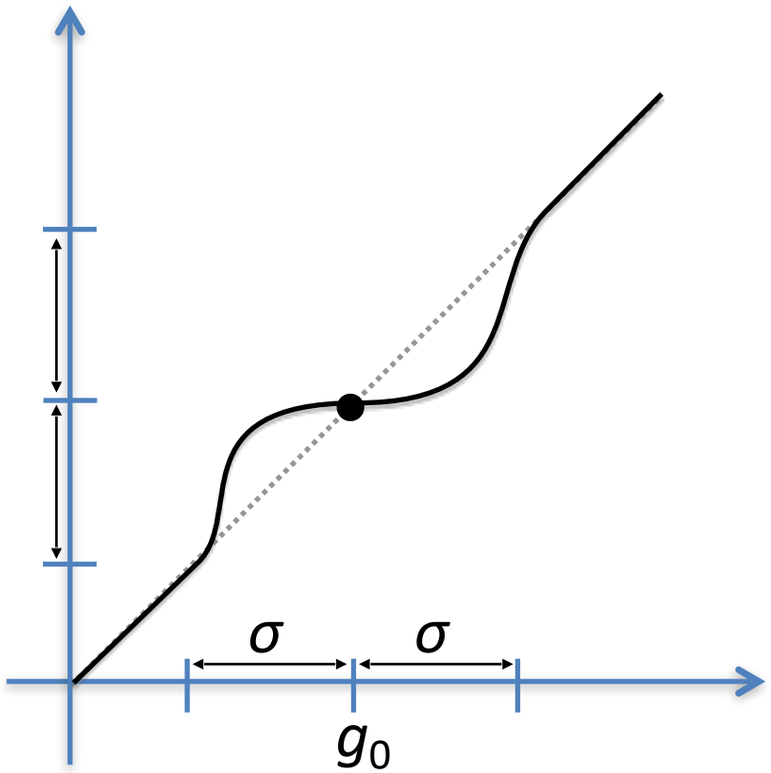
input image



details enhanced

Possible Nonlinearities

- Detail manipulation





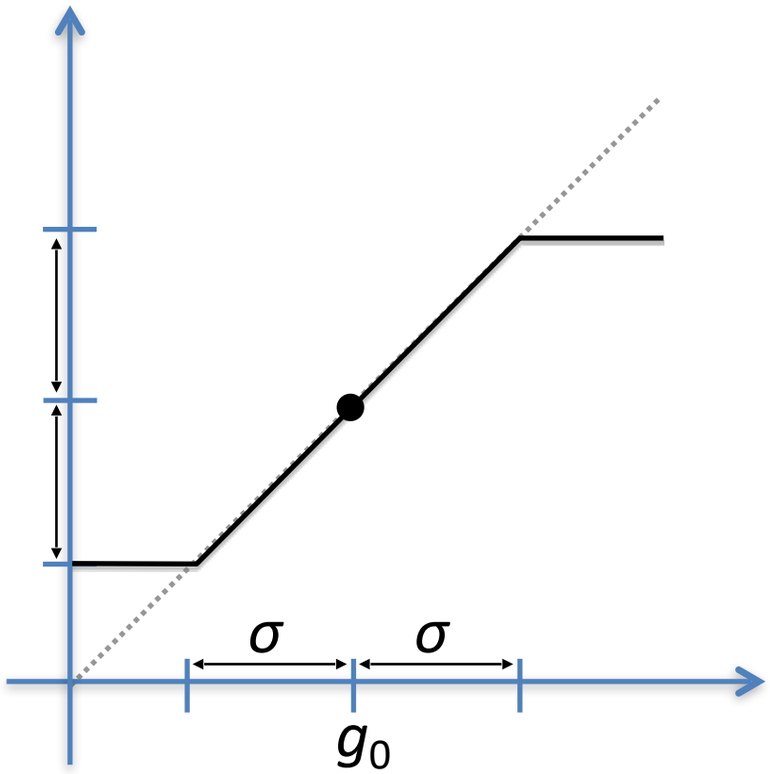
input image



details reduced

Possible Nonlinearities

- Dynamic range manipulation





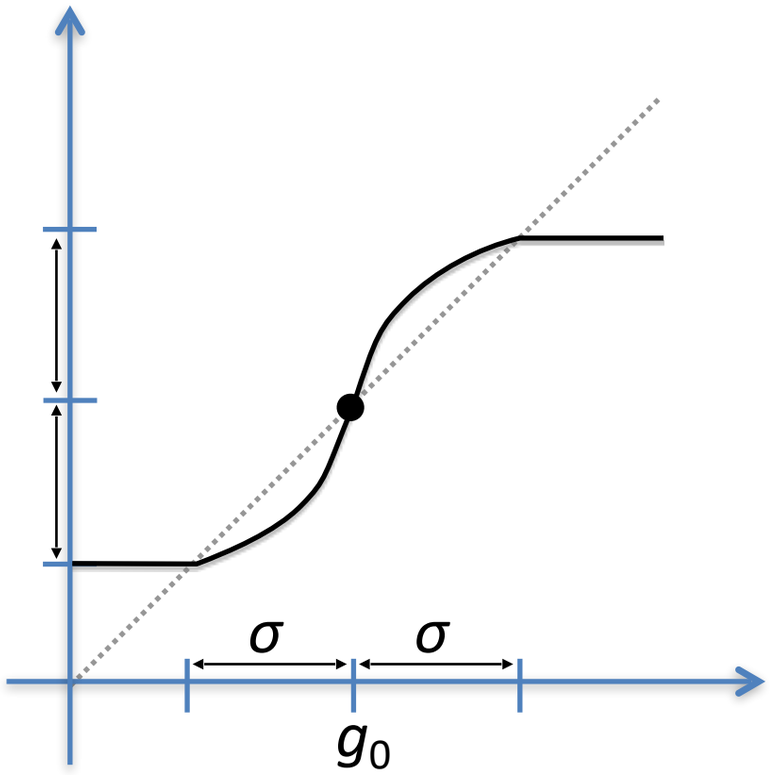
input image



tone mapped

Possible Nonlinearities

- Can be combined, e.g. tone map + boost details





input image



tone mapped + details enhanced

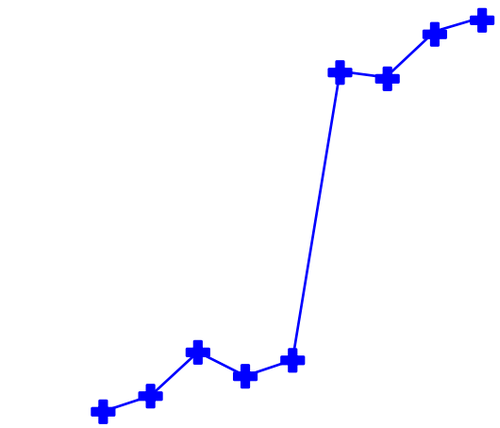
Talk Outline

- Background and related work
- Our algorithm
- Analysis
- Results

Analysis

- Laplacian pyramids built with tiny kernels
  - Each Laplacian coefficient depends only on 9×9 Gaussian coefficients at same scale
- Laplacian coefficients akin to 2nd derivatives

Intuition for 1D Edge

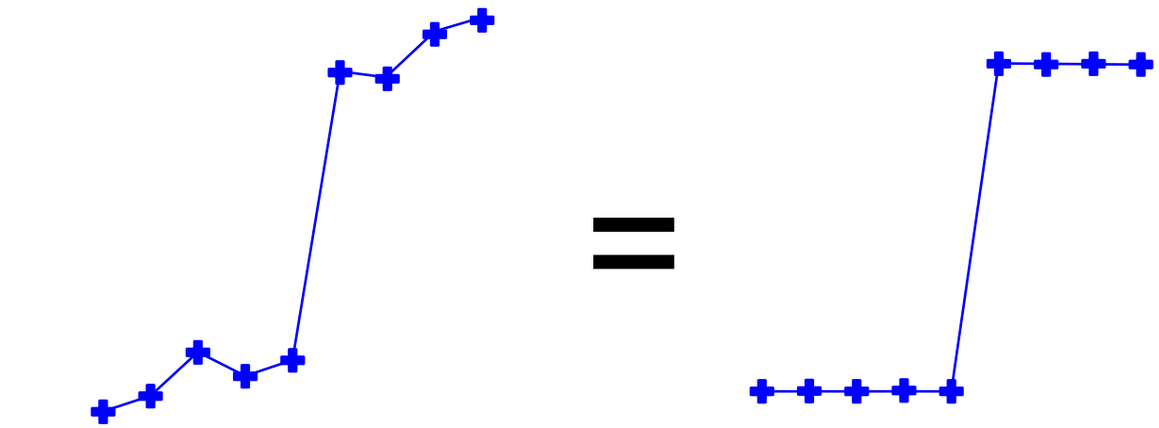


=

Input signal

- Decomposition for the sake of analysis only
  - We do not compute it in practice

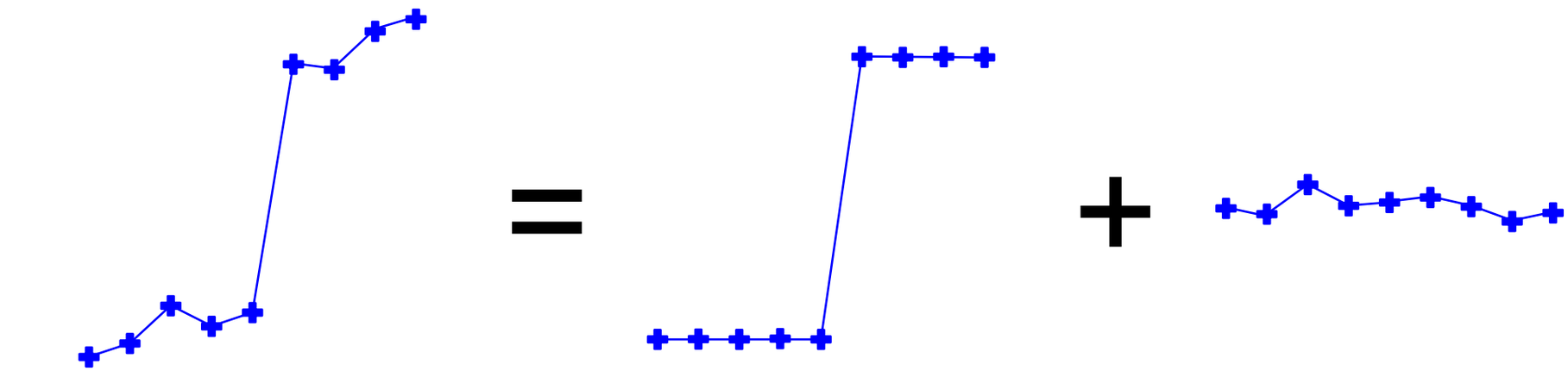
Intuition for 1D Edge



Input signalDiscontinuity

- Decomposition for the sake of analysis only
  - We do not compute it in practice

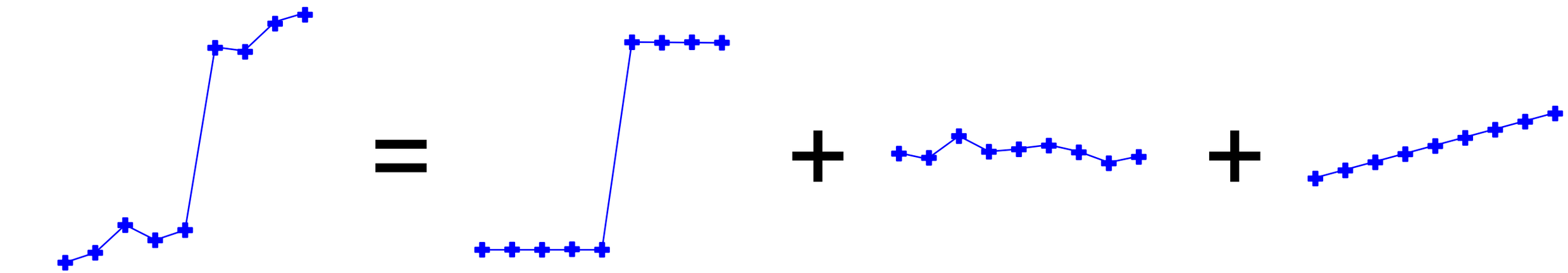
Intuition for 1D Edge



Input signal Discontinuity Texture

- Decomposition for the sake of analysis only
  - We do not compute it in practice

Intuition for 1D Edge



Input signal Discontinuity Texture Smooth

- Decomposition for the sake of analysis only
  - We do not compute it in practice

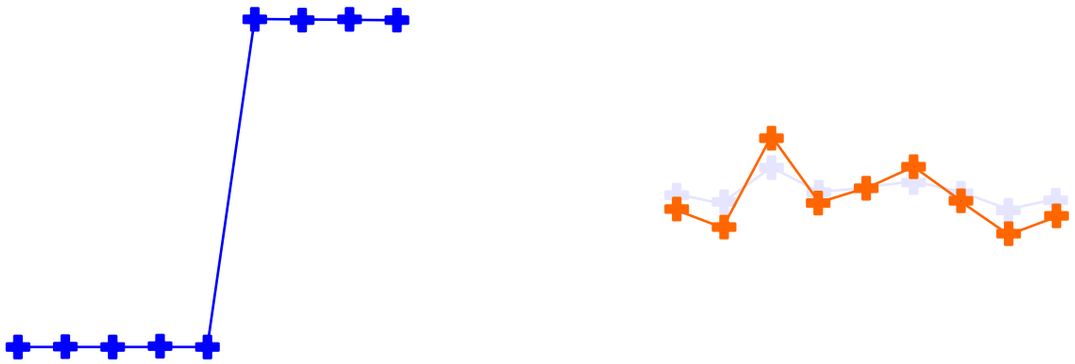
Intuition for 1D Edge



Smooth

Does not contribute to Lap. pyramid at that scale ($d^2/dx^2=0$)

Ideal Texture Increase

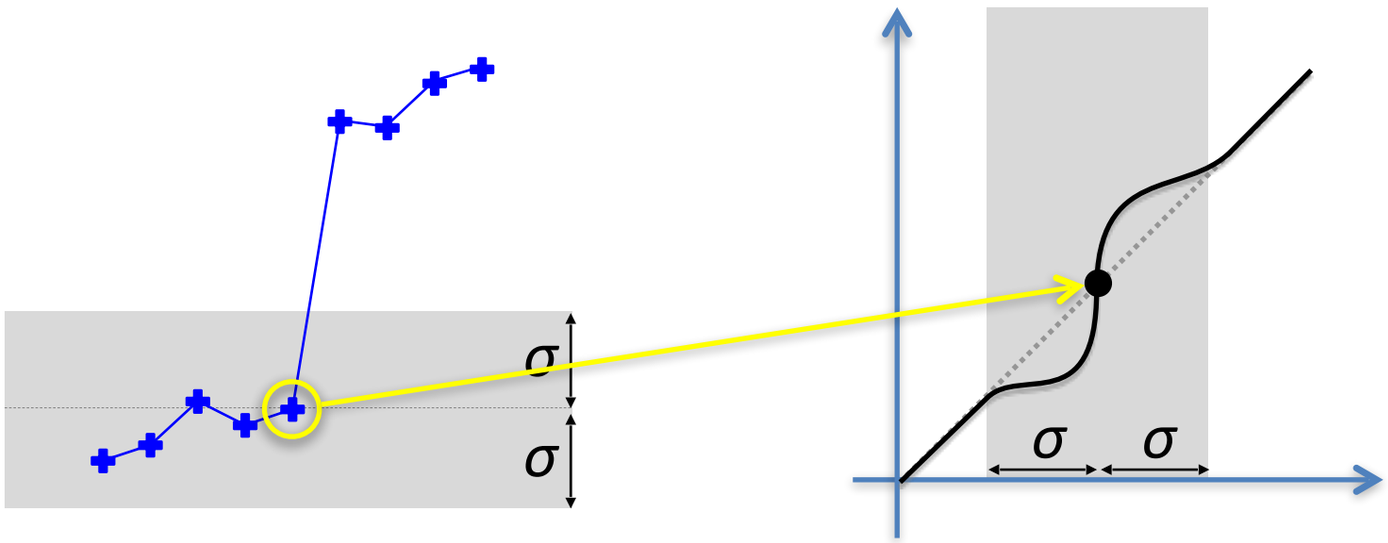


Discontinuity **Texture**

Keep unchanged

Amplify

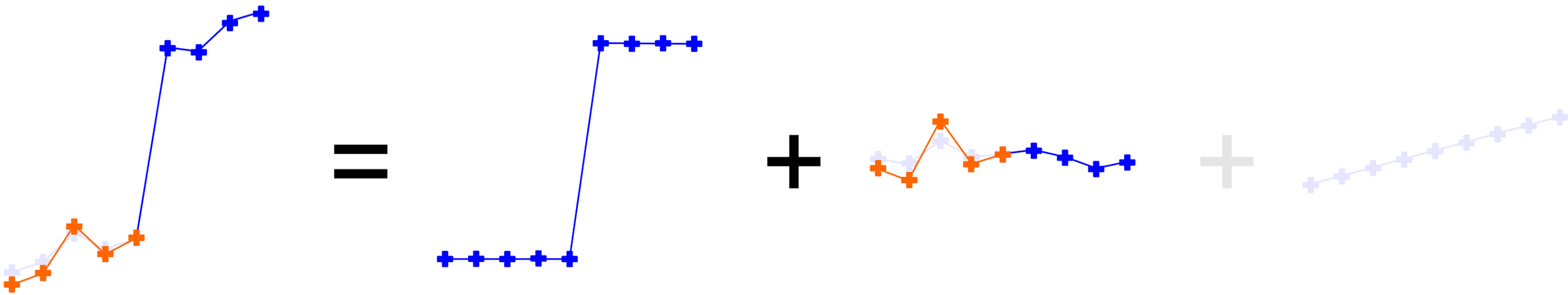
Our Texture Increase



Input signal **Local nonlinearity** **“Locally good” version**

user-defined parameter σ defines texture vs. edges

Our Texture Increase



Locally good
Only left side is affected

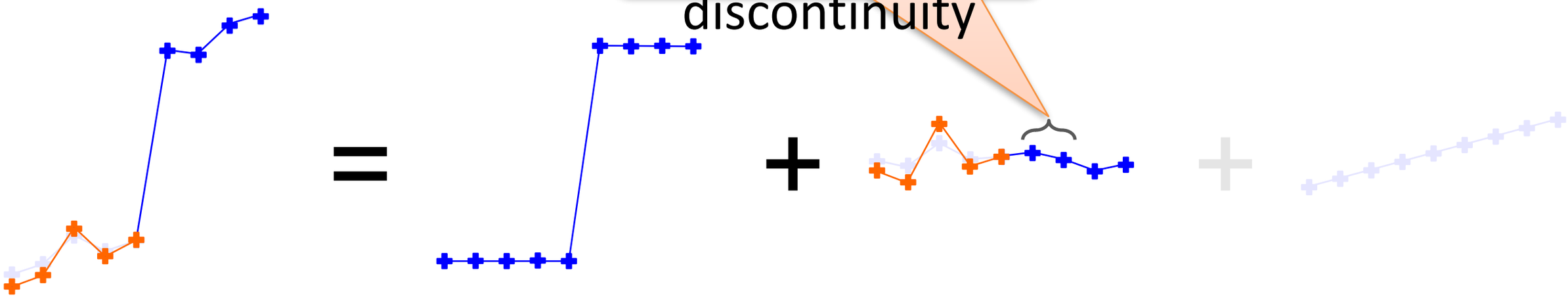
Discontinuity
Unaffected
☺

Texture
Left side is ok, right side is not
😐

Smooth
Does not contribute to Lap. pyramid at that scale
$(d^2/dx^2=0)$

Discussion

Negligible because collocated with discontinuity



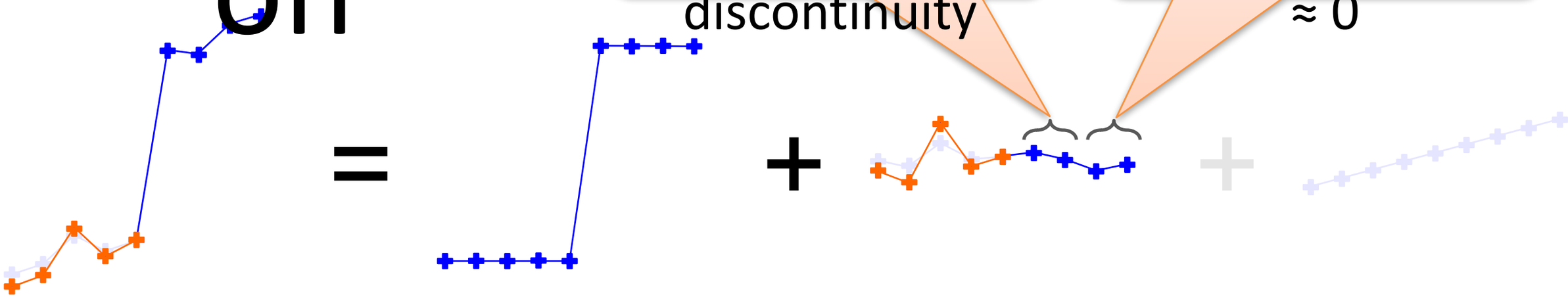
Locally good
Only left side is affected

Discontinuity
Unaffected
☺

Texture
Left side is ok, right side is not
😐

Smooth
Does not contribute to Lap. pyramid at that scale
($d^2/dx^2=0$)

Discussion



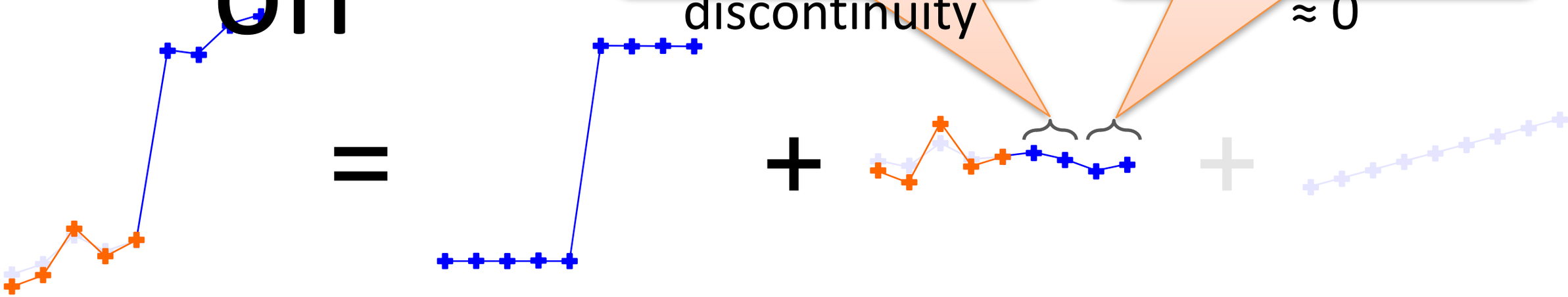
Locally good
Only left side is affected

Discontinuity
Unaffected
☺

Texture
Left side is ok, right side is not
😐

Smooth
Does not contribute to Lap. pyramid at that scale
($d^2/dx^2=0$)

Discussion



Locally good
Only left side is affected

Discontinuity
Unaffected
☺

Texture
Left side is ok, right side is not
😐

Smooth
Does not contribute to Lap. pyramid at that scale
($d^2/dx^2=0$)

Good approximation to ideal case overall ☺
(formal treatment in paper)

Talk Outline

- Background and related work
- Our algorithm
- Analysis
- Results

Running Times

- Main shortcoming of the approach: it is slow
  - 1 megapixel in 1 minute
  - Can be sped up with heuristic (see paper)
- Highly data parallel
  - 8 cores: 1 megapixel in 4 seconds

Texture Manipulation
Input



Texture Manipulation
Decrease




Texture Manipulation
Small Increase

Texture Manipulation
Large Increase



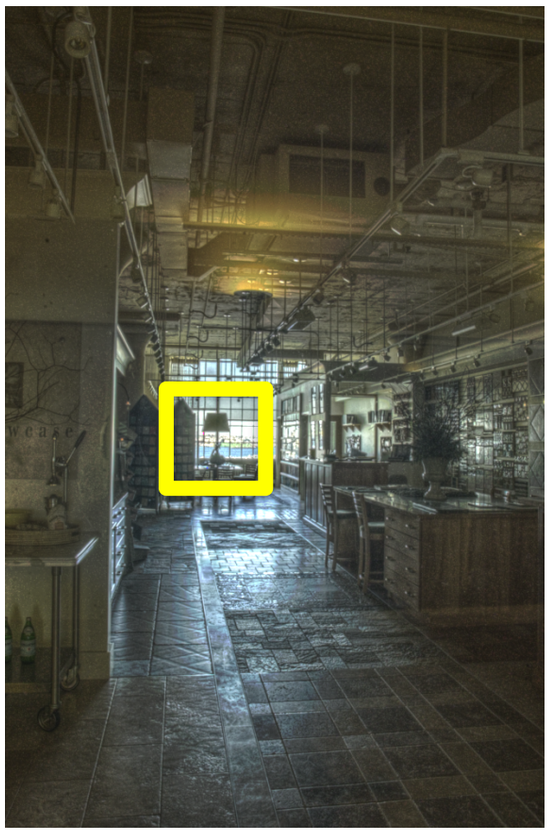
Texture Manipulation
Input



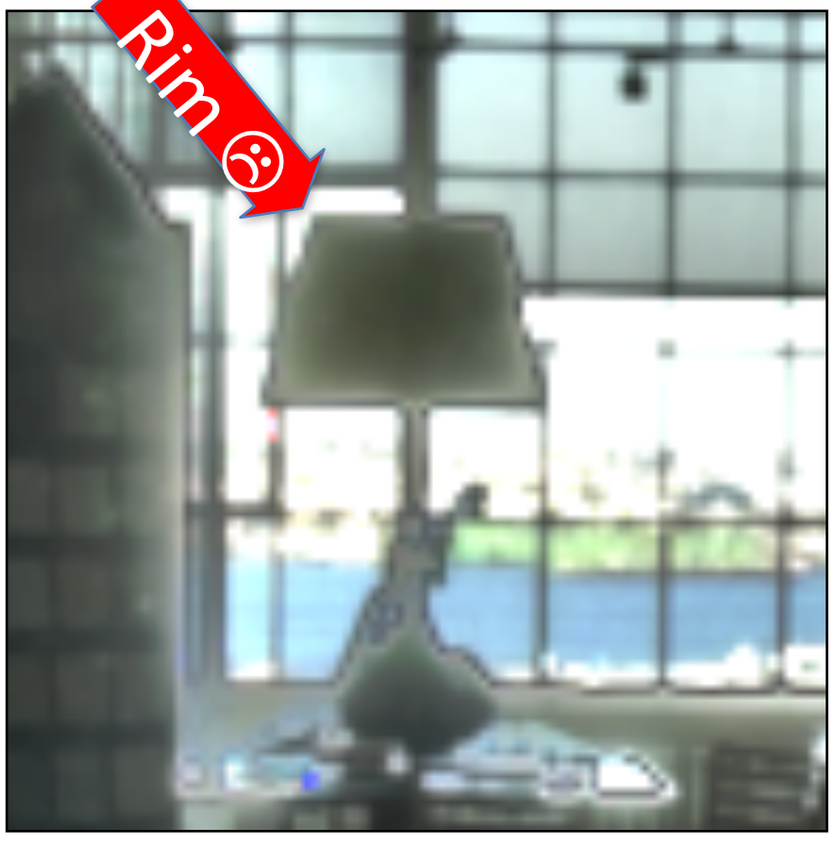
Texture Manipulation
Large Increase



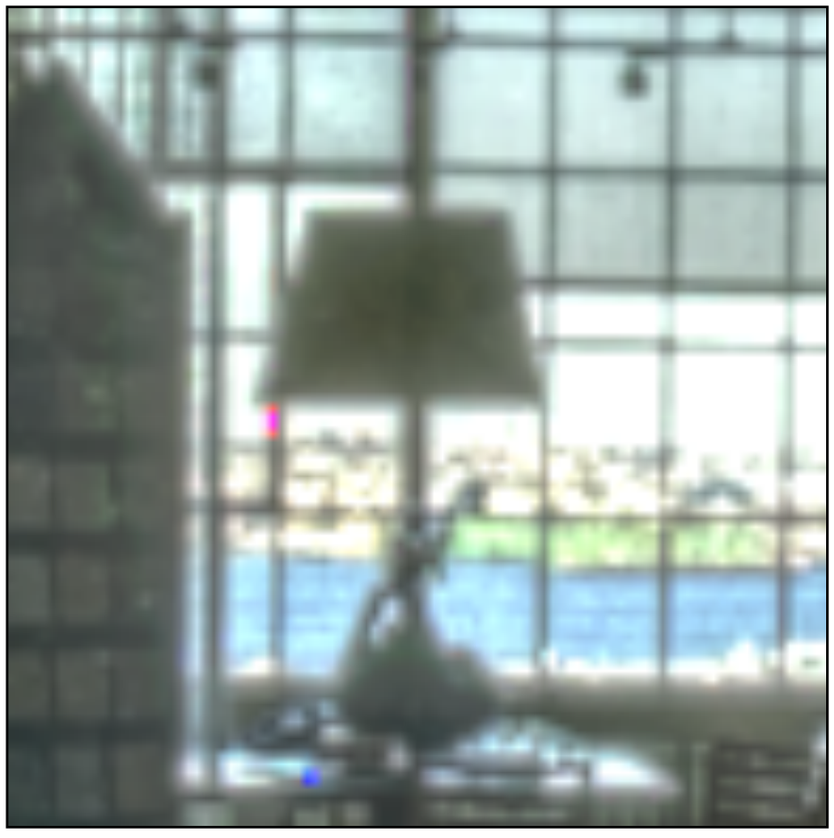
Compared to Bilateral Filter



tone mapped HDR image

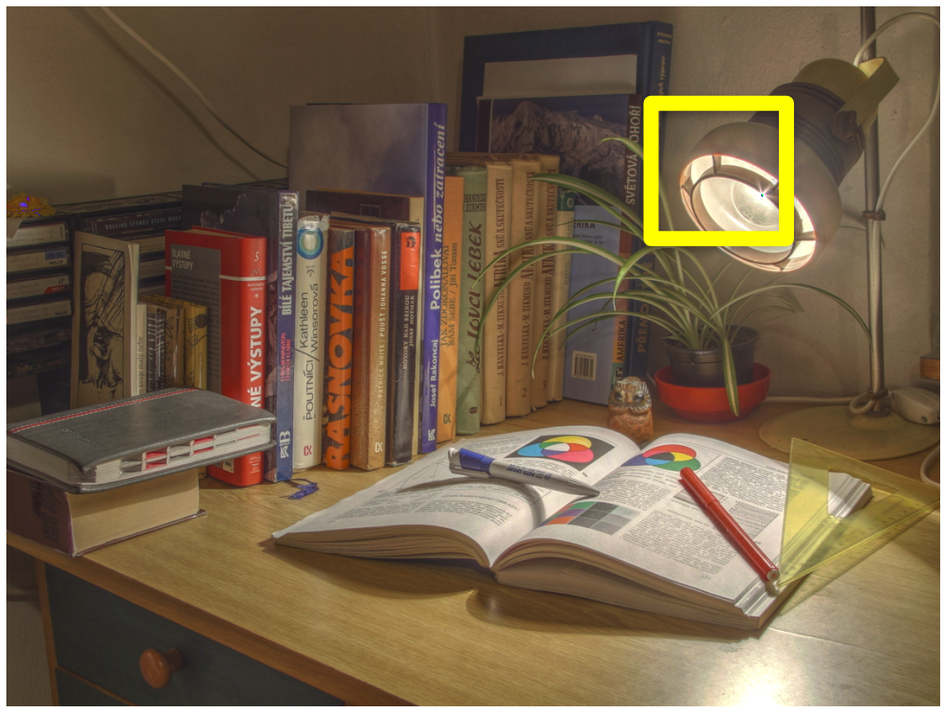


close-up (bilateral filter)



close-up (our result)

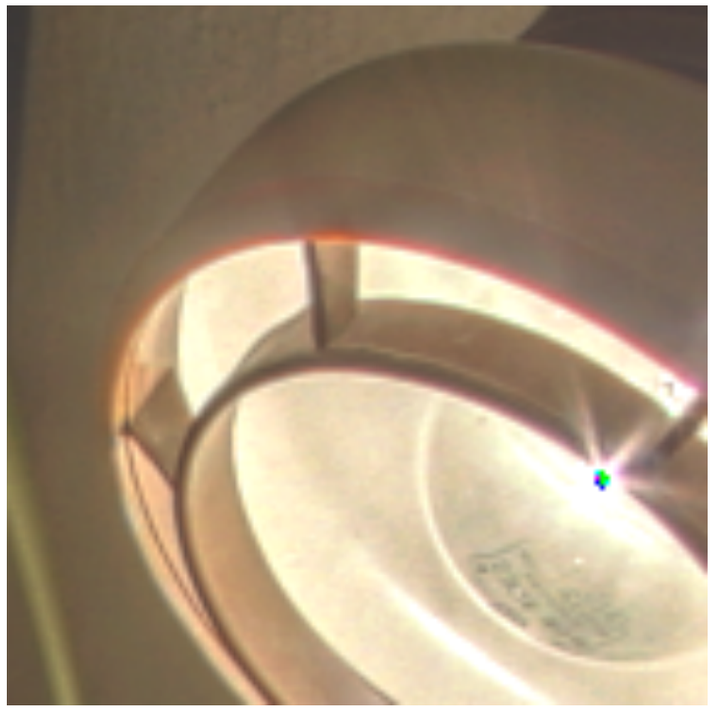
Compared to Edge-avoiding Wavelets



tone mapped HDR image



close-up (edge-avoiding wavelets) reproduced from [Fattal 09]



close-up (our result)

Naïve gamma compression

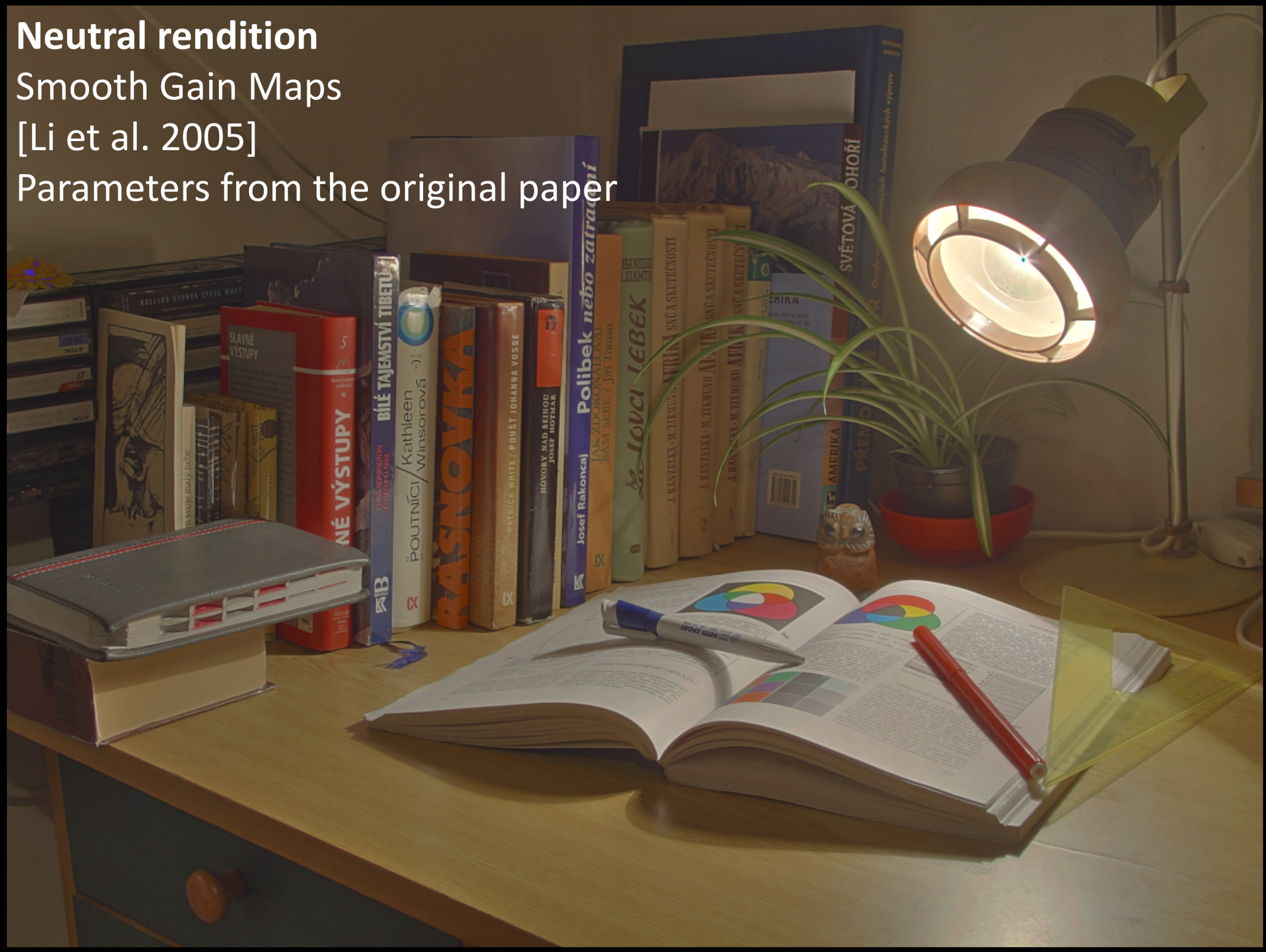


Neutral rendition

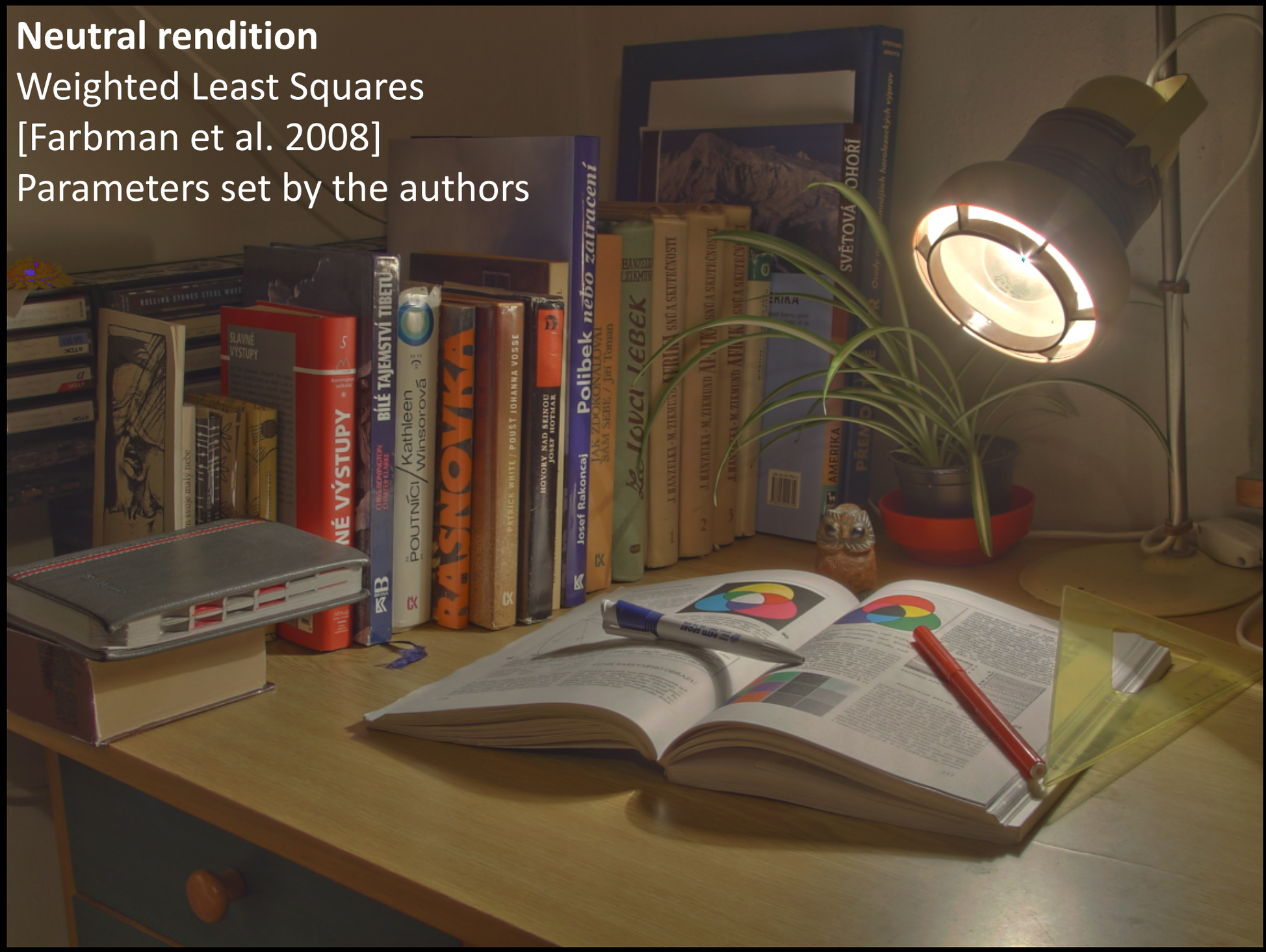
Smooth Gain Maps

[Li et al. 2005]

Parameters from the original paper

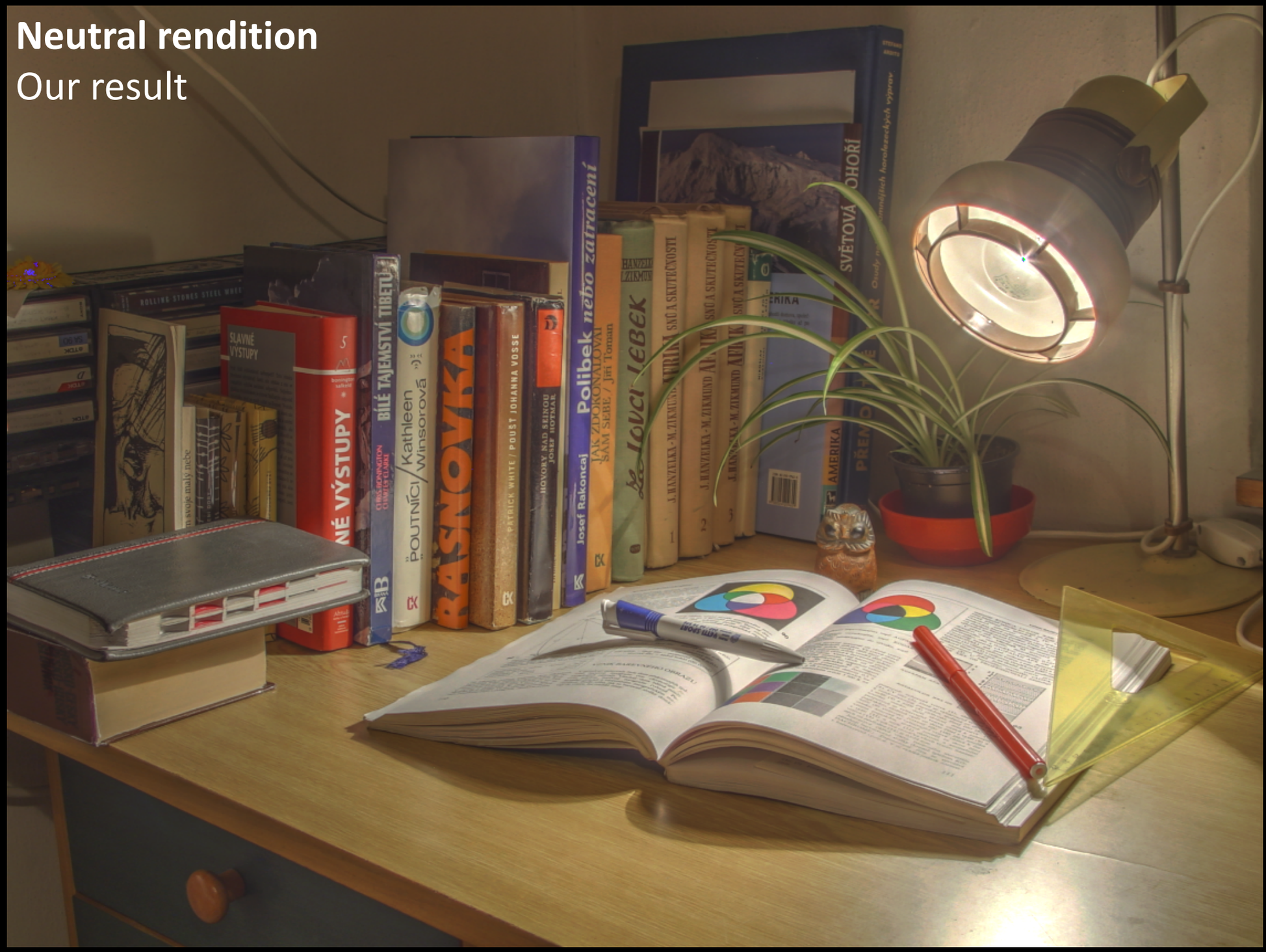


Neutral rendition
Weighted Least Squares
[Farbman et al. 2008]
Parameters set by the authors



Neutral rendition

Our result



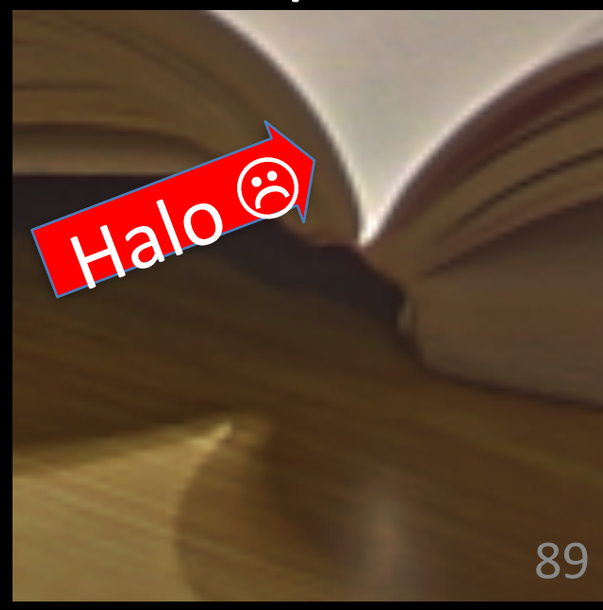
Exaggerated rendition
Smooth Gain Maps
[Li et al. 2005]
Parameters from the original paper

Limited details ☹

close-up



close-up



Exaggerated rendition
Weighted Least Squares
[Farbman et al. 2008]
Parameters set by the authors

close-up

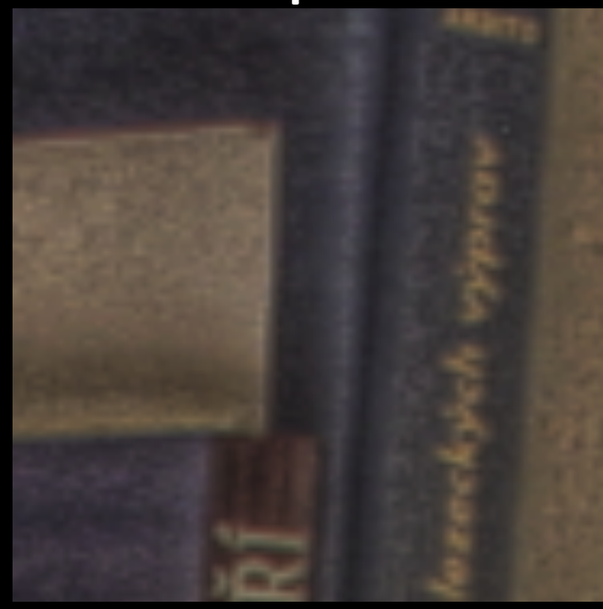


close-up

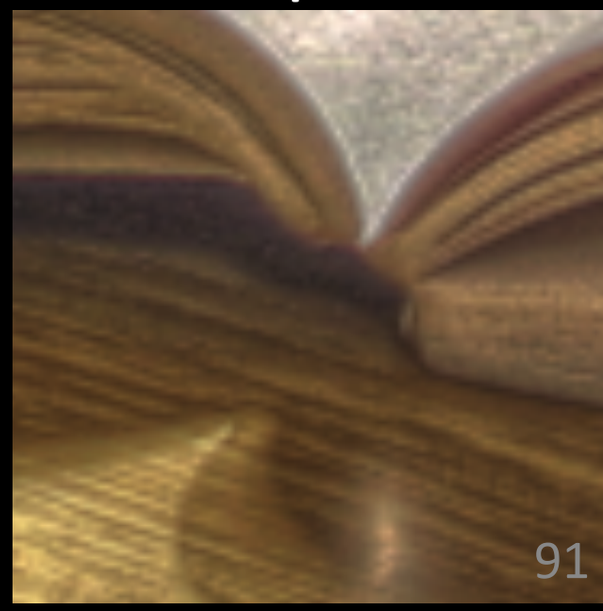


Exaggerated rendition
Our result

close-up



close-up

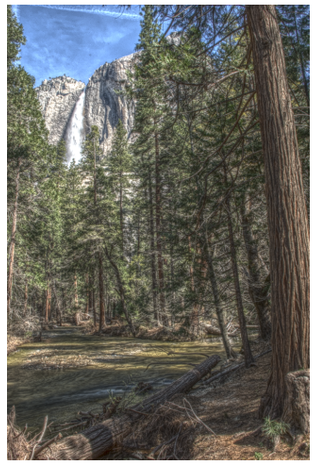


Summary of Comparisons

- All methods do well on neutral renditions
- Previous work exhibits halos for large increases in detail
- Our approach produces clean results even on hard cases: no aliasing, no rims, no halos

In Supplemental Material

- Extensive validation and comparison
  - No hand tuning: all results generated by a script




- Matlab source code to reproduce the results

Conclusions

- Laplacian pyramids good for edge-aware editing
  - Classical notion of multi-scale
  - Standard image resizing routines
- Enables wider range of effect



We would like to thank…

- **Ted Adelson**, **Bill Freeman**, **Frédo Durand**, and the **MIT Graphics Group** for their insightful discussion and feedback
- The **people who make their data available** on the web
- The **authors of [Li et al. 05] and [Farbman et al. 08]** who helped us compare to their work
- The **anonymous reviewers** for their constructive comments
- This work was supported in part by an **NSERC** Postdoctoral Fellowship, the **Quanta** T-Party, **NGA** NEGI-1582-04-0004, **MURI** Grant N00014-06-1- 0734, and gifts from **Microsoft**, **Google**, and **Adobe**

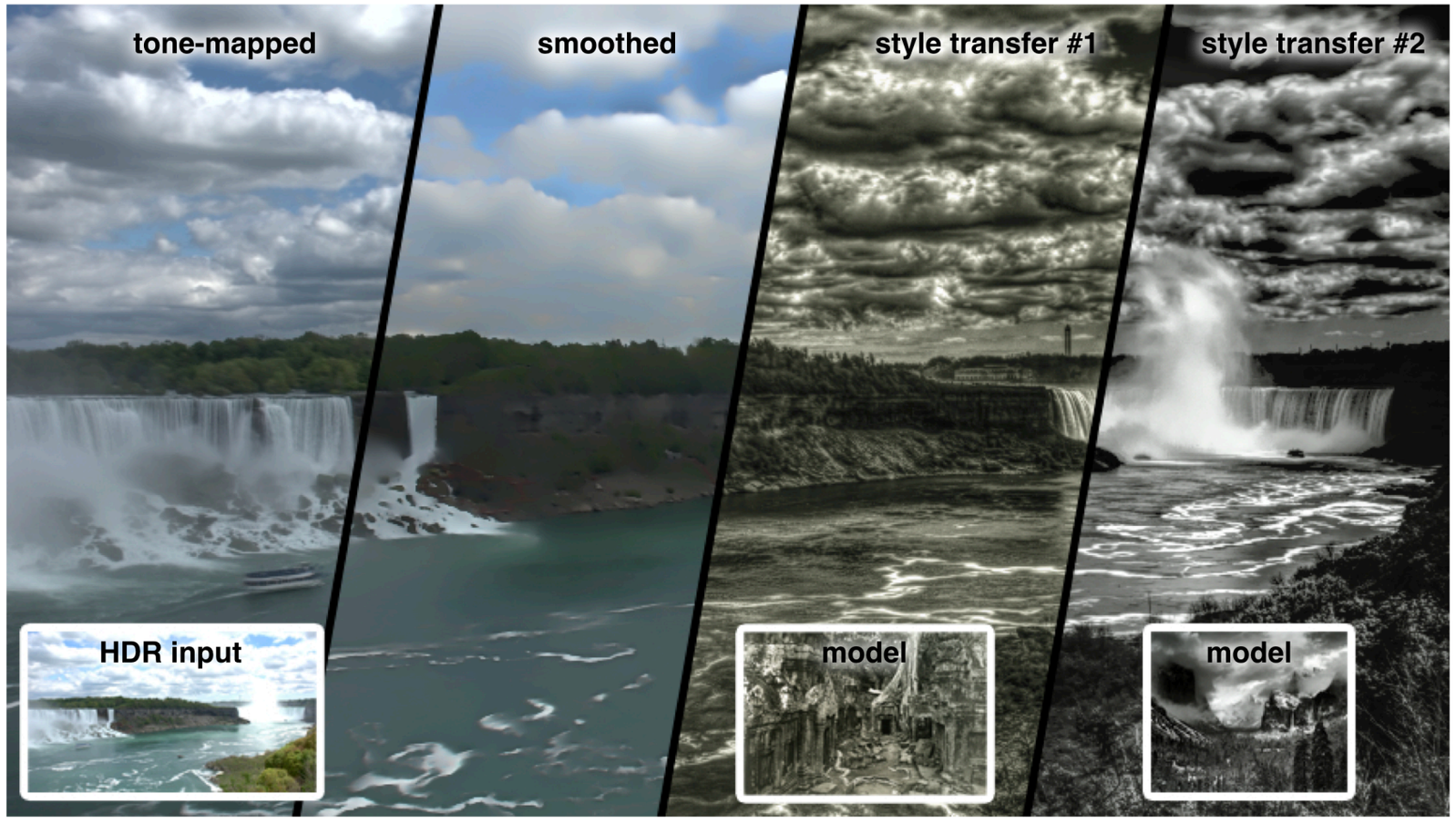
Conclusions

- Laplacian pyramids good for edge-aware editing
  - Classical notion of multi-scale
  - Standard image resizing routines
- Enables wider range of effect



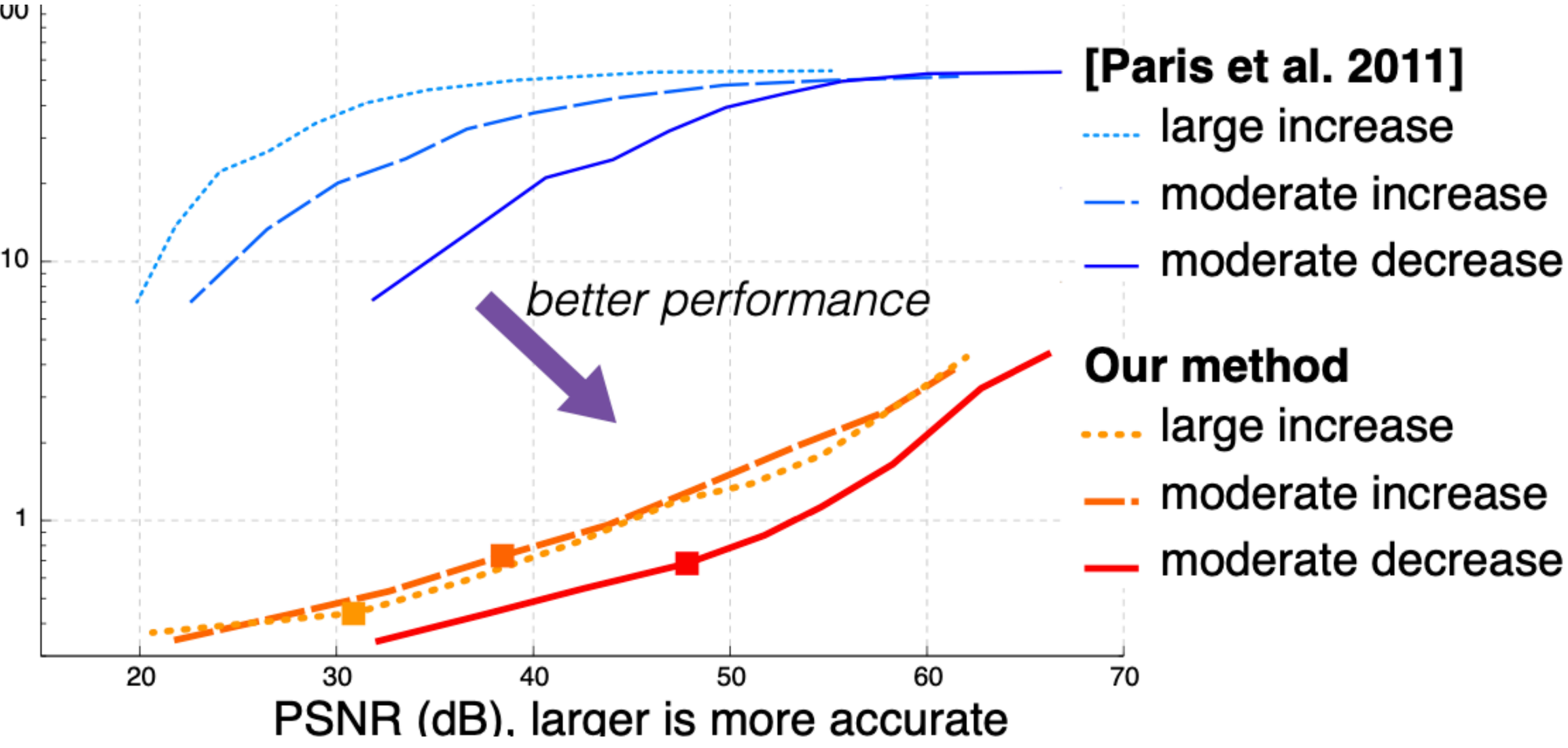
Next step : fast version

- Aubry et al.
- Fredo’s favorite trick: discretize pixel values
- https://people.csail.mit.edu/sparis/publi/2014/tog/Aubry_14-Fast_Local_Laplacian_Filters.pdf



Fast and accurate

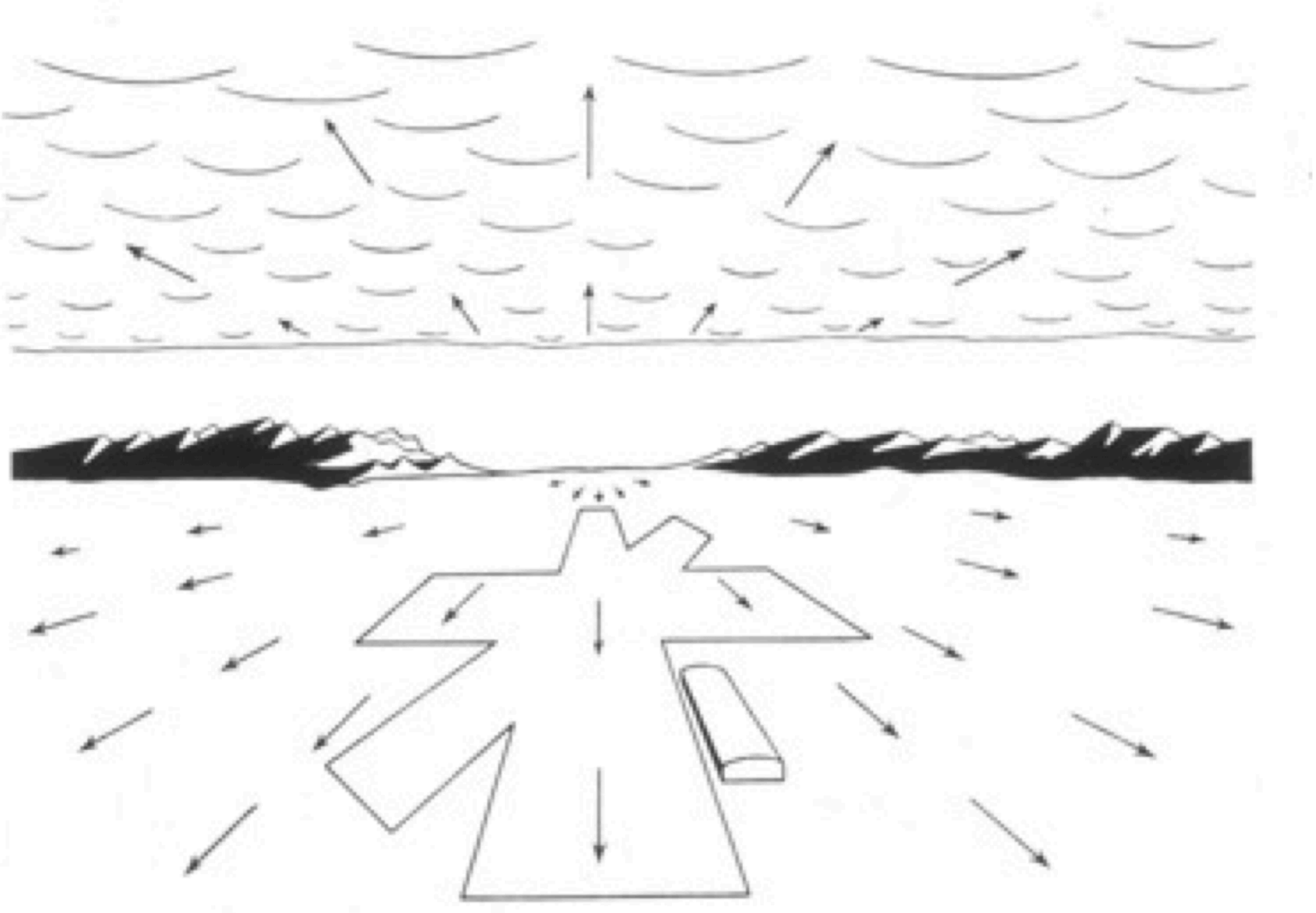
- That is what is used at Adobe



Optical flow

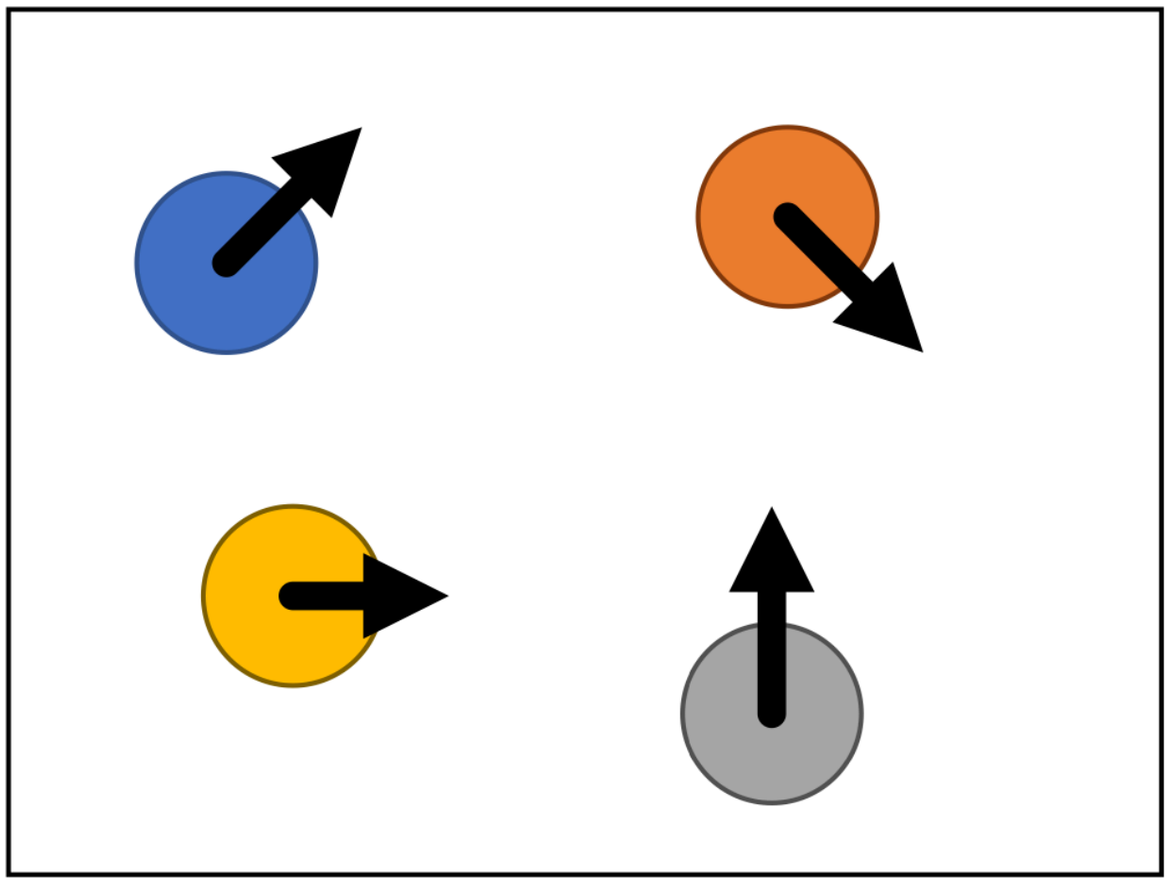
Optical flow

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

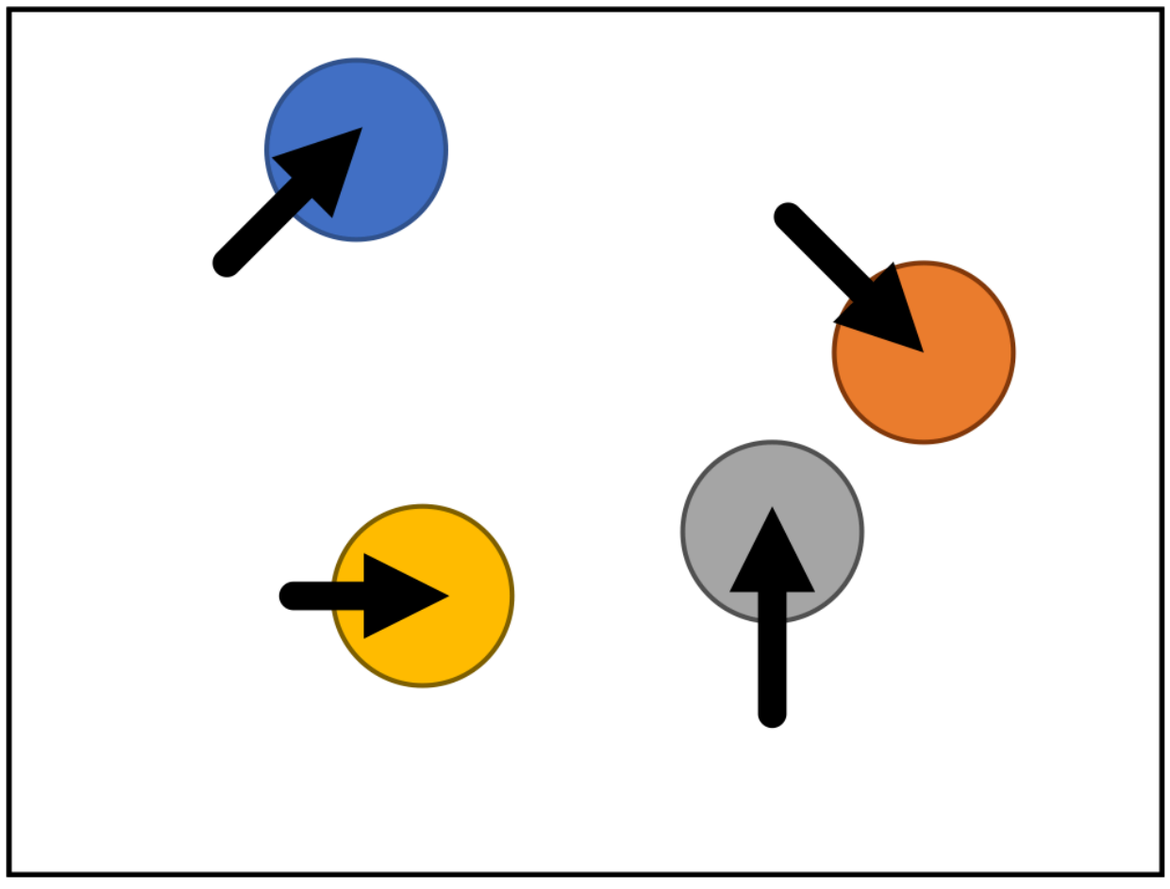


Problem statement

- Estimate pixel motion from image at time t to t+1



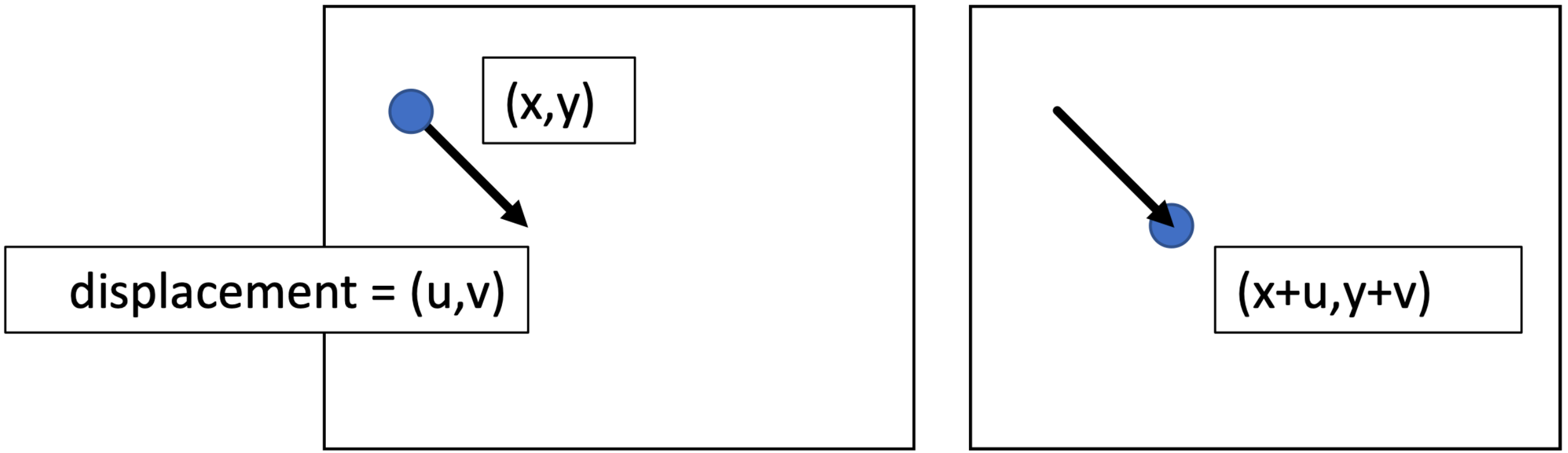
I(x,y,t)



I(x,y,t+1)

https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture19.pdf

Brightness constancy



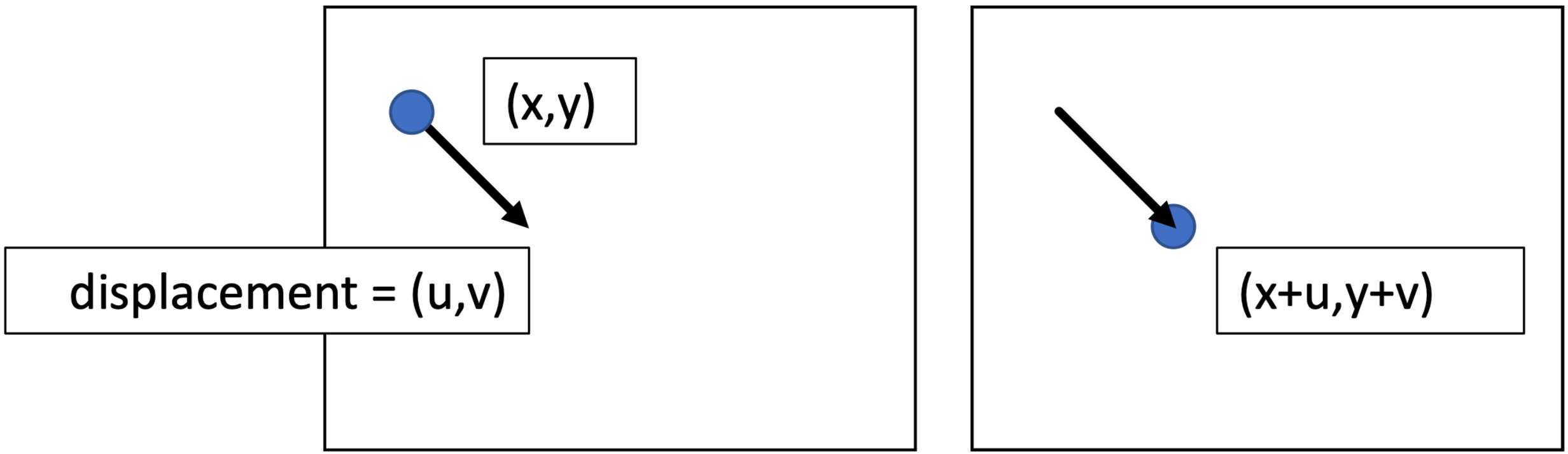
Brightness constancy:

$$I(x, y, t) = I(x + u, y + v, t + 1)$$

Recall Taylor Expansion:

$$I(x + u, y + v, t) = I(x, y, t) + I_x u + I_y v + \cdots$$

Brightness constancy assumption



Brightness constancy:

$$I(x, y, t) = I(x + u, y + v, t + 1)$$

$$I(x+u, y+v, t+1) = I(x, y, t)$$

$$0 \approx I(x+u, y+v, t+1) - I(x, y, t)$$

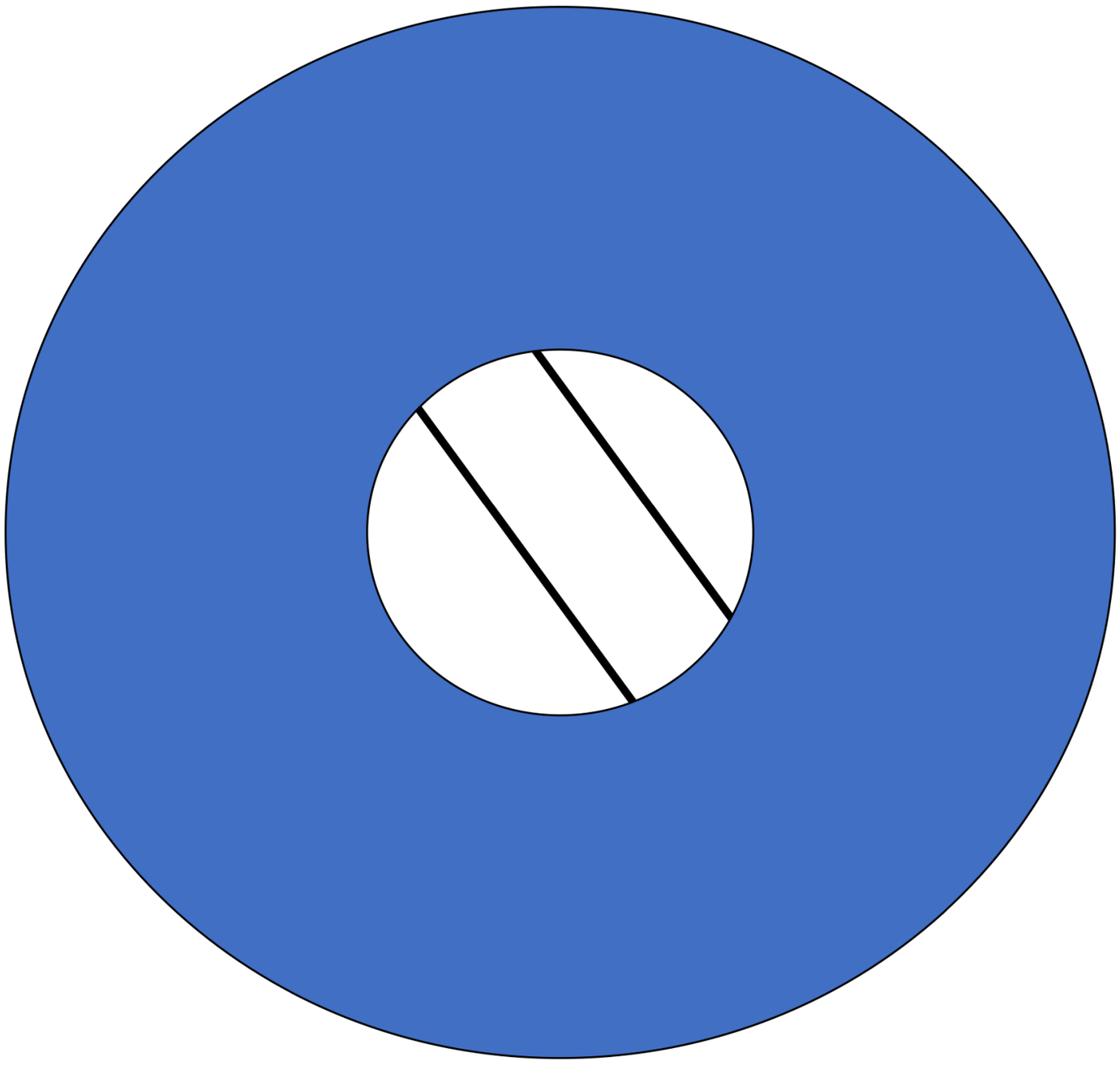
$$I(x+u, y+v, t+1) = I(x, y, t)$$

$$\begin{aligned} 0 &\approx I(x+u, y+v, t+1) - I(x, y, t) \\ &= I(x, y, t+1) + I_x u + I_y v - I(x, y, t) \\ &= I(x, y, t+1) - I(x, y, t) + I_x u + I_y v \\ &= I_t + I_x u + I_y v \\ &= I_t + \nabla I \cdot [u, v] \end{aligned}$$

Taylor Expansion

1D example

Aperture problem



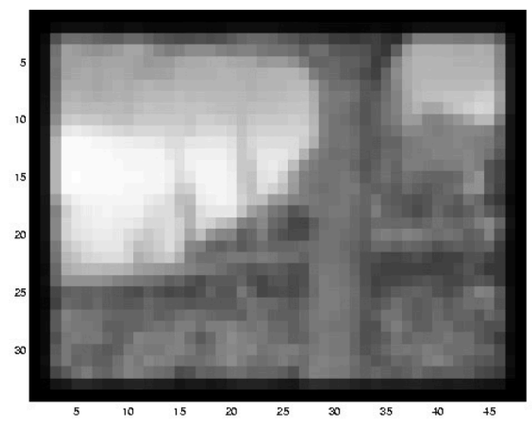
bnik

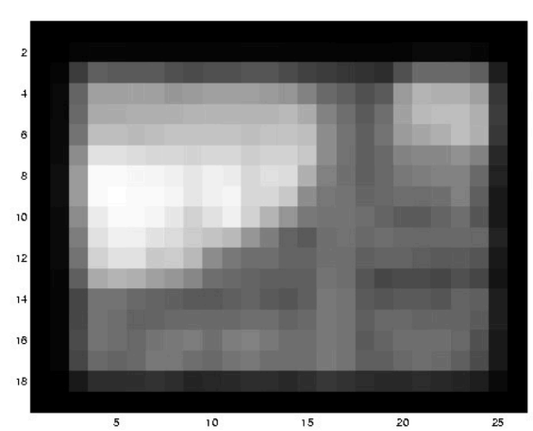
Coarse to fine

- Motivation: Taylor expansion valid only for small motion
- Reduce image resolution so that motion is small

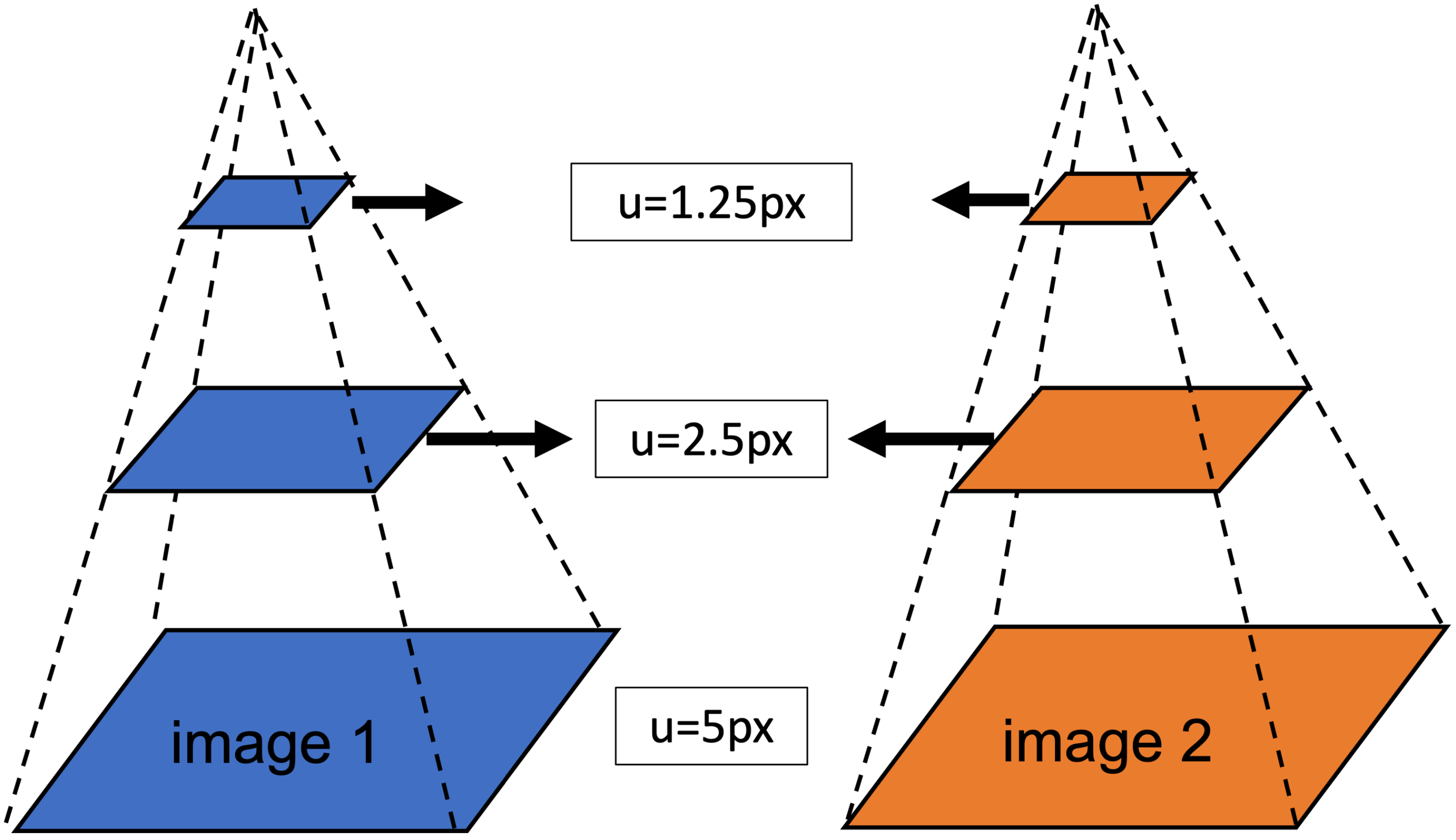








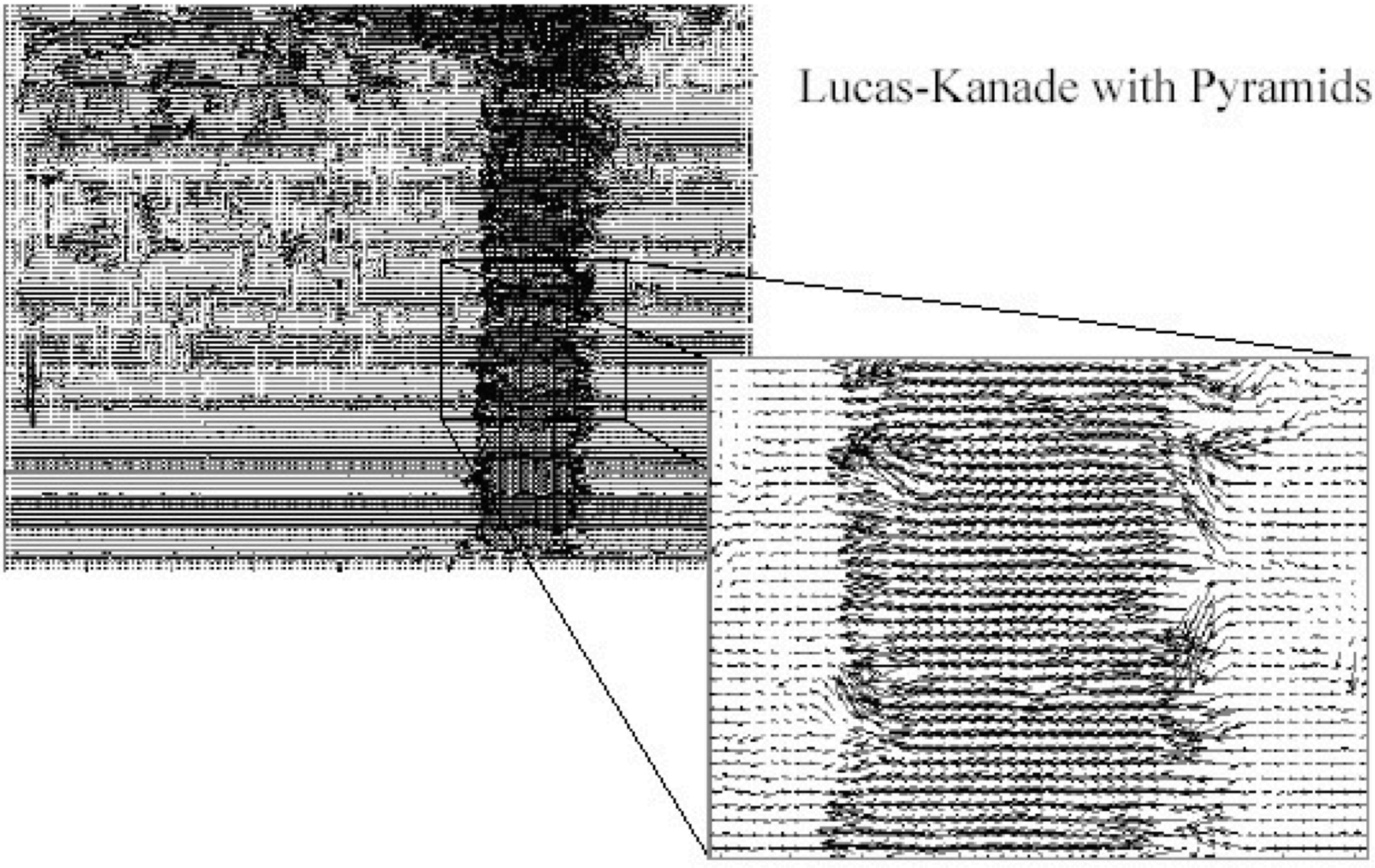
Coarse-to-fine optical flow estimation



Slide credit: S. Lazebnik

Typically called Gaussian Pyramid

Optical flow results



credit: S. Lazebnik

Video Magnification



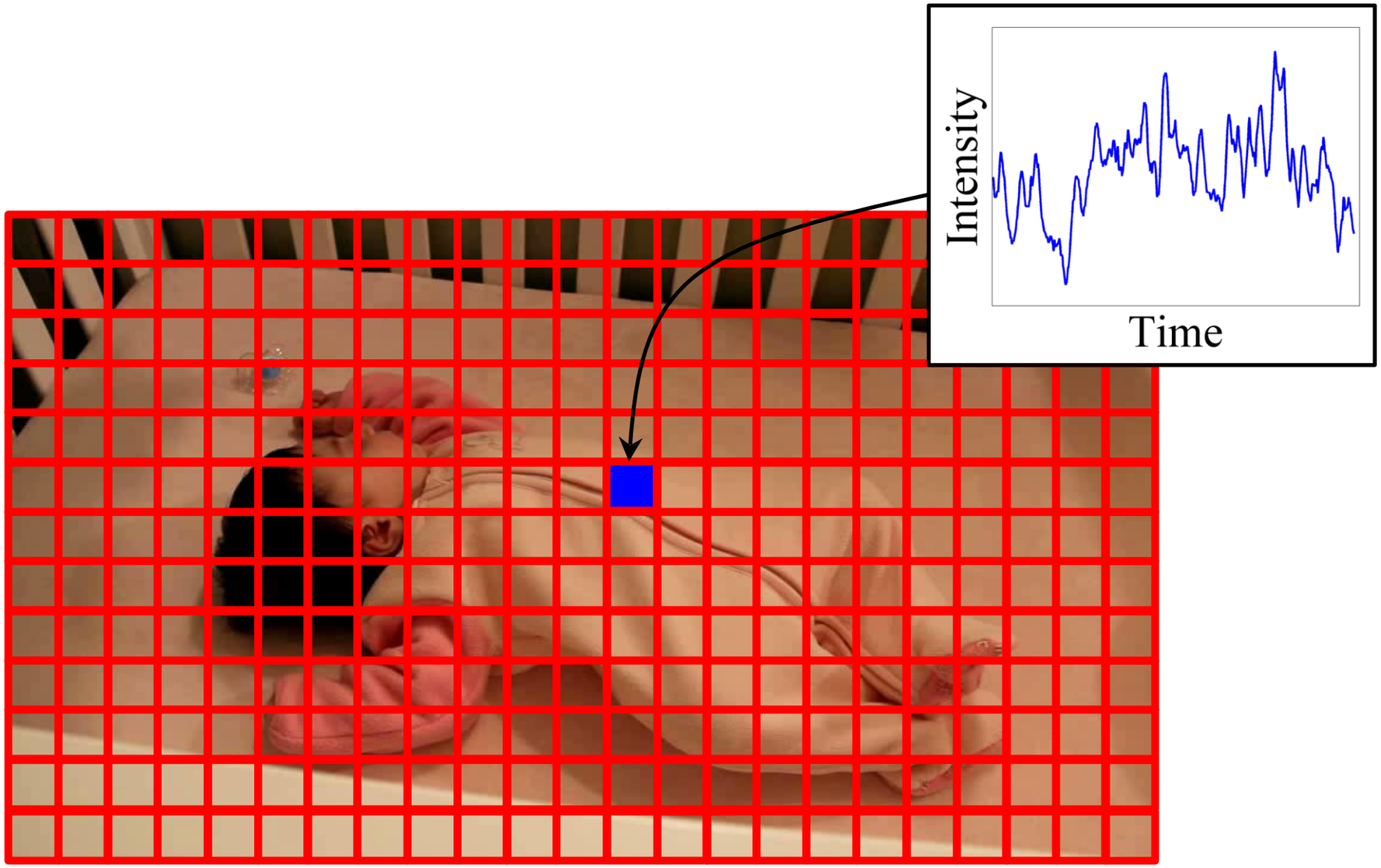
Massachusetts Institute of Technology



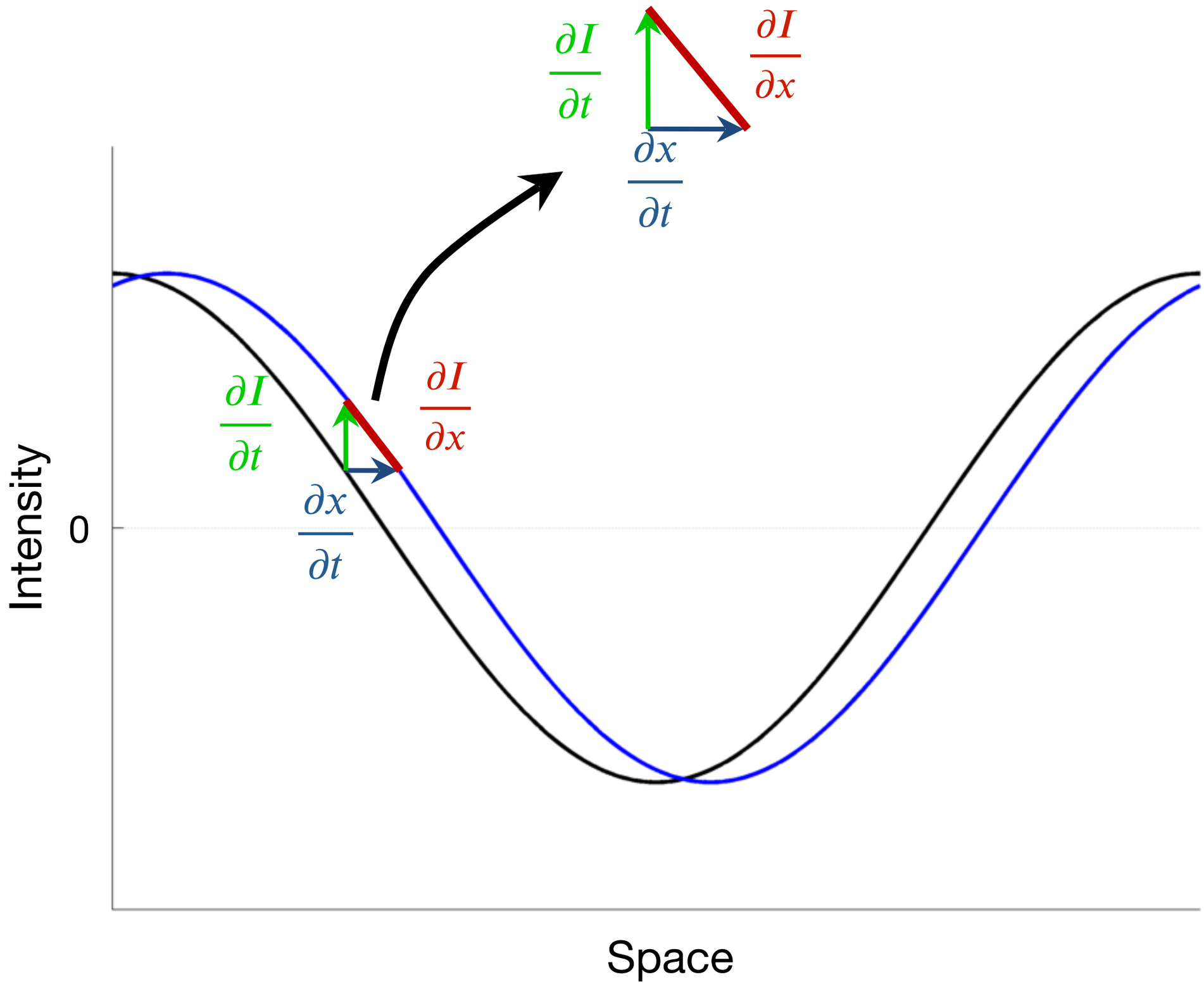
Revealing Invisible Changes In The World

Created for the NSF International Science & Engineering Visualization Challenge 2012

Why It Amplifies Motion



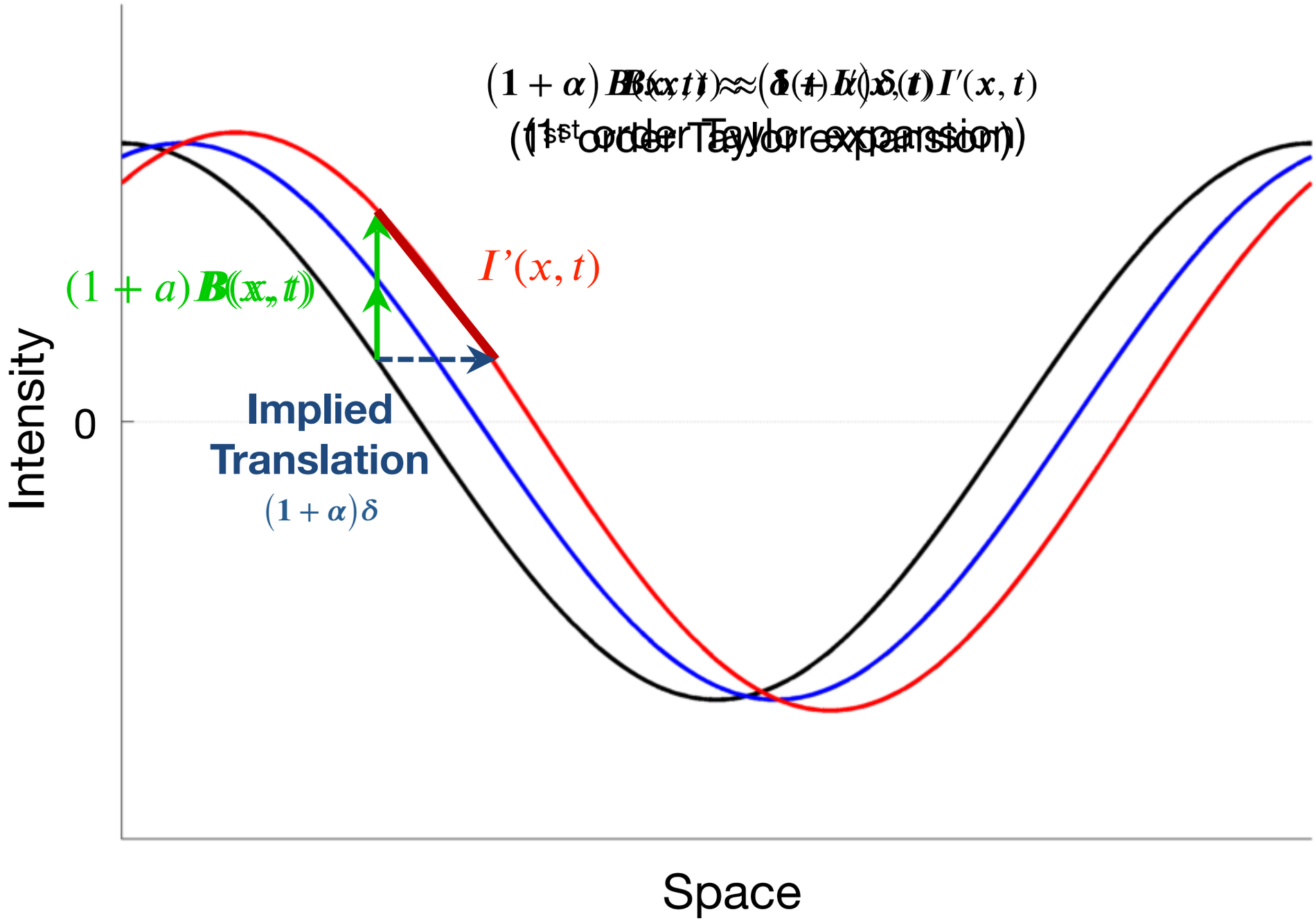
Differential Brightness Constancy



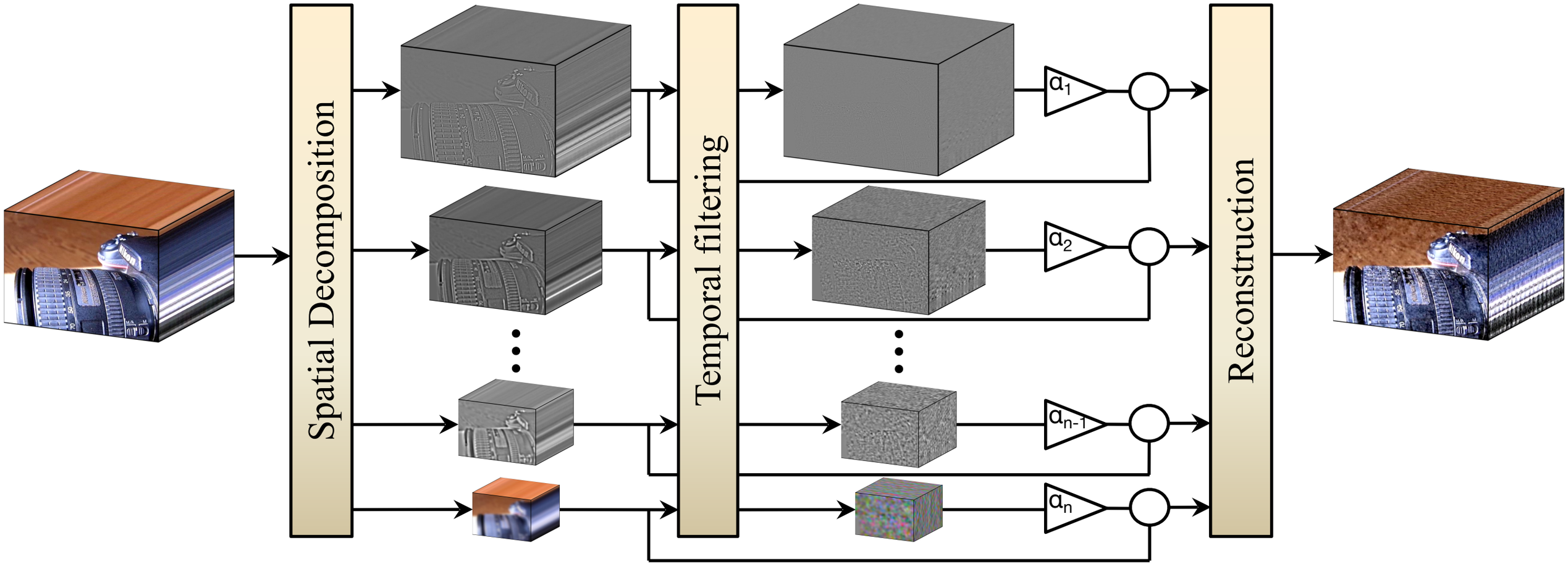
Relating Temporal and Spatial Changes

$$(1+\alpha)\,B(x,t) \approx B(x,t) + \alpha\,\delta(t)\,I'(x,t)$$

(1st order Taylor expansion)



Multi-scale Processing



Pyramids

- **Original version: Laplacian pyramid**
  - Works because motion is small and Taylor expansion arguments mean that amplifying intensity changes also amplifies movement (or any other complex change)
- **New version: complex-valued steerable pyramids**
  - Amplify phase changes
  - Because sine waves diagonalize translations, and translation is a phase shift

Simplified Eulerian Video Magnification

Simplified Video Magnification

- Pyramid so far mostly used to avoid amplifying high spatial frequencies
  - where the approximation doesn't work (motion)
  - where SNR is too bad (color)
- Instead, we'll use a simple low pass (Gaussian blur)

- Temporal band pass filtering:
  - remove high frequencies
  - remove low frequencies
- Perform this with Gaussians

Approach

- Input: Video (4D numpy array)

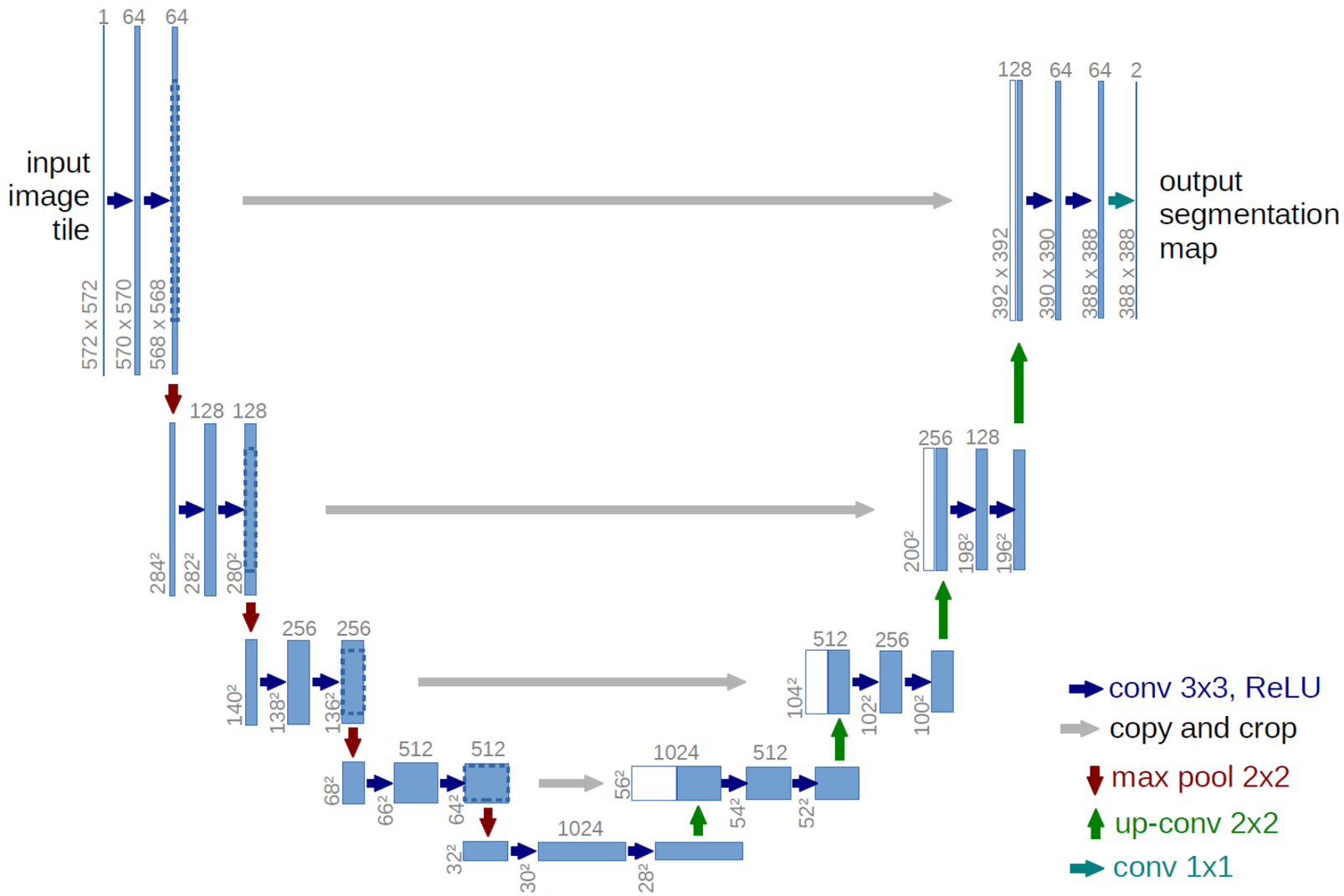
Approach

- Input: Video (4D numpy array)
- Extract signal of interest
  - Spatial low pass (Gaussian blur)
  - Remove high frequencies: (Gaussian across time)
  - Subtract low frequencies: (Another Gaussian across time)
- Amplify signal of interest, add back
  - Bell/whistle: filter in YUV and amplify Y differently from UV

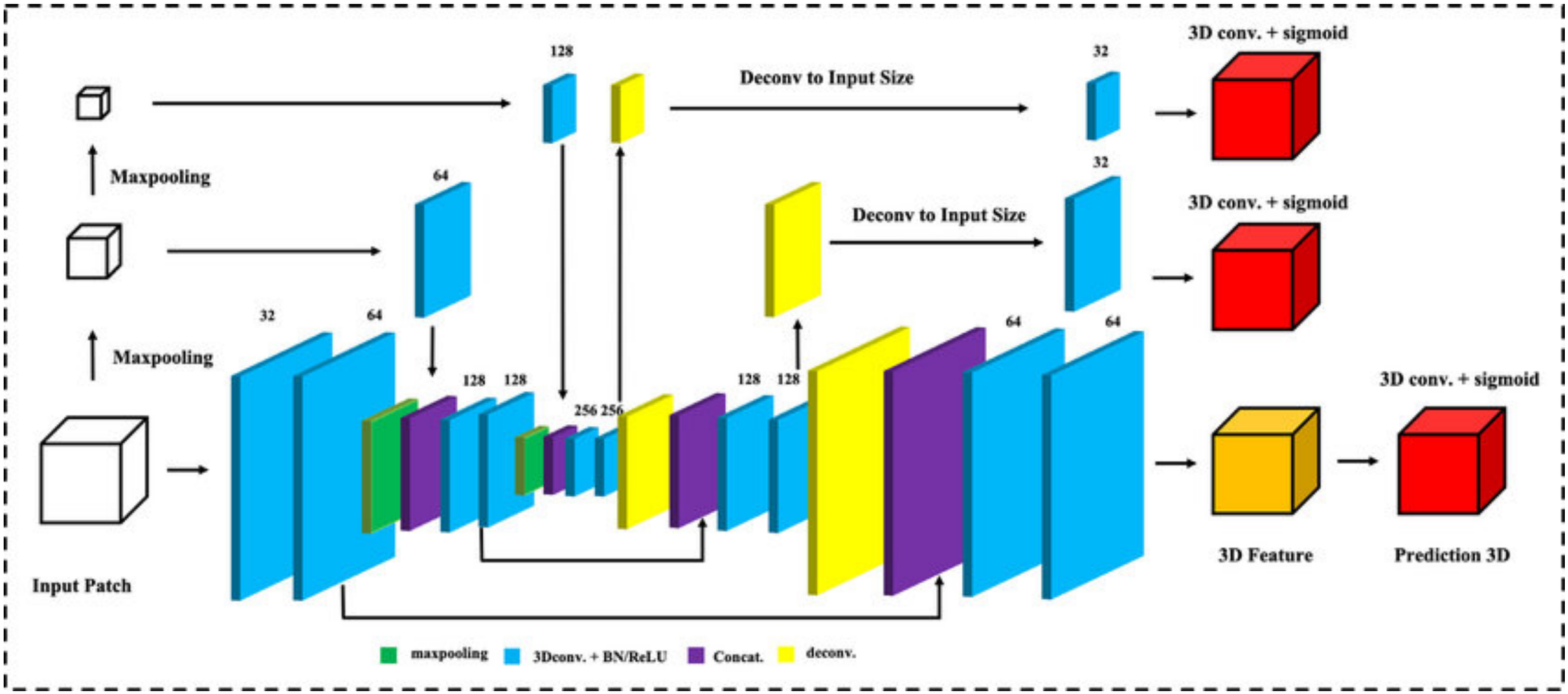
U-Net

U-Net

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



U-Net


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

Extra notes

- Laplacian Pyramids
- Wavelets
- Steerable Pyramids

- UNet

- Image Fusion
- Denoising
- Local Laplacian Filtering
- Motion Magnification
- Mipmapping
- Fast processing?
- Coarse-to-fine (optical flow)
- Hybrid images
- UNet and coarsening
- Compression
- Texture synthesis