Computational Photography

Digital Visual Effects, Spring 2007

Yung-Yu Chuang

2007/5/22

with slides by Fredo Durand, Ramesh Raskar, Sylvain Paris, Soonmin Bae

Computational photography

wikipedia:

Computational photography refers broadly to computational imaging techniques that enhance or extend the capabilities of digital photography. The output of these techniques is an ordinary photograph, but one that could not have been taken by a traditional camera.

What is computational photography

- Convergence of image processing, computer vision, computer graphics and photography
- Digital photography:
  - Simply mimics traditional sensors and recording by digital technology
  - Involves only simple image processing
- Computational photography
  - More elaborate image manipulation, more computation
  - New types of media (panorama, 3D, etc.)
  - Camera design that take computation into account

Computational photography

DigiVFX

- One of the most exciting fields.
- Symposium on Computational Photography and Video, 2005
- Full-semester courses in MIT, CMU, Stanford, GaTech, University of Delaware
- A new book by Raskar and Tumblin is coming out in SIGGRAPH 2007.

Siggraph 2006 Papers (16/86=18.6%)

Hybrid Images
Drag-and-Drop Pasting
Two-scale Tone Management for Photographic Look
Interactive Local Adjustment of Tonal Values
Image-Based Material Editing
Flash Matting
Natural Video Matting using Camera Arrays
Removing Camera Shake From a Single Photograph
Coded Exposure Photography: Motion Deblurring
Photo Tourism: Exploring Photo Collections in 3D
AutoCollage
Photographing Long Scenes With Multi-Viewpoint Panoramas
Projection Defocus Analysis for Scene Capture and Image Display
Multiview Radial Catadioptric Imaging for Scene Capture
Light Field Microscopy
Fast Separation of Direct and Global Components of a Scene Using High Frequency Illumination

Siggraph 2007 Papers (23/108=21.3%)

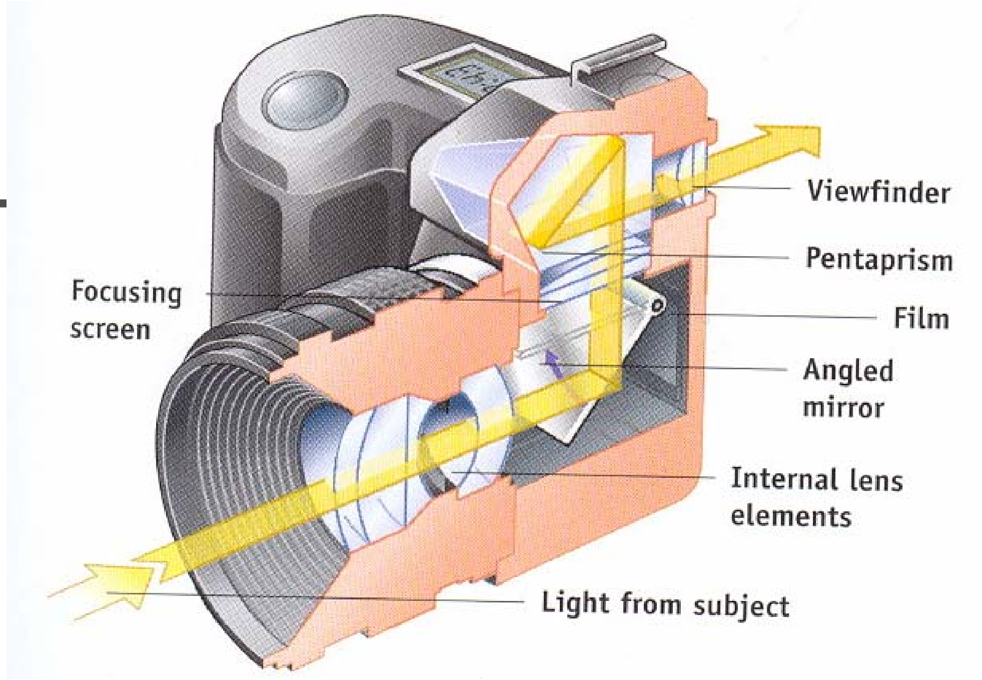
Image Deblurring with Blurred/Noisy Image Pairs
Photo Clip Art
Scene Completion Using Millions of Photographs
Soft Scissors: An Interactive Tool for Realtime High Quality Matting
Seam Carving for Content-Aware Image Resizing
Detail-Preserving Shape Deformation in Image Editing
Veiling Glare in High Dynamic Range Imaging
Do HDR Displays Support LDR content? A Psychophysical Evaluation
Ldr2hdr: On-the-fly Reverse Tone Mapping of Legacy Video and Photographs
Rendering for an Interactive 360-Degree Light Field Display
Multiscale Shape and Detail Enhancement from Multi-light Image Collections
Post-Production Facial Performance Relighting Using Reflectance Transfer
Active Refocusing of Images and Videos
Multi-aperture Photography
Dappled Photography: Mask-Enhanced Cameras for Heterodyned Light Fields and Coded Aperture Refocusing
Image and Depth from a Conventional Camera with a Coded Aperture
Capturing and Viewing Gigapixel Images
Efficient Gradient-Domain Compositing Using Quadtrees
Image Upsampling via Imposed Edges Statistics
Joint Bilateral Upsampling
Factored Time-Lapse Video
Computational Time-Lapse Video
Real-Time Edge-Aware Image Processing With the Bilateral Grid

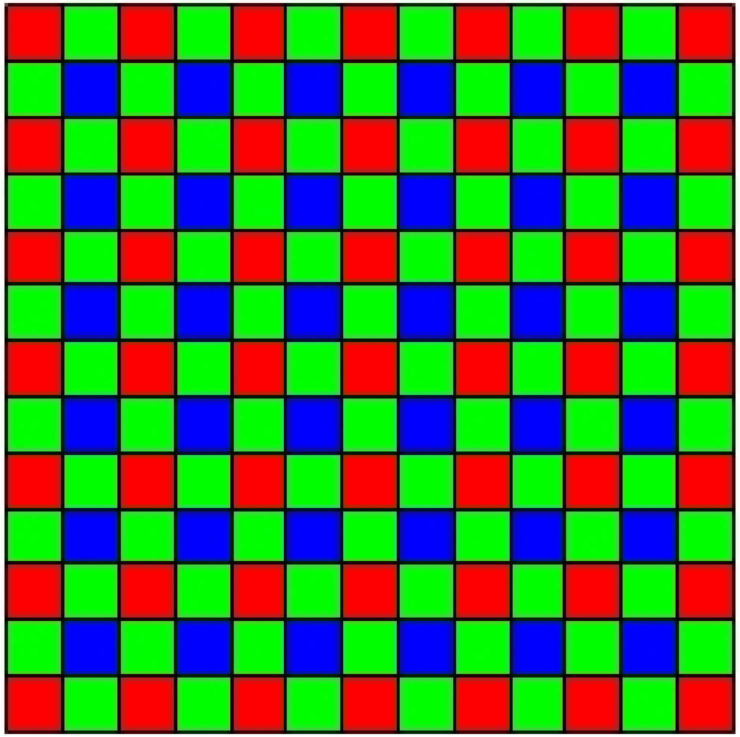
Scope

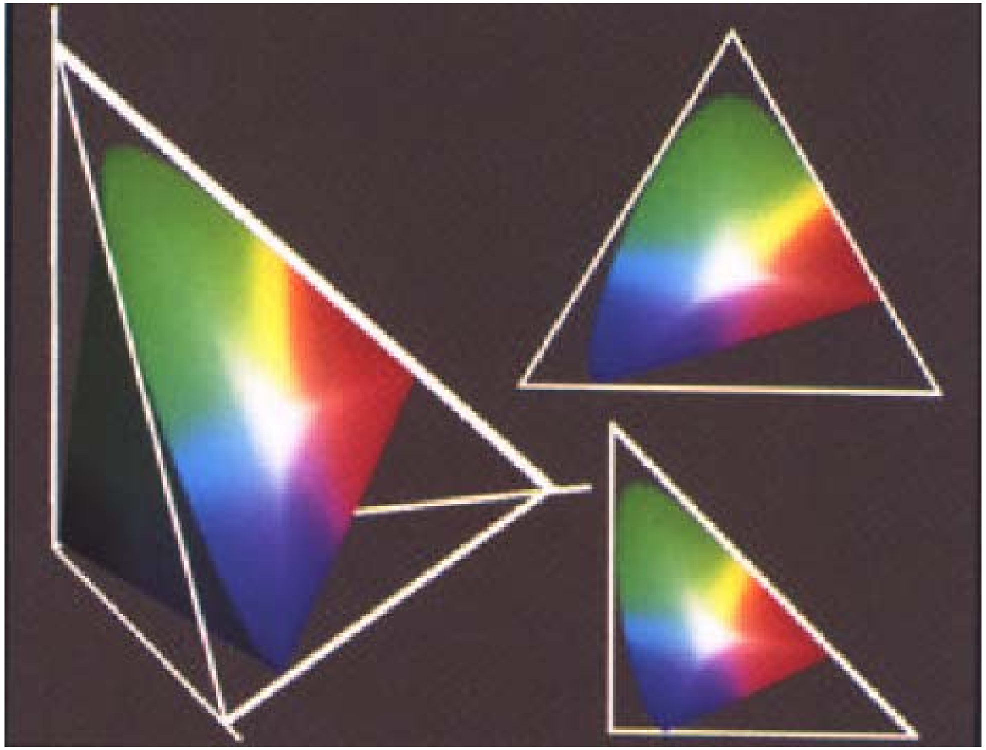
- We can't yet set its precise definition. The following are scopes of what researchers are exploring in this field.
  - Record a richer visual experience
  - Overcome long-standing limitations of conventional cameras
  - Enable new classes of visual signal
  - Enable synthesis impossible photos

Scope

- Image formation
- Color and color perception

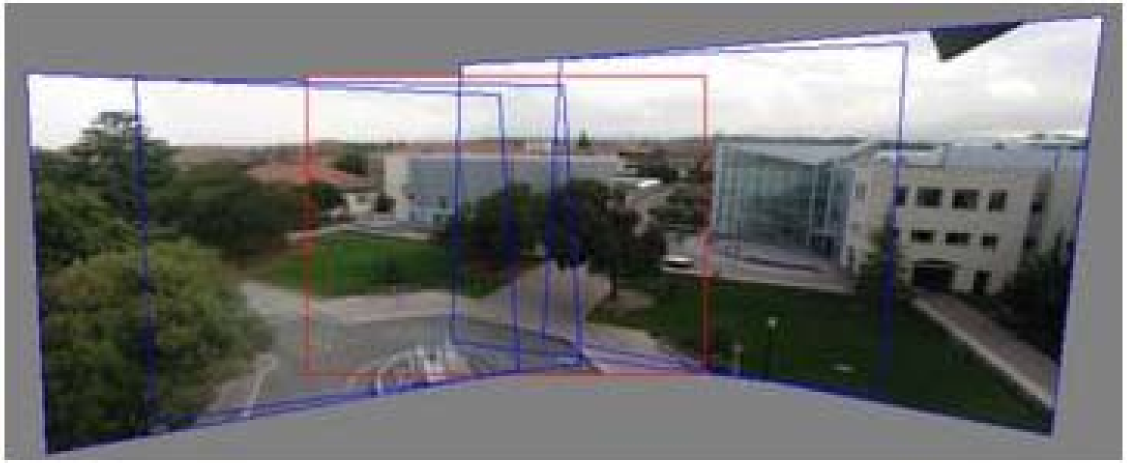




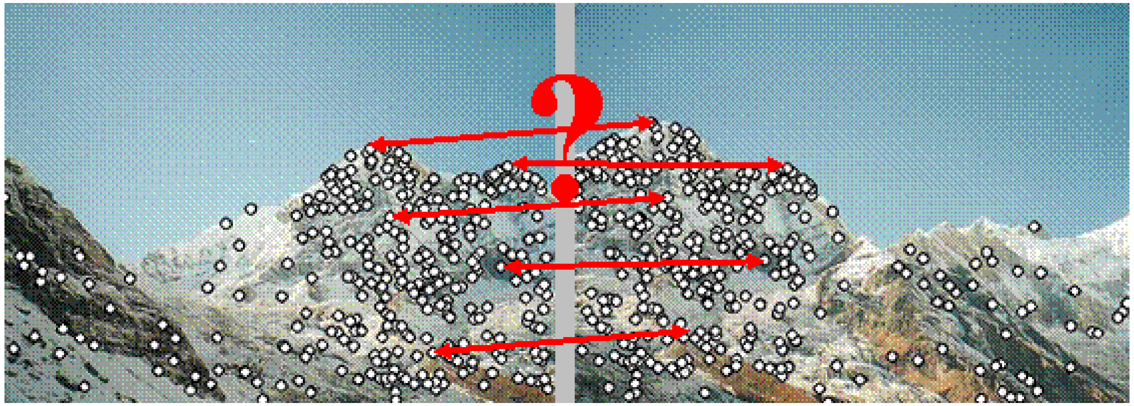


Scope

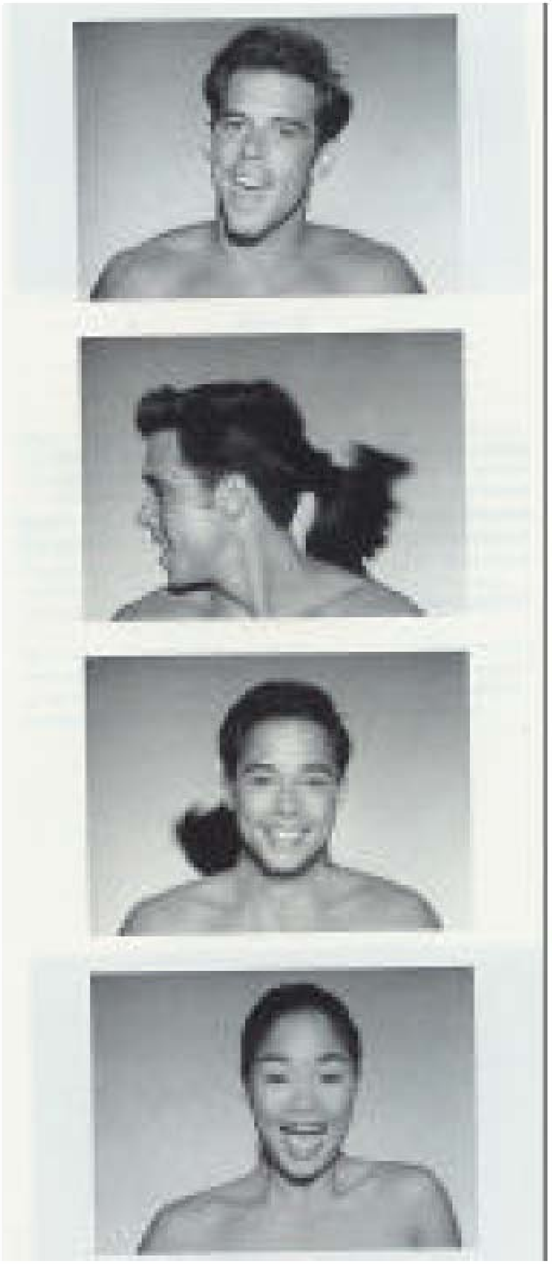
- Panoramic imaging



- Image and video registration



- Spatial warping operations



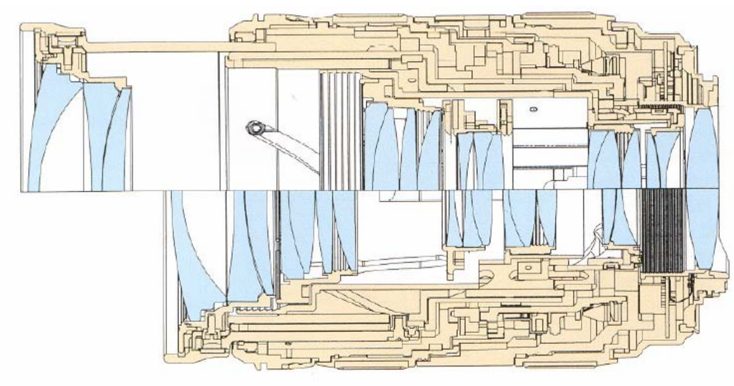
Scope

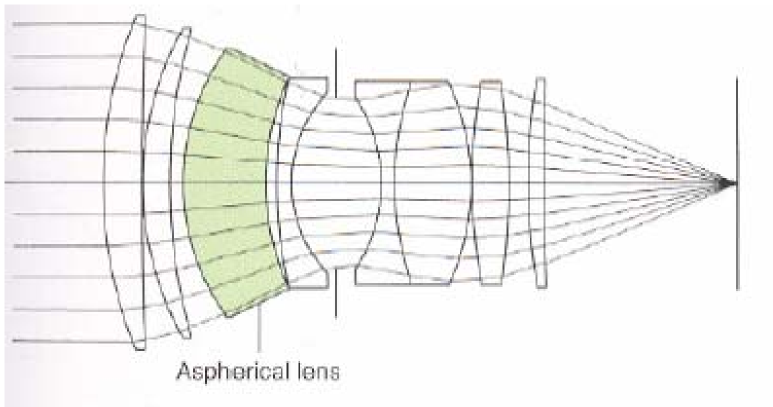
- High Dynamic Range Imaging
- Bilateral filtering and HDR display
- Matting

Scope

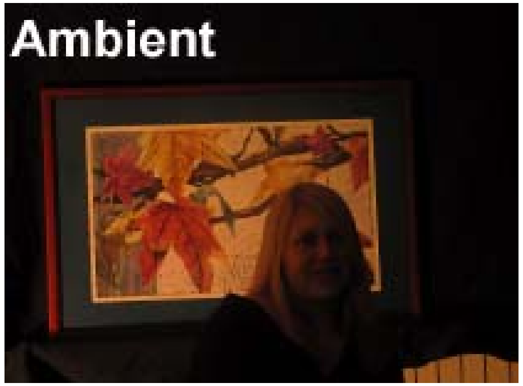
DigiVFX

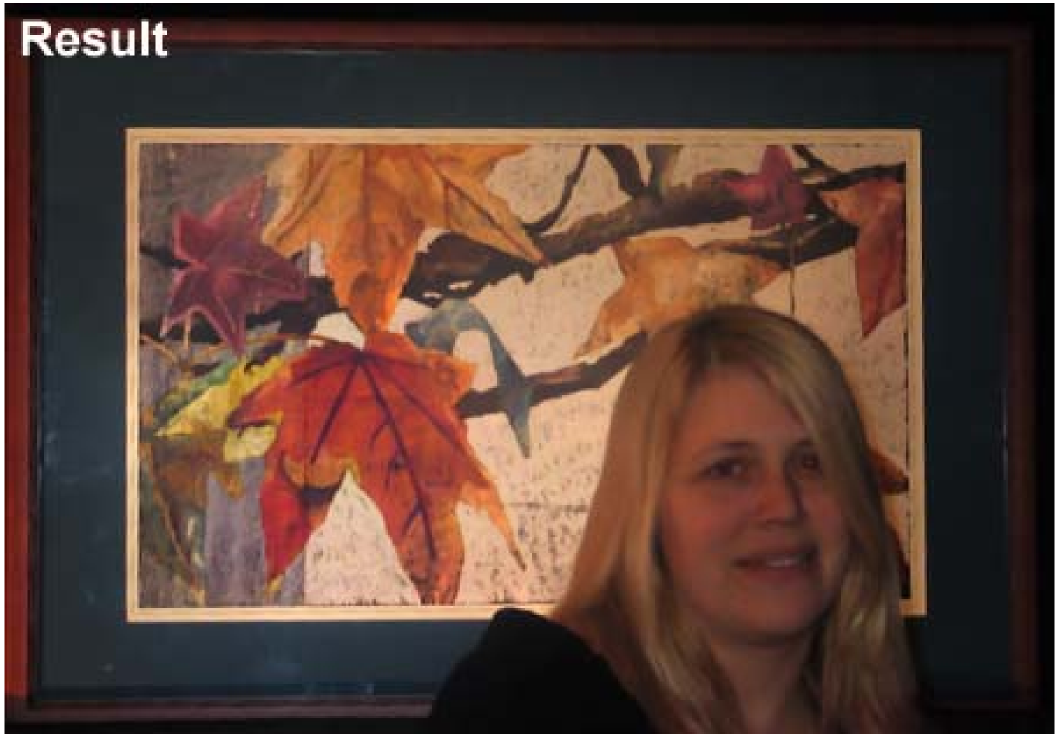
- Active flash methods
- Lens technology
- Depth and defocus





Removing Photography Artifacts using Gradient Projection and Flash-Exposure Sampling





Continuous flash

Flash = 0.0

Flash = 1.0

Flash = 0.3

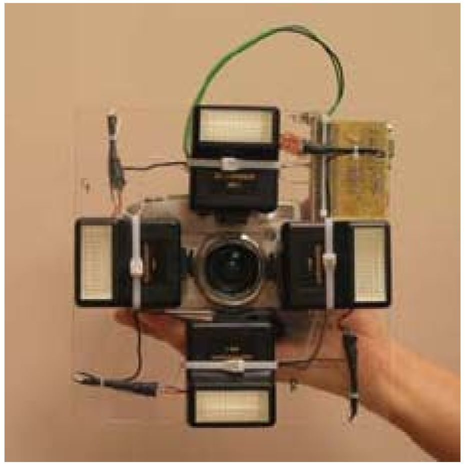
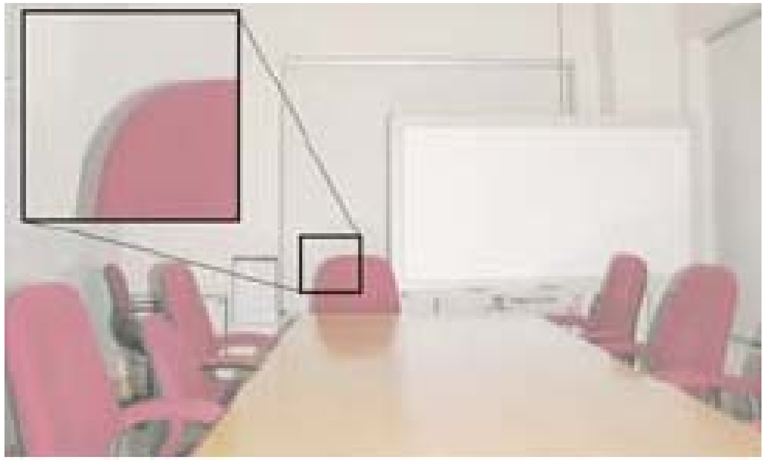
Flash = 0.7

Flash = 1.4

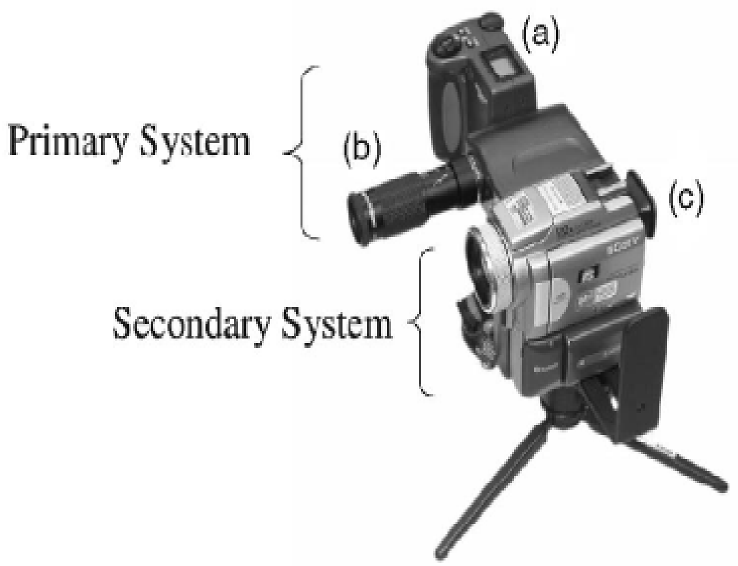
Flash matting

Depth Edge Detection and Stylized Rendering Using a Multi-Flash Camera

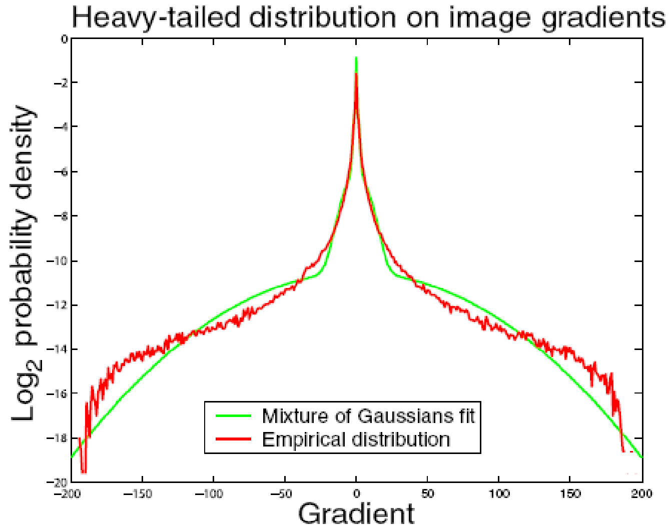


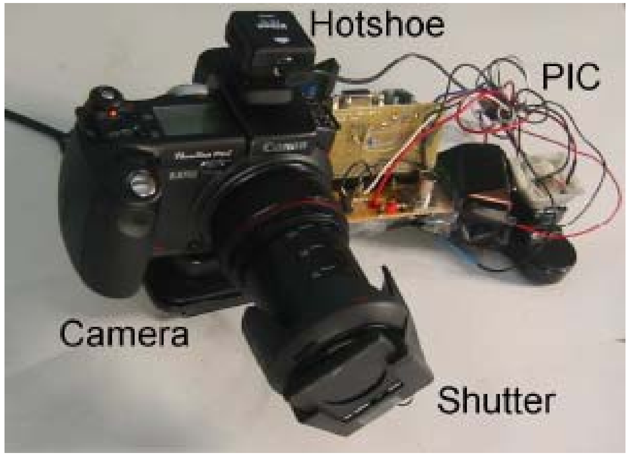
Motion-Based Motion Deblurring

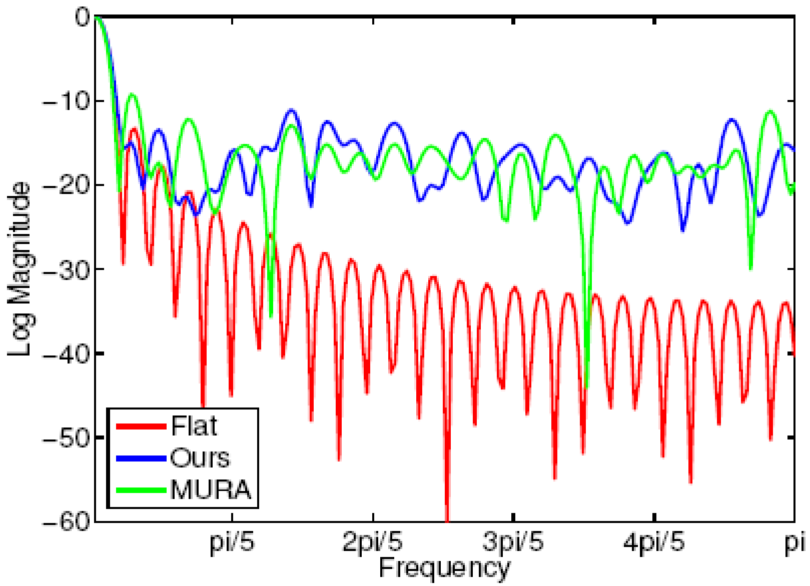


Removing Camera Shake from a Single Photograph



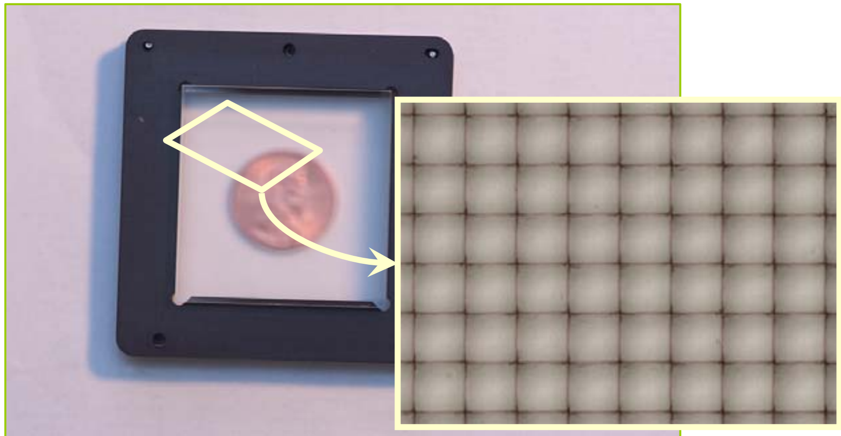
Motion Deblurring using Fluttered Shutter





Scope

- Future cameras
- Plenoptic function and light fields



Scope

- Gradient image manipulation

sources/destinations

cloning

seamless cloning

Scope

- Taking great pictures

Art Wolfe

Ansel Adams

Scope

- Non-parametric image synthesis, inpainting, analogies

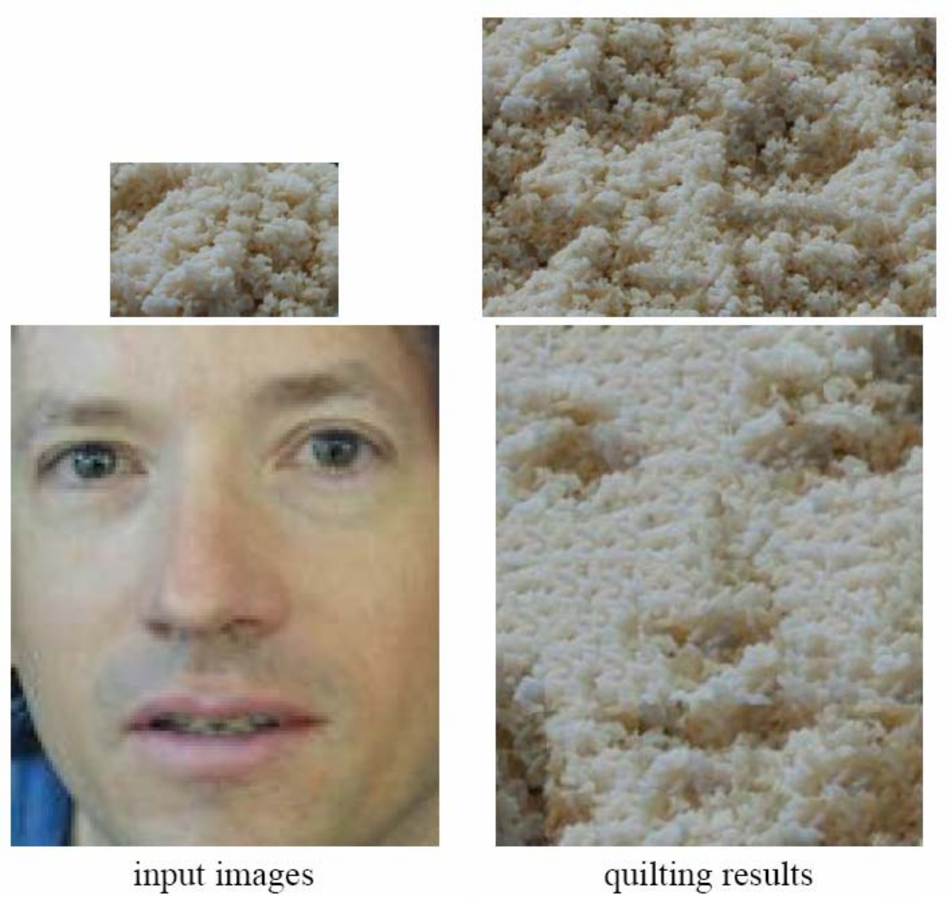


Figure 1 An image analogy. Our problem is to compute a new "analogous" image B' that relates to B in "the same way" as A' relates to A. Here, A, A', and B are inputs to our algorithm, and B' is the output. The full-size images are shown in Figures 10 and 11.

Scope

Motion analysis

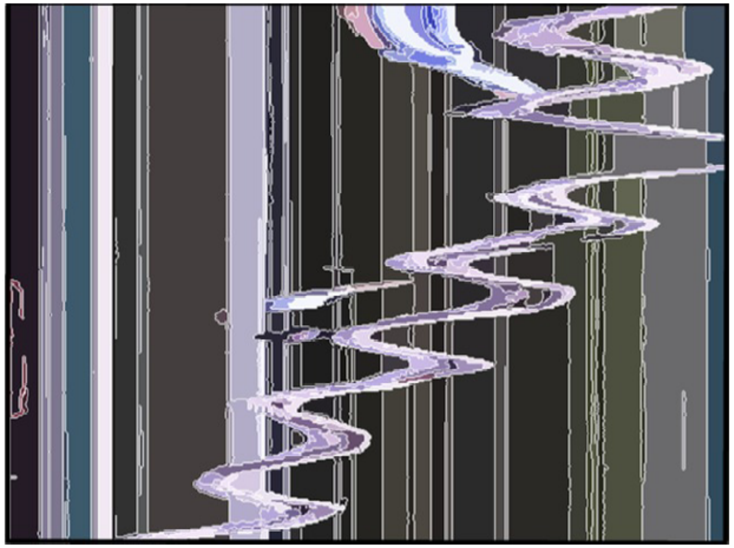
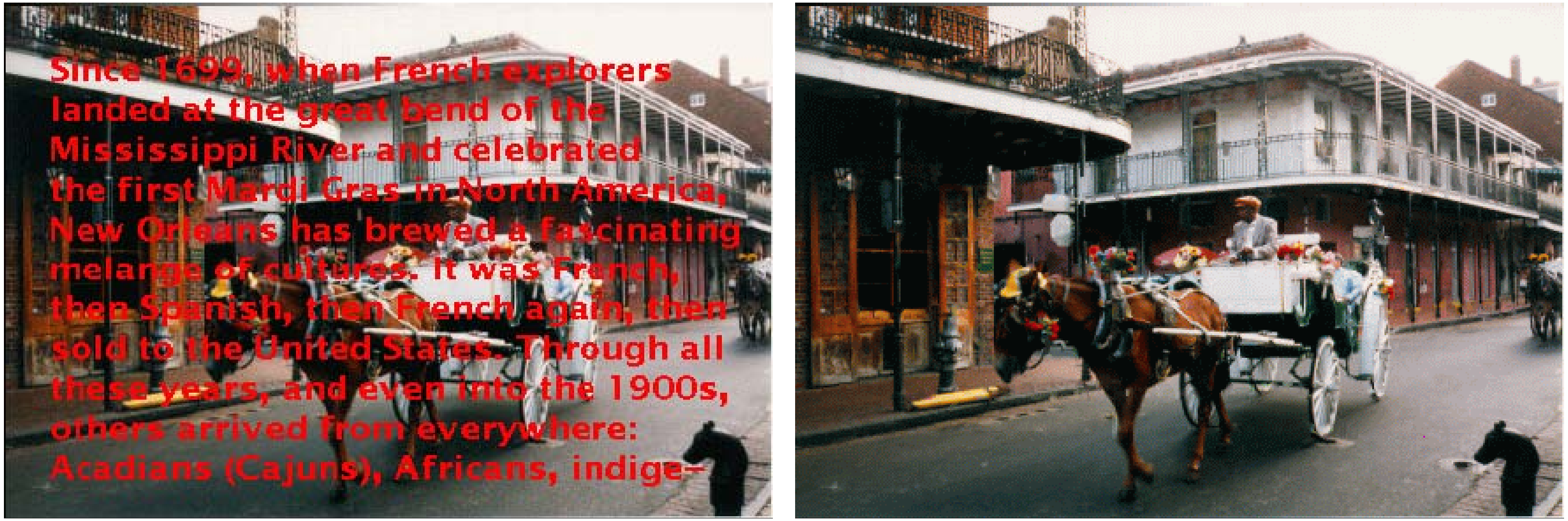


Image Inpainting

DigiVFX



Object Removal by Exemplar-Based Inpainting

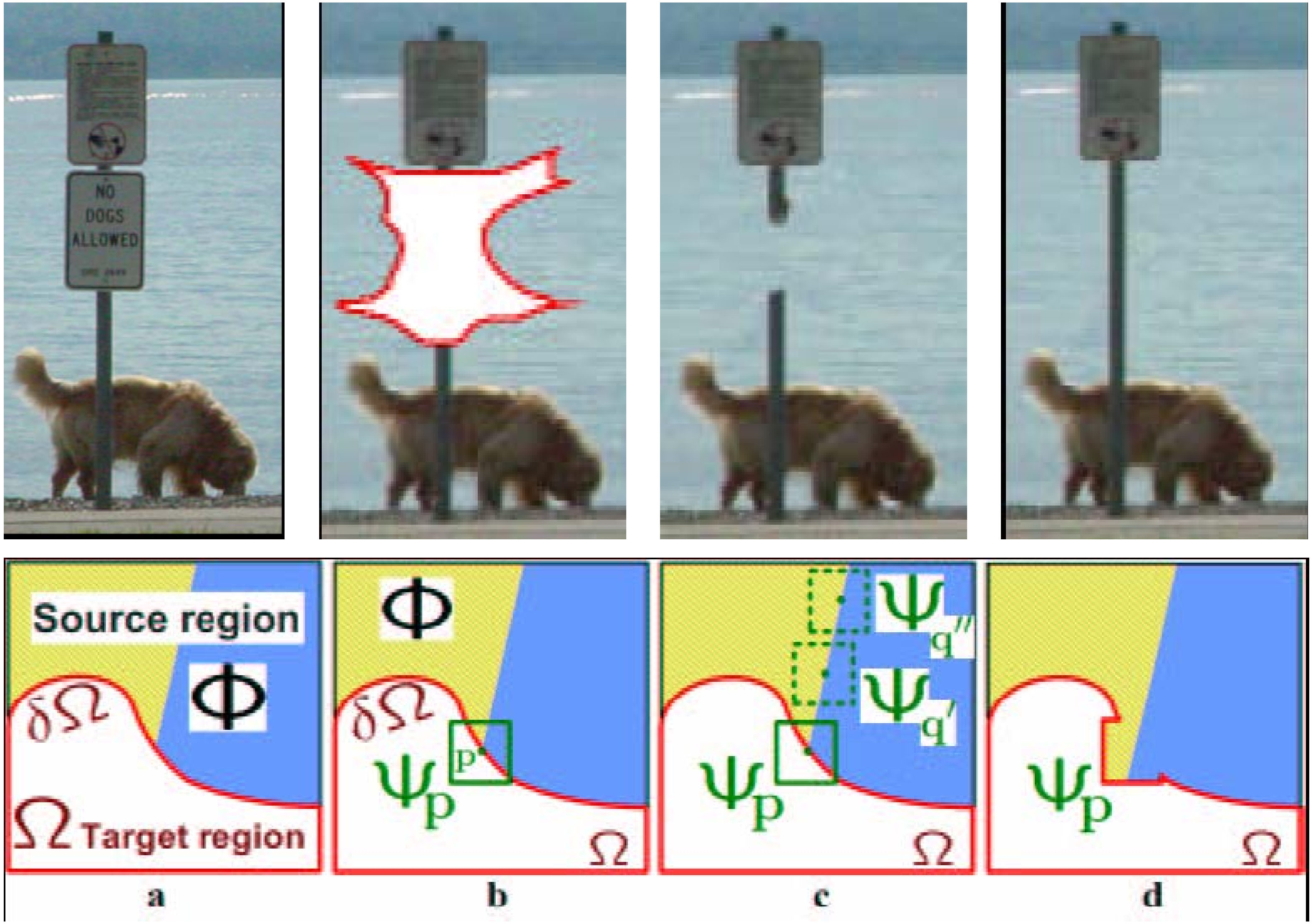
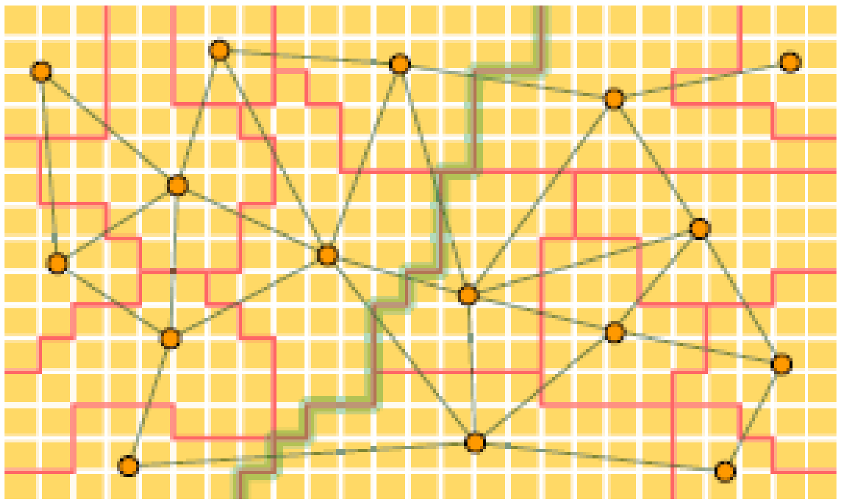


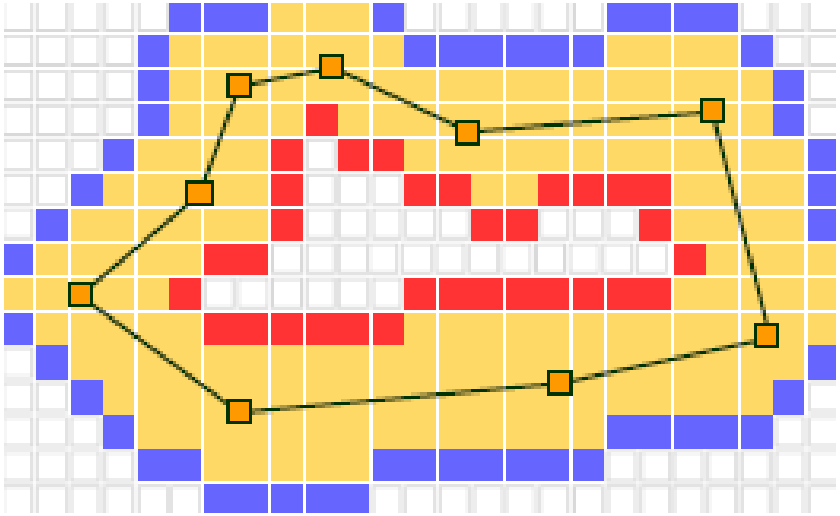
Image Completion with Structure Propagation

Lazy snapping

Lazy snapping

- Pre-segmentation
- Boundary Editing





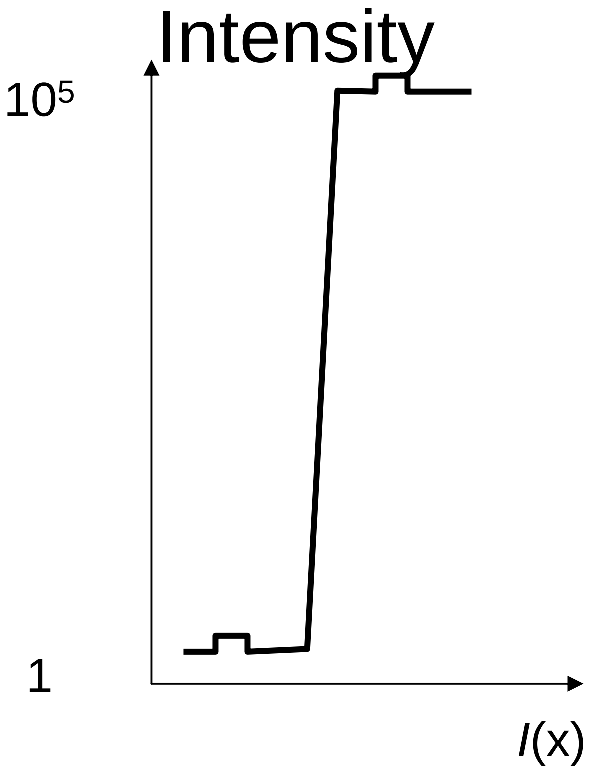
Grab Cut - Interactive Foreground Extraction using Iterated Graph Cuts

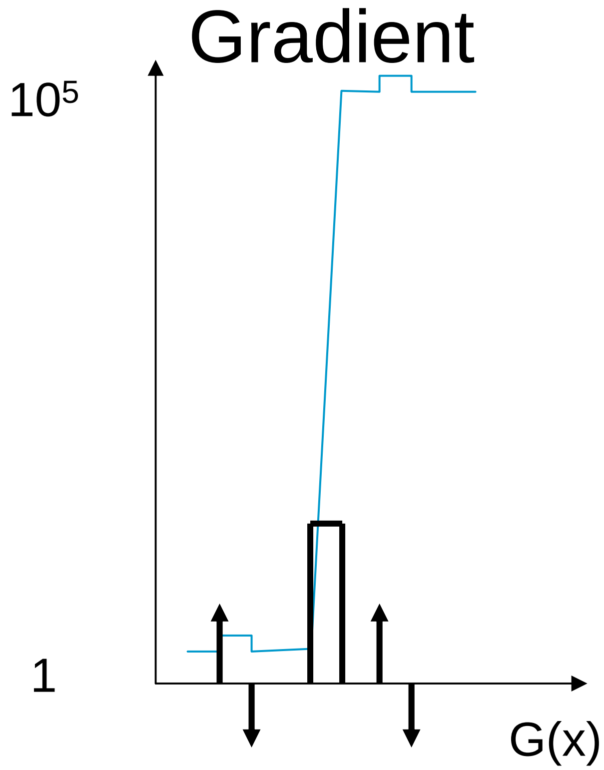
Image Tools

- Gradient domain operations,
  - Tone mapping, fusion and matting
- Graph cuts,
  - Segmentation and mosaicing
- Bilateral and Trilateral filters,
  - Denoising, image enhancement

Gradient domain operators

Intensity Gradient in 1D



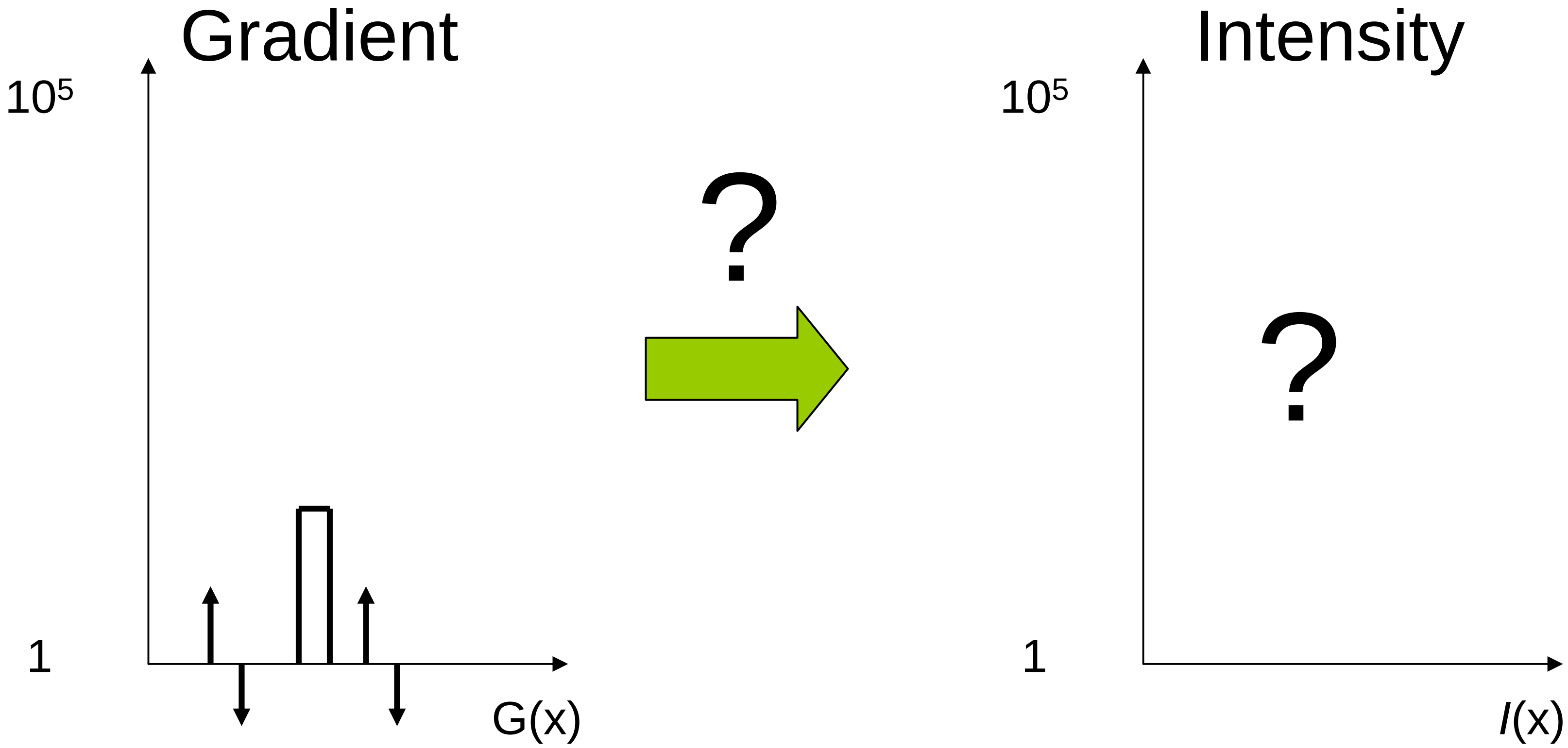


Gradient at x,

$$G(x) = I(x+1) - I(x)$$

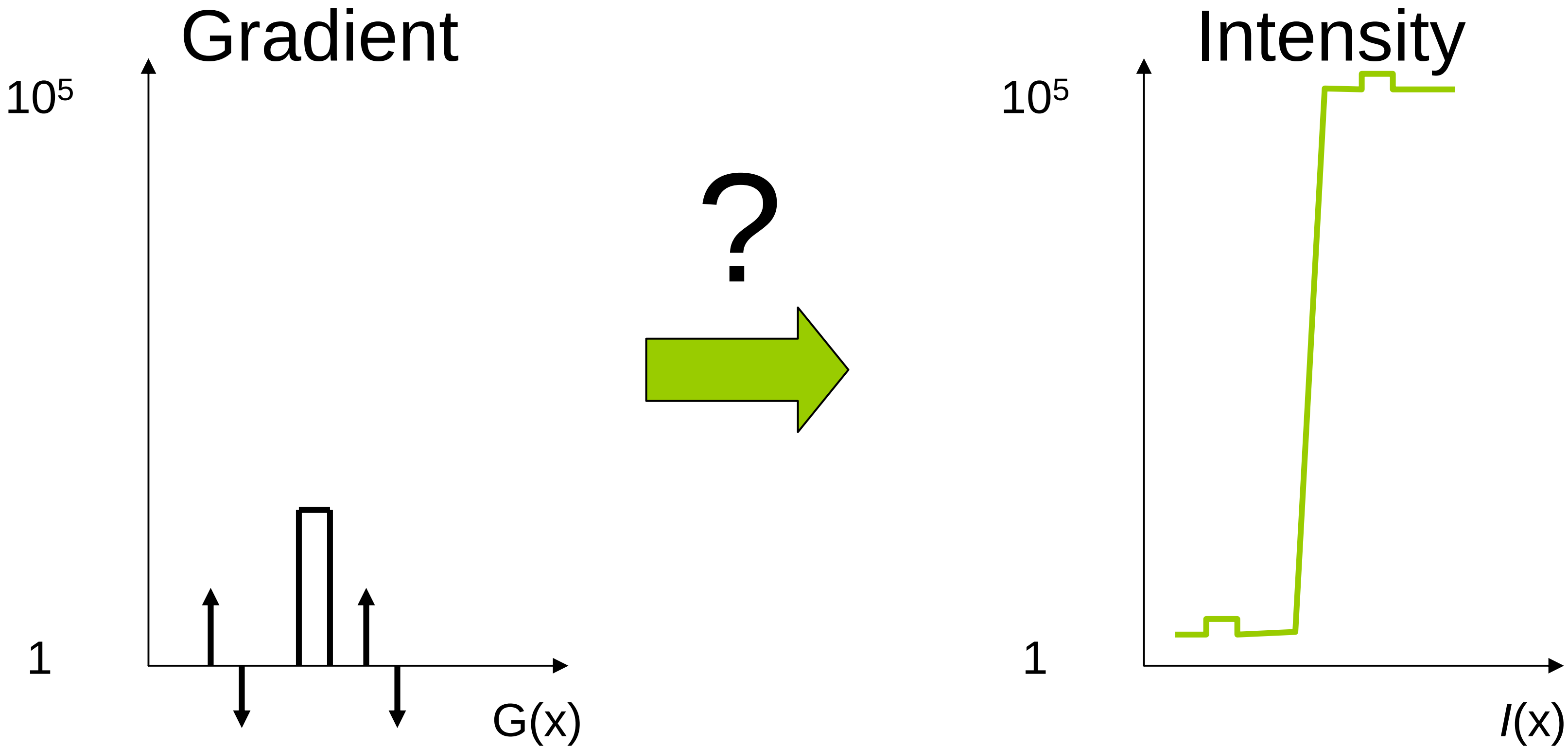
Forward Difference

Reconstruction from Gradients



For *n* intensity values, about *n* gradients

Reconstruction from Gradients

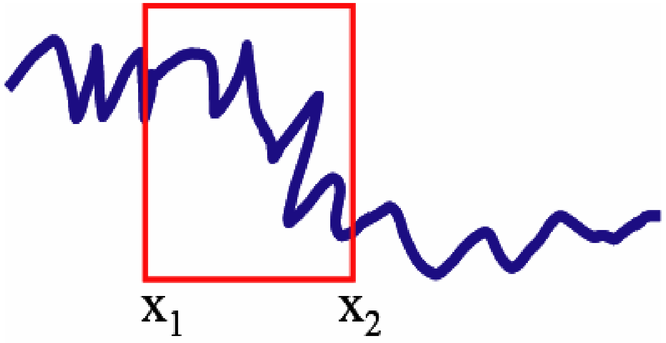


1D Integration

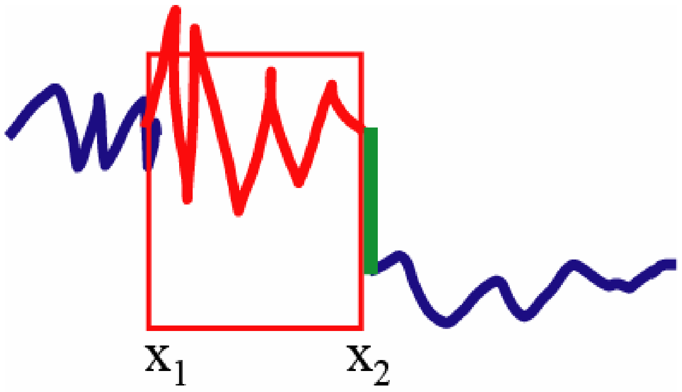
$$I(x) = I(x-1) + G(x)$$

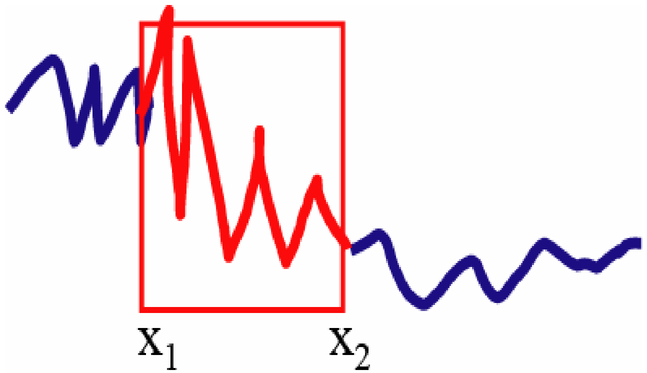
Cumulative sum

1D case with constraints

Seamlessly paste  onto 

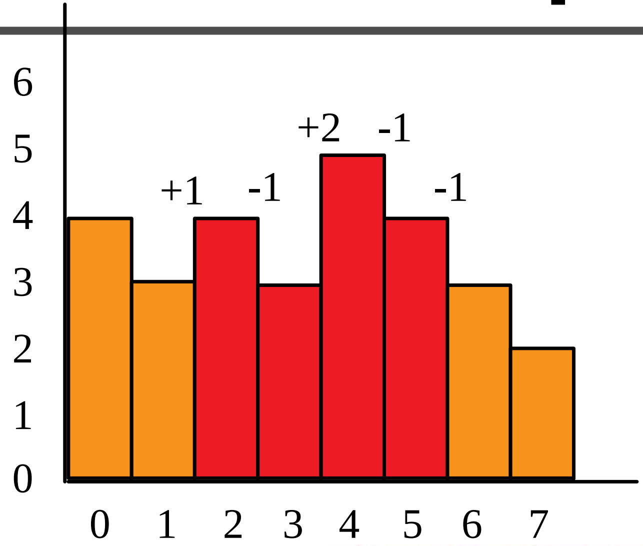
Just add a linear function so that the boundary condition is respected



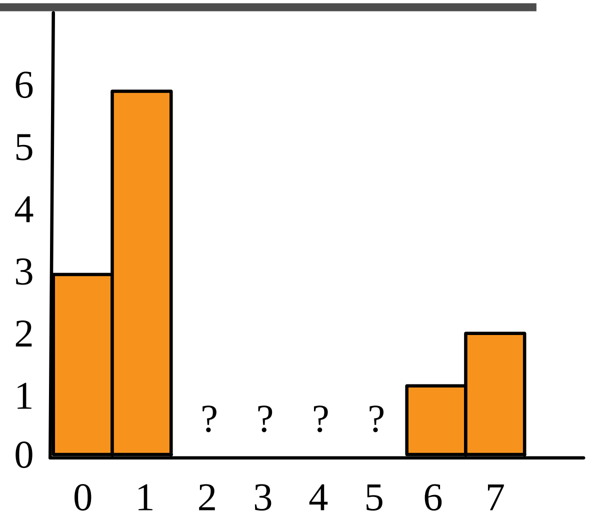


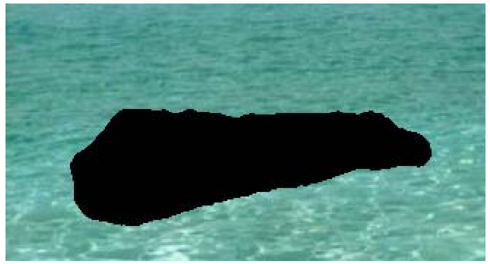
Discrete 1D example: minimization

- Copy



to

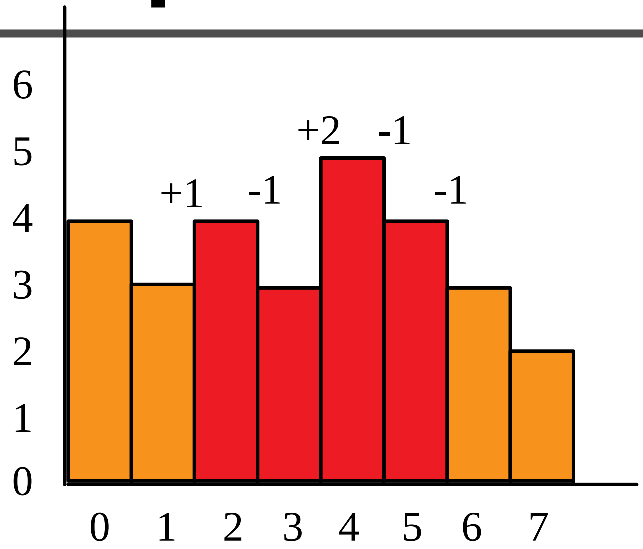


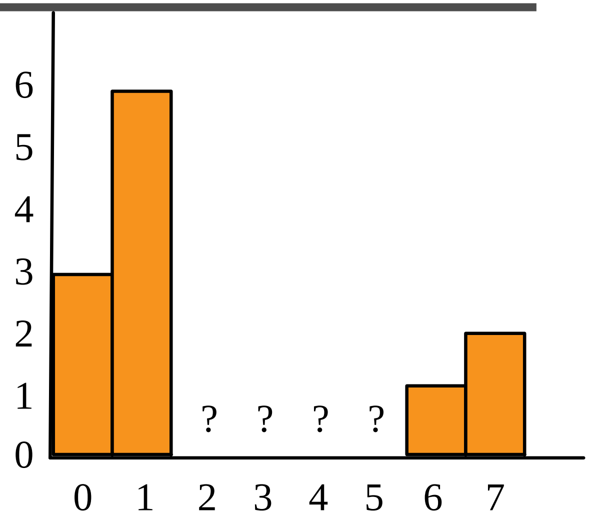


- Min $((f_2-f_1)-1)^2$
- Min $((f_3-f_2)-(-1))^2$
- Min $((f_4-f_3)-2)^2$
- Min $((f_5-f_4)-(-1))^2$
- Min $((f_6-f_5)-(-1))^2$

With
$f_1=6$
$f_6=1$

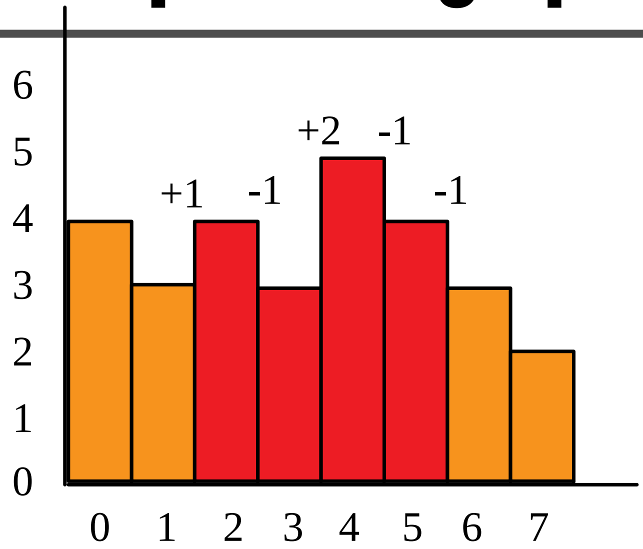
1D example: minimization

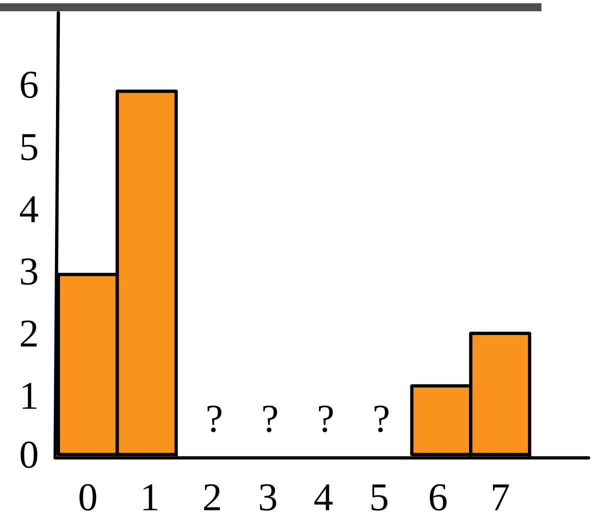
- Copy 

to 

- Min $((f_2-6)-1)^2$ ==> $f_2^2+49-14f_2$
- Min $((f_3-f_2)-(-1))^2$ ==> $f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
- Min $((f_4-f_3)-2)^2$ ==> $f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
- Min $((f_5-f_4)-(-1))^2$ ==> $f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
- Min $((1-f_5)-(-1))^2$ ==> $f_5^2+4-4f_5$

1D example: big quadratic

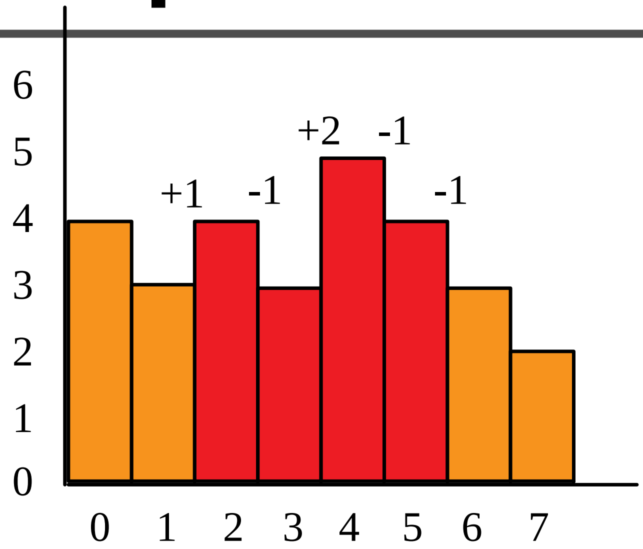
- Copy 

to 

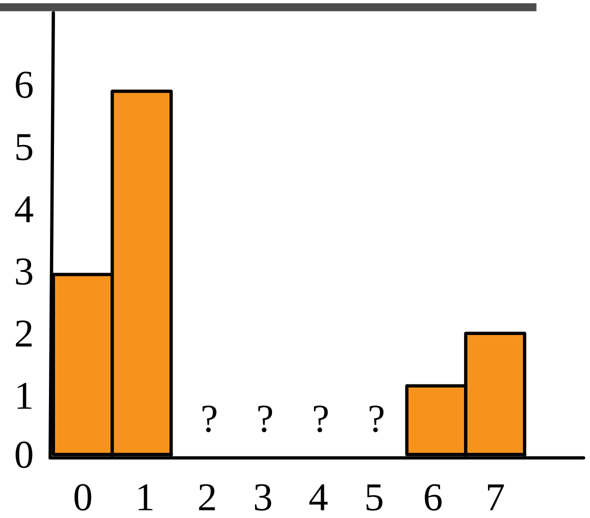
- Min $(f_2^2+49-14f_2$
  $+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
  $+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
  $+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
  $+ f_5^2+4-4f_5)$
  Denote it Q

1D example: derivatives

- Copy



to



Min $(f_2^2+49-14f_2$
$+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
$+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
$+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
$+ f_5^2+4-4f_5)$

Denote it Q

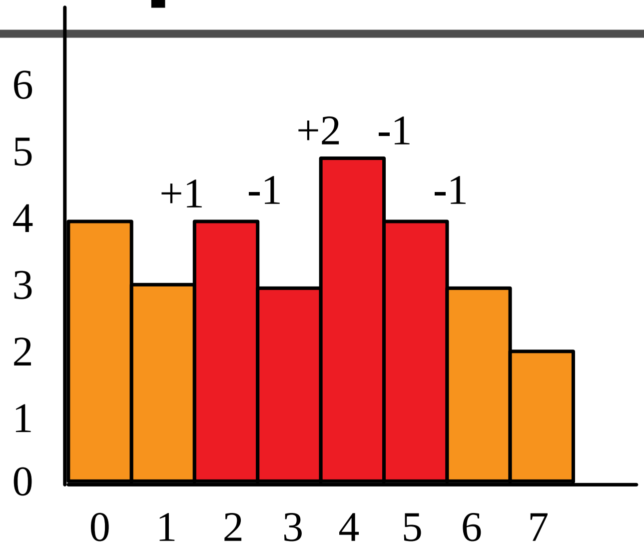
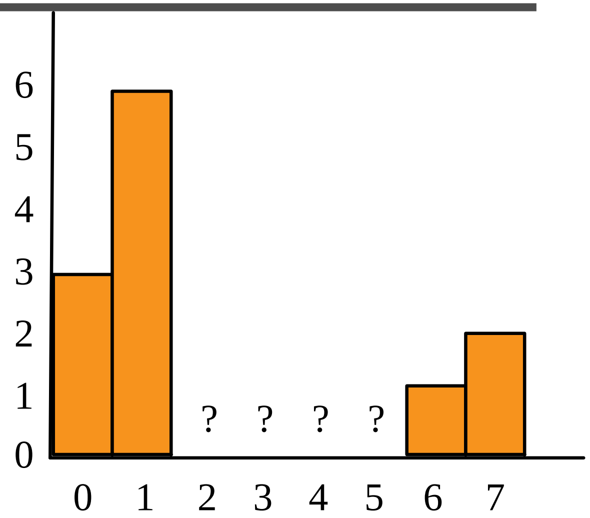
$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$$

$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$$

$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$$

$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$$

1D example: set derivatives to zero

- Copy  to 

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$$

$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$$

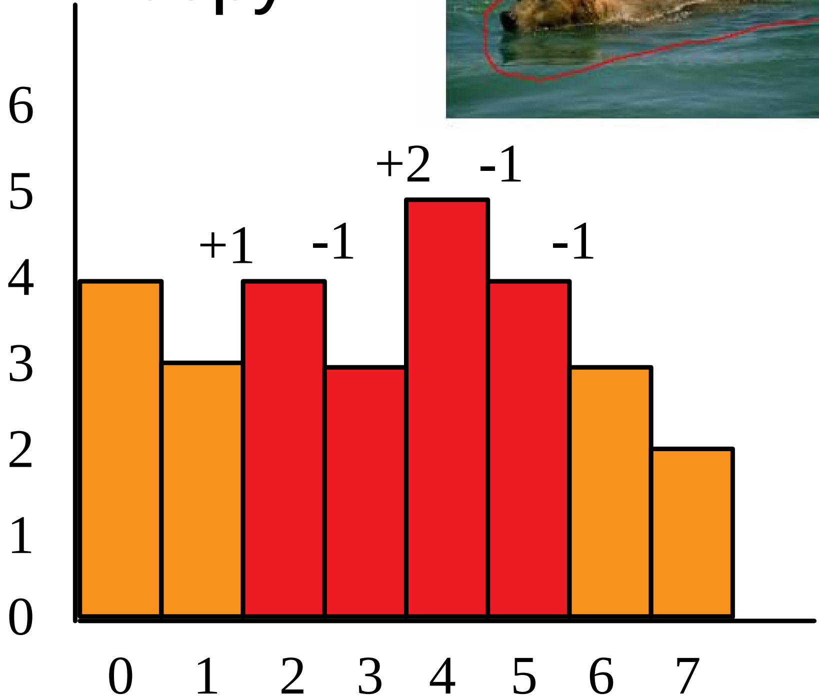
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$$

$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$$

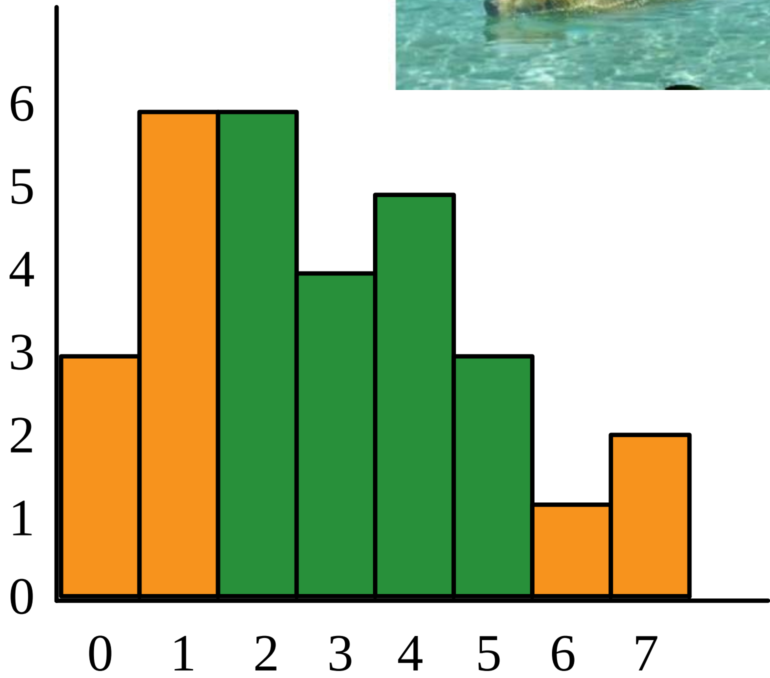
==> $$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

1D example

- Copy



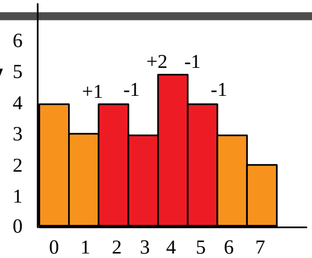
to

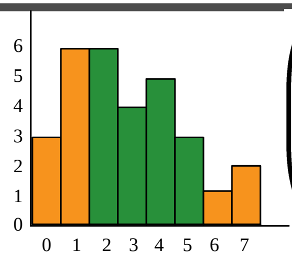


$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 5 \\ 3 \end{pmatrix}$$

1D example: remarks

- Copy 

to 

$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

- Matrix is sparse
- Matrix is symmetric
- Everything is a multiple of 2
  - because square and derivative of square
- Matrix is a convolution (kernel -2 4 -2)
- Matrix is independent of gradient field. Only RHS is
- Matrix is a second derivative

Intensity Gradient in 2D

Gradient at x,y as Forward Differences

$$G_x(x,y) = I(x+1, y) - I(x,y)$$
$$G_y(x,y) = I(x, y+1) - I(x,y)$$

$$G(x,y) = (G_x, G_y)$$

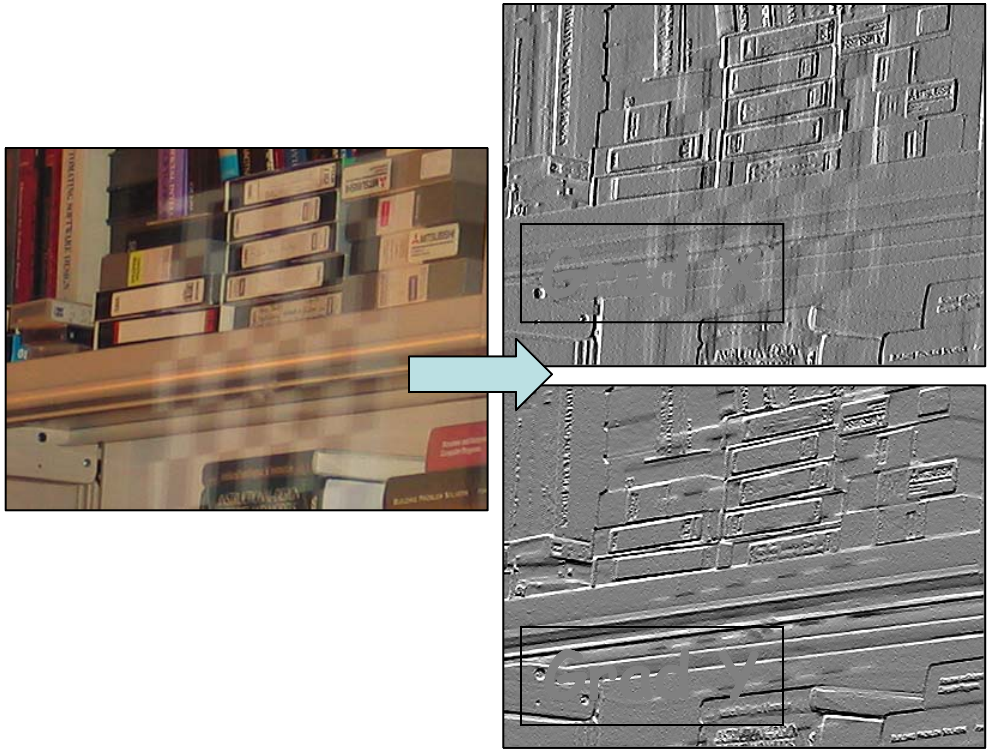
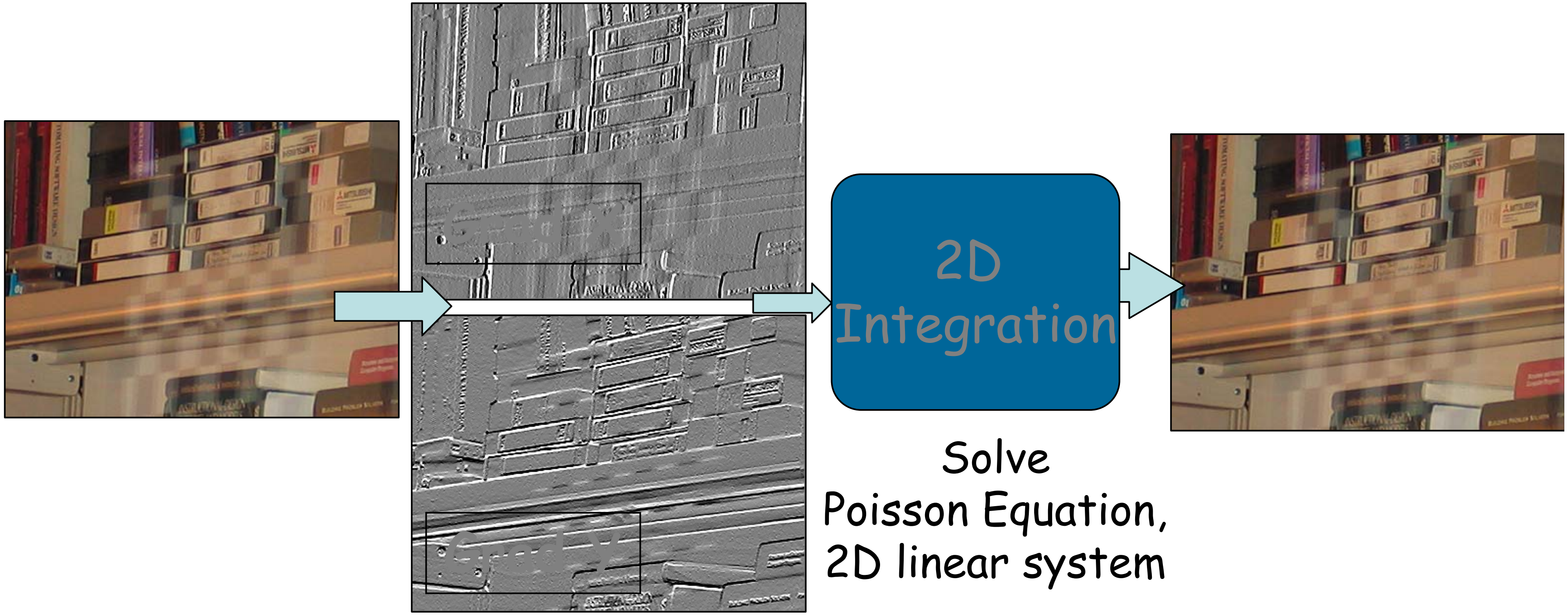


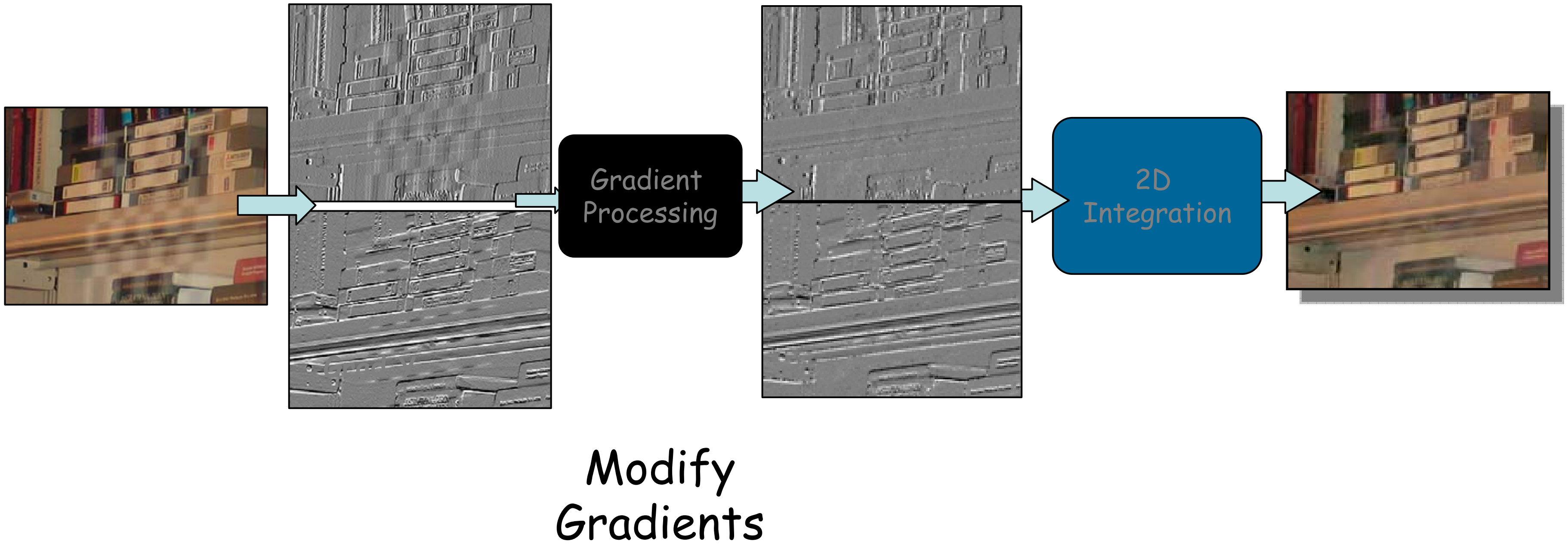
Image Intensity Gradients in 2D

Sanity Check:
Recovering Original Image



Intensity Gradient Manipulation

A Common Pipeline



2D case with constraints

- Given vector field *v* (pasted gradient), find the value of *f* in unknown region that optimize:

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

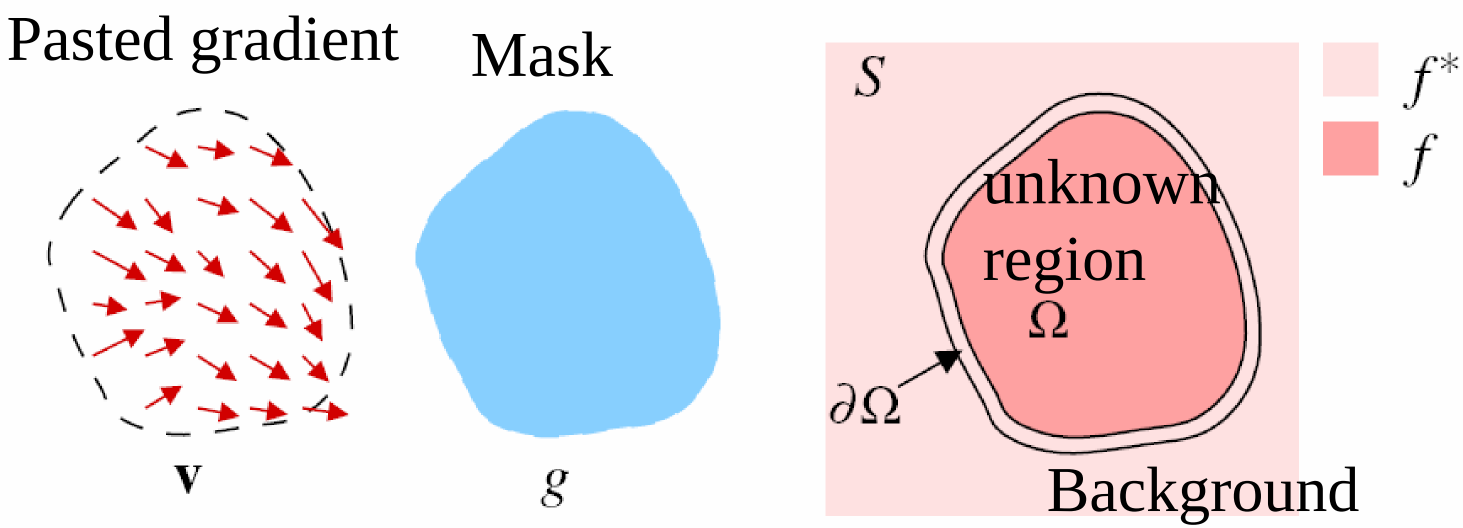
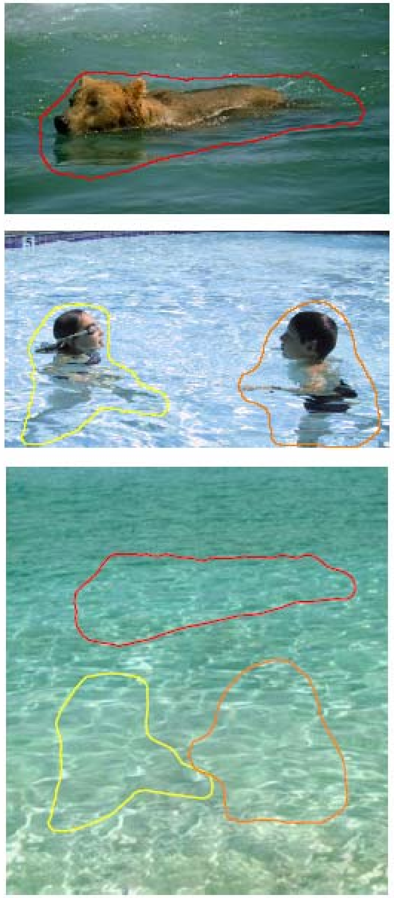


Figure 1: **Guided interpolation notations**. Unknown function f interpolates in domain Ω the destination function f^*, under guidance of vector field $\mathbf{v}$, which might be or not the gradient field of a source function g.

Poisson image editing

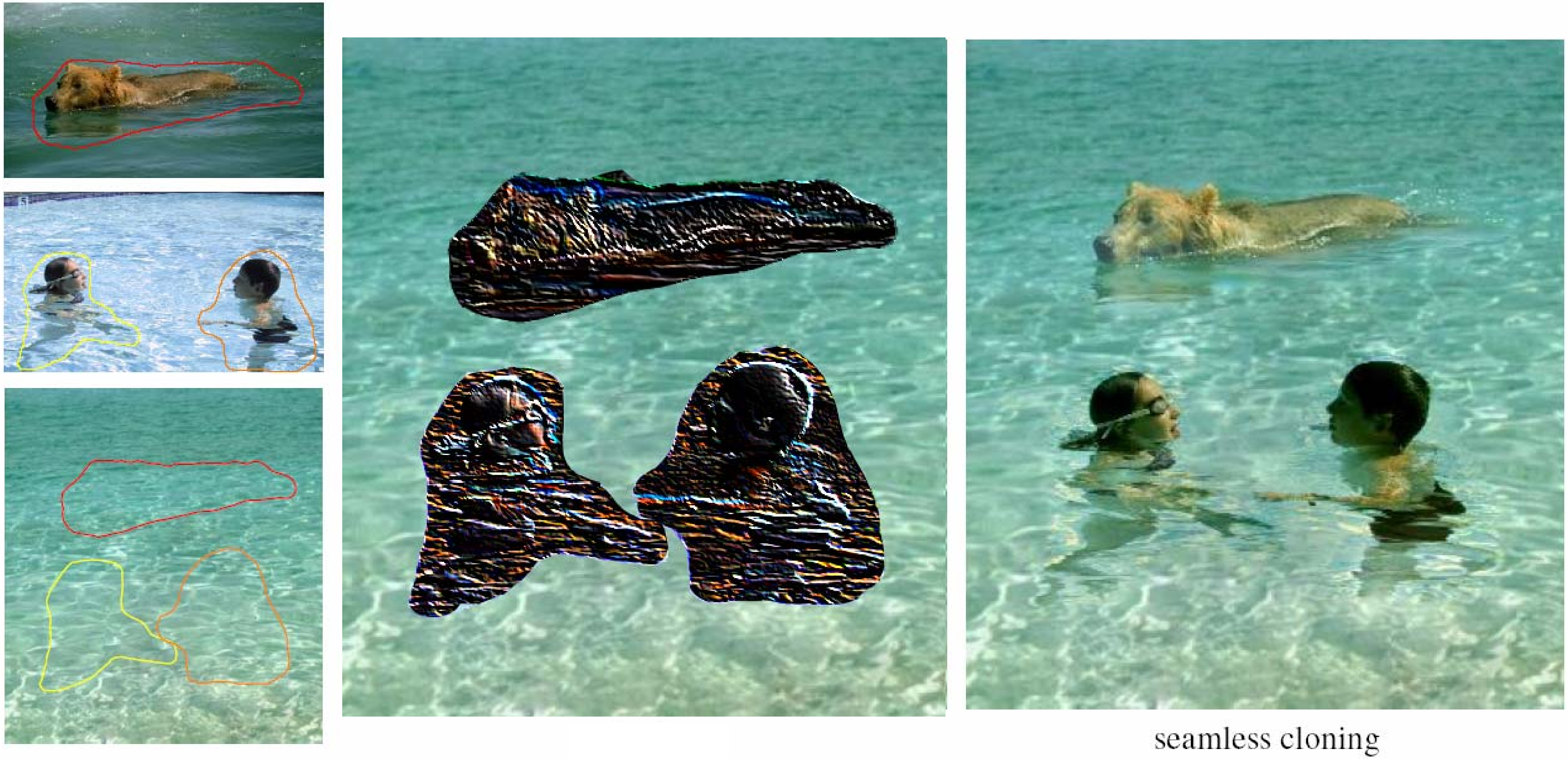
Problems with direct cloning


sources/destinations


cloning

From Perez et al. 2003

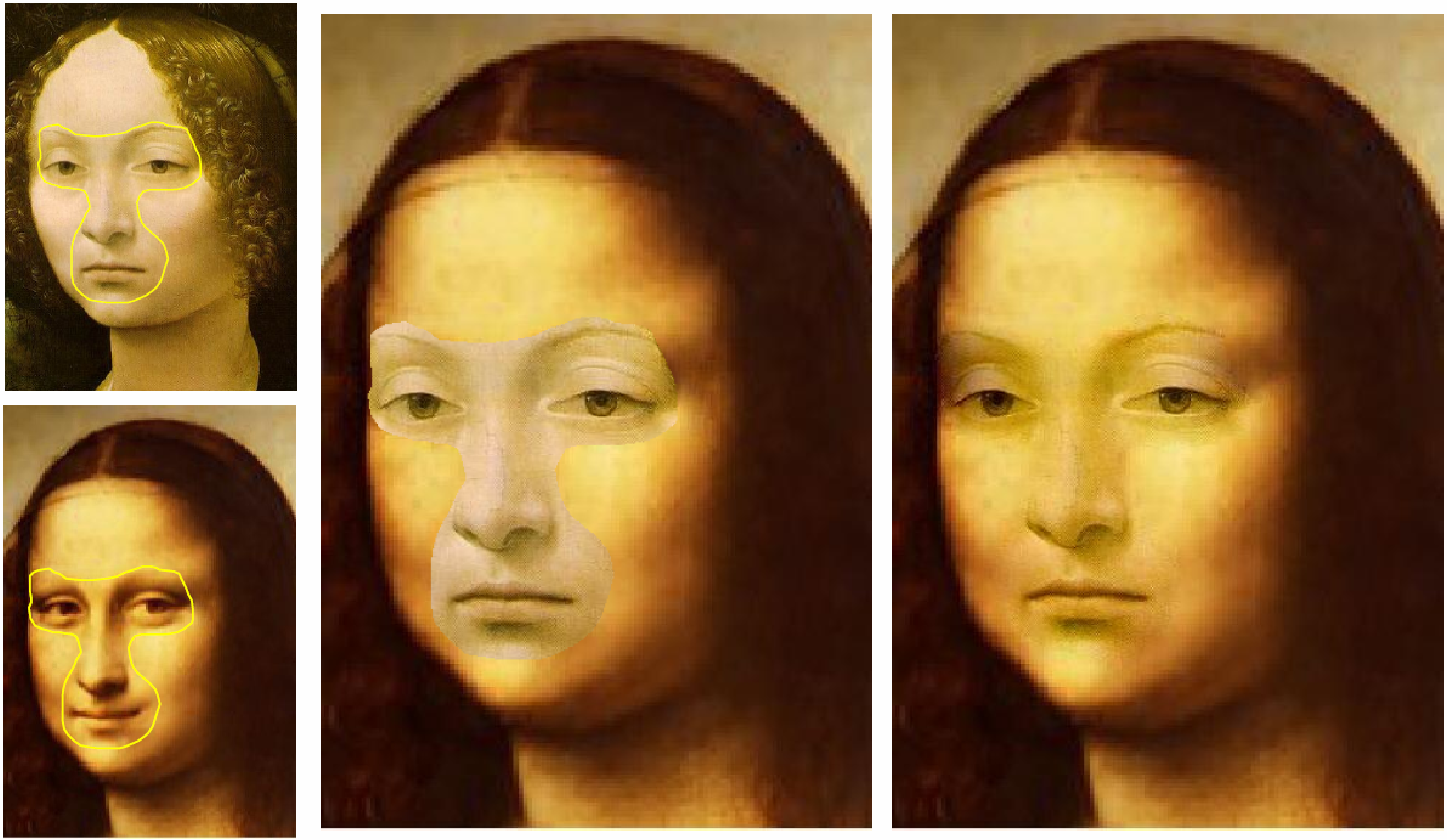
Solution: clone gradient



sources/destinations

seamless cloning

Result

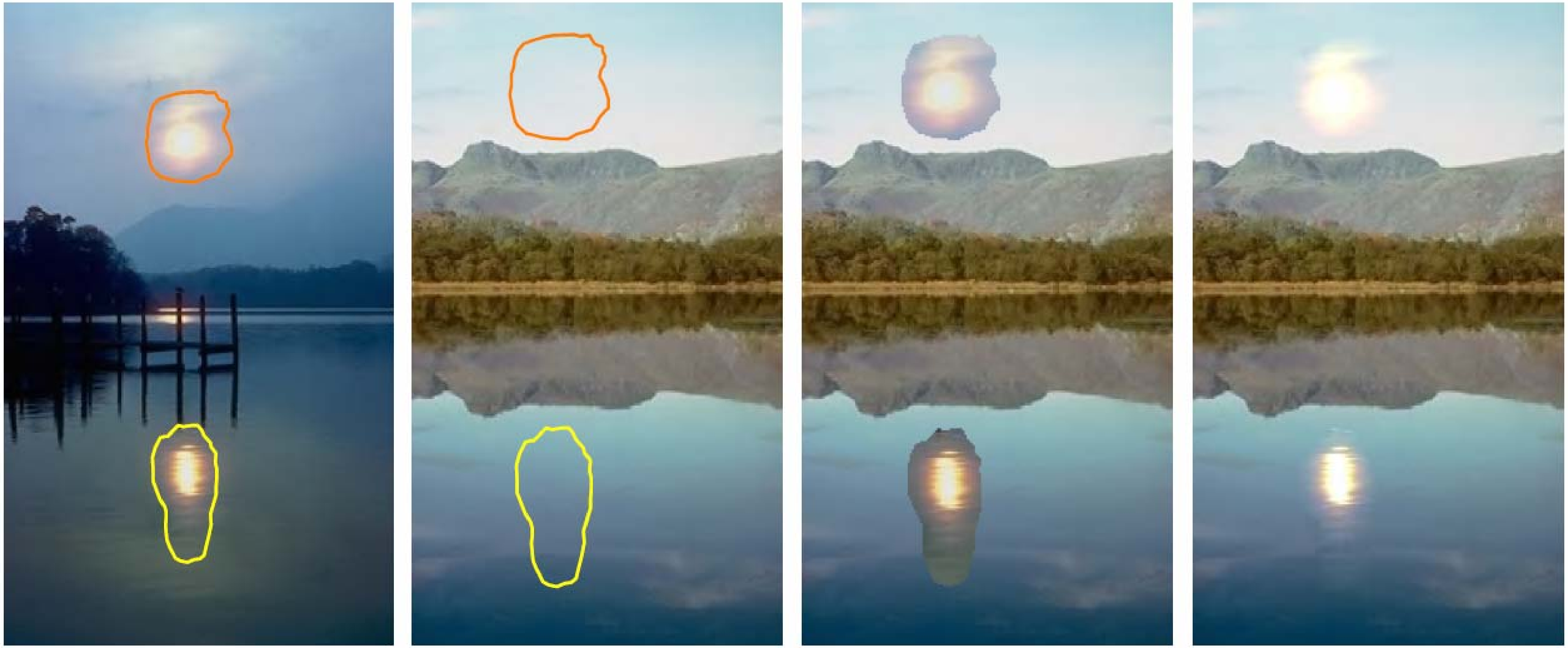


source/destination

cloning

seamless cloning

Figure 2: **Concealment**. By importing seamlessly a piece of the background, complete objects, parts of objects, and undesirable artifacts can easily be hidden. In both examples, multiple strokes (not shown) were used.



sources destinations cloning seamless cloning

swapped textures

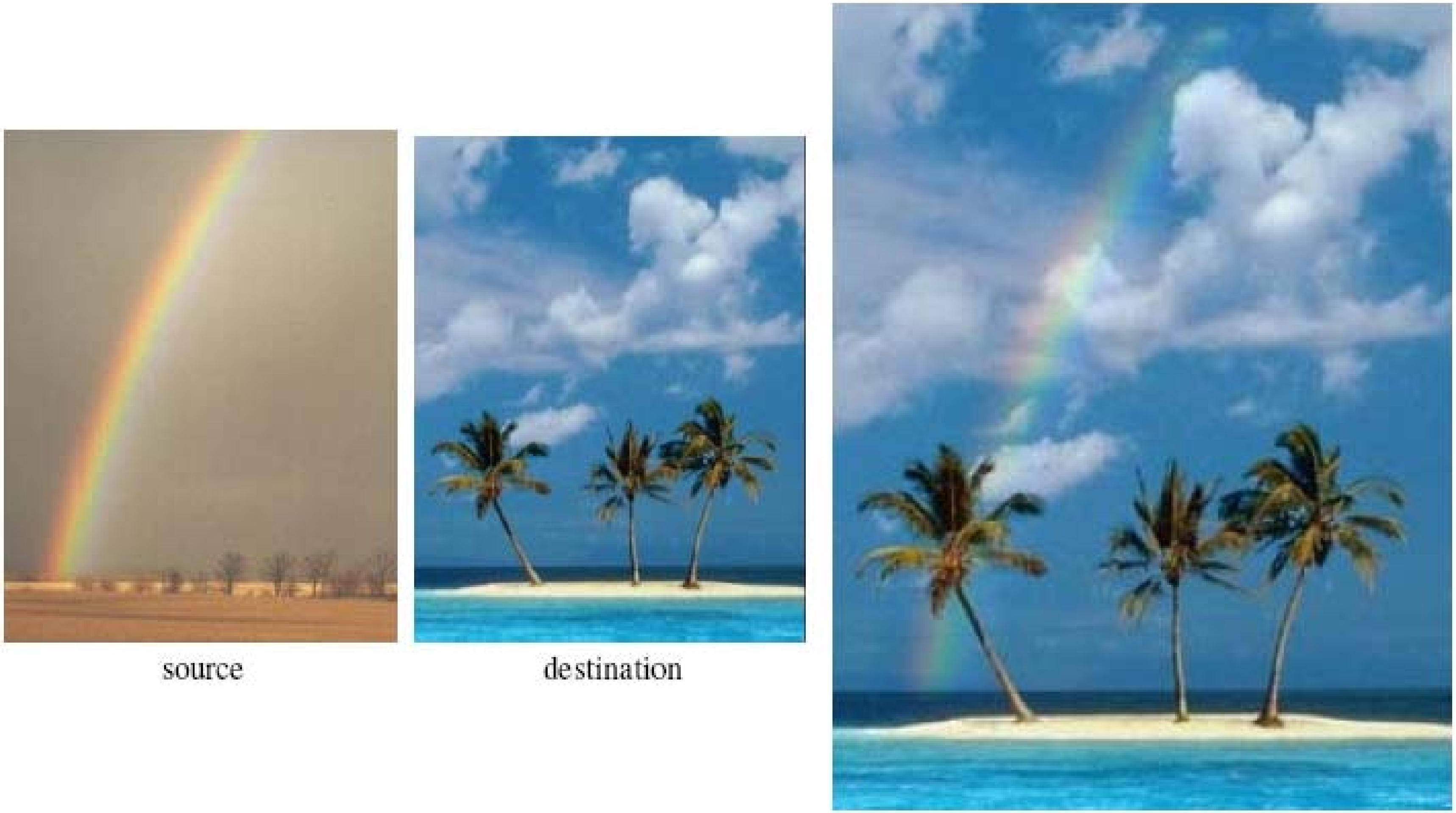


Figure 7: **Inserting transparent objects**. Mixed seamless cloning facilitates the transfer of partly transparent objects, such as the rainbow in this example. The non-linear mixing of gradient fields picks out whichever of source or destination structure is the more salient at each location.

Reduce big gradients

- Dynamic range compression
- Fattal et al. 2002

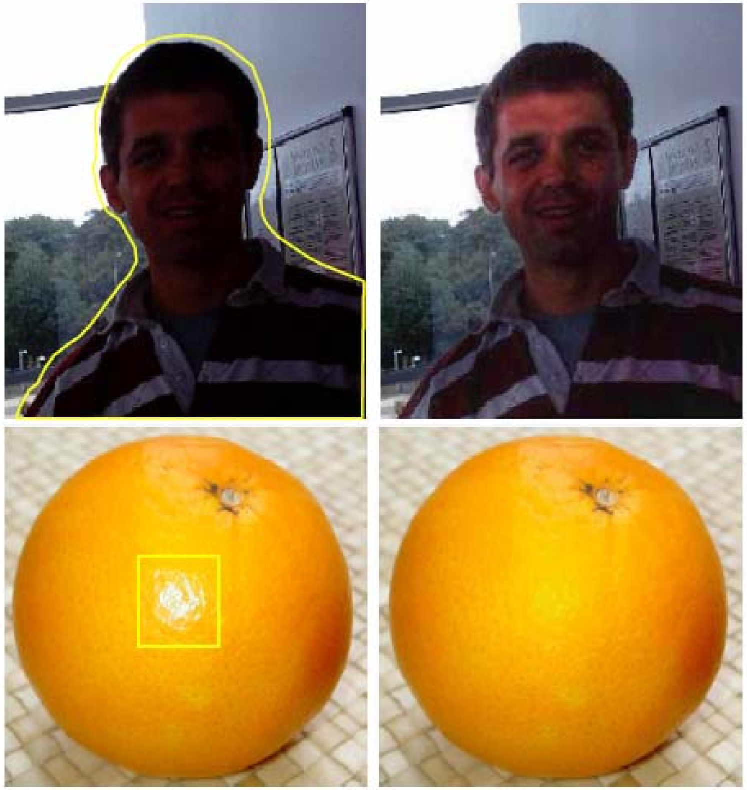


Figure 10: **Local illumination changes**. Applying an appropriate non-linear transformation to the gradient field inside the selection and then integrating back with a Poisson solver, modifies locally the apparent illumination of an image. This is useful to highlight under-exposed foreground objects or to reduce specular reflections.

Seamless Image Stitching in the Gradient Domain

- **Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss**
  http://www.cs.huji.ac.il/~alevin/papers/eccv04-blending.pdf
  http://eprints.pascal-network.org/archive/00001062/01/tips05-

ing)

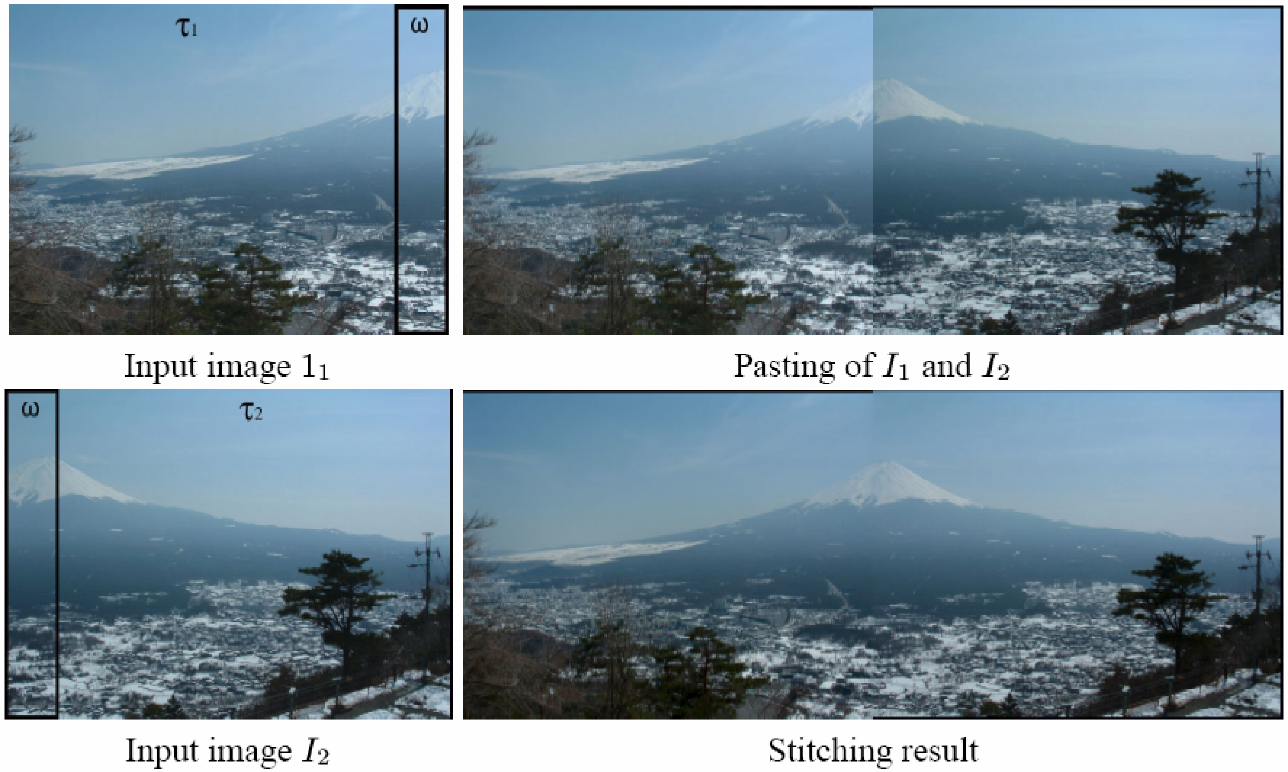
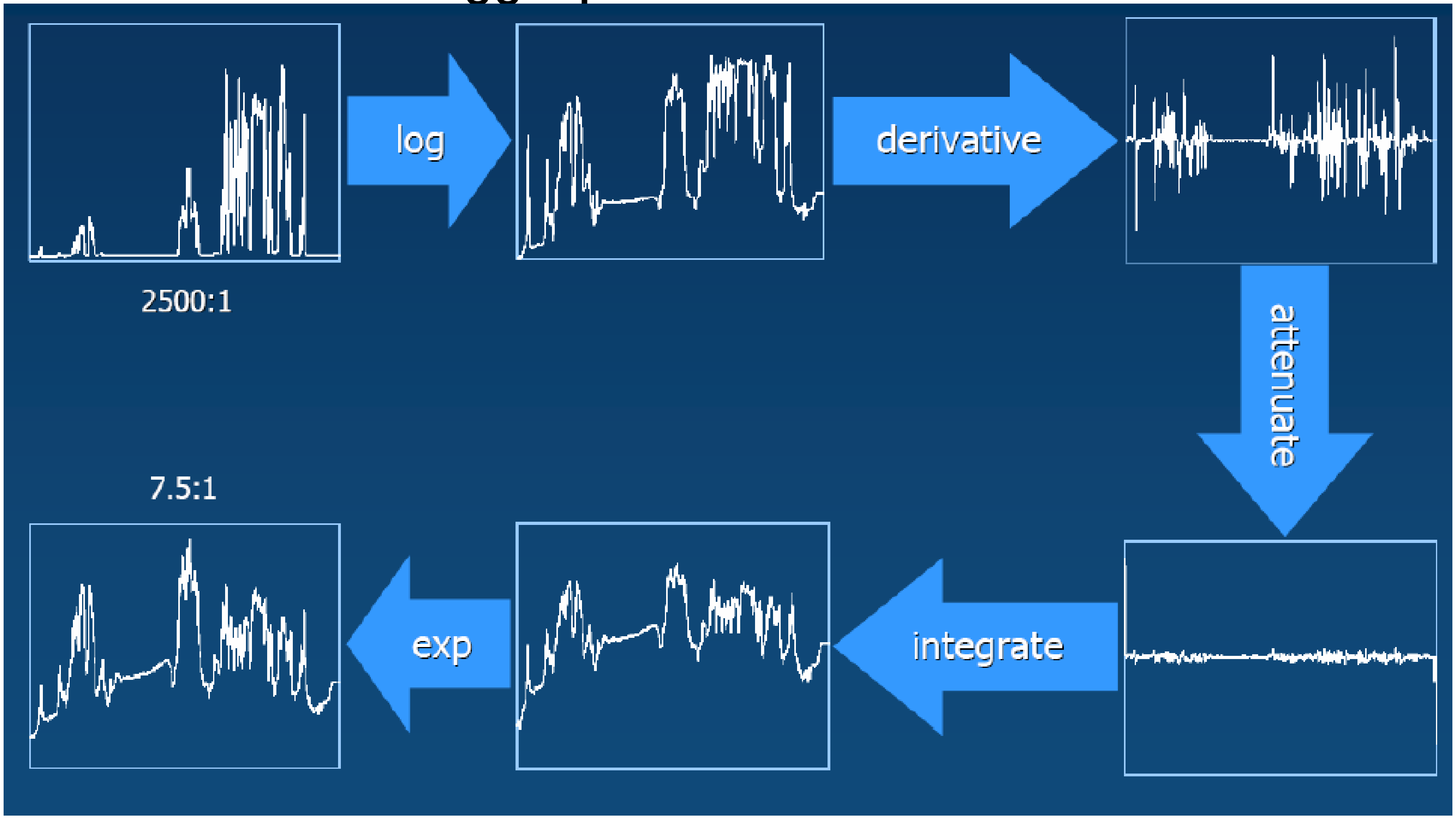


Fig. 1. Image stitching. On the left are the input images. ω is the overlap region. On top right is a simple pasting of the input images. On the bottom right is the result of the GIST1 algorithm.

Gradient tone mapping

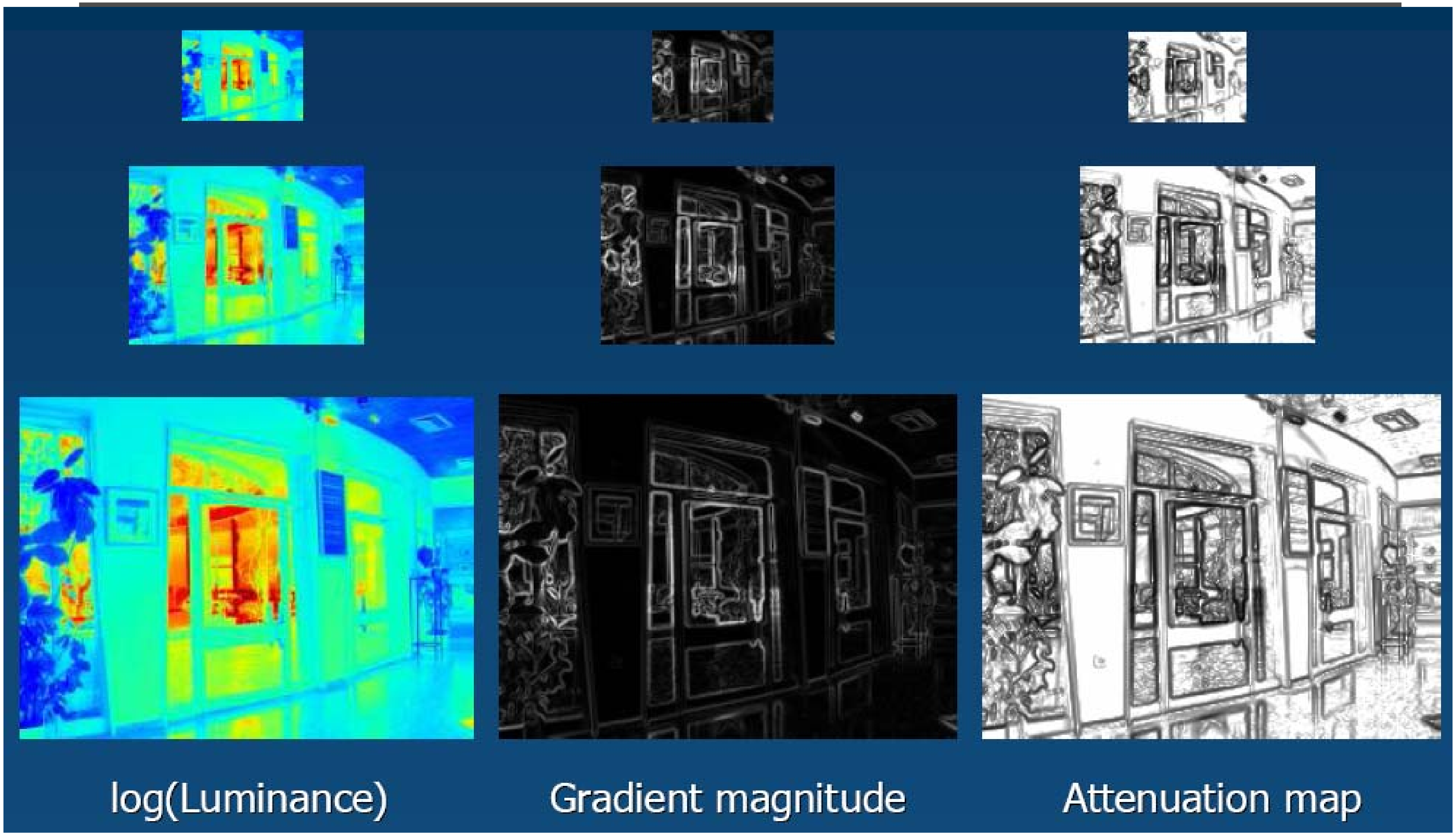
DigiVFX

- Fattal et al. Siggraph 2002



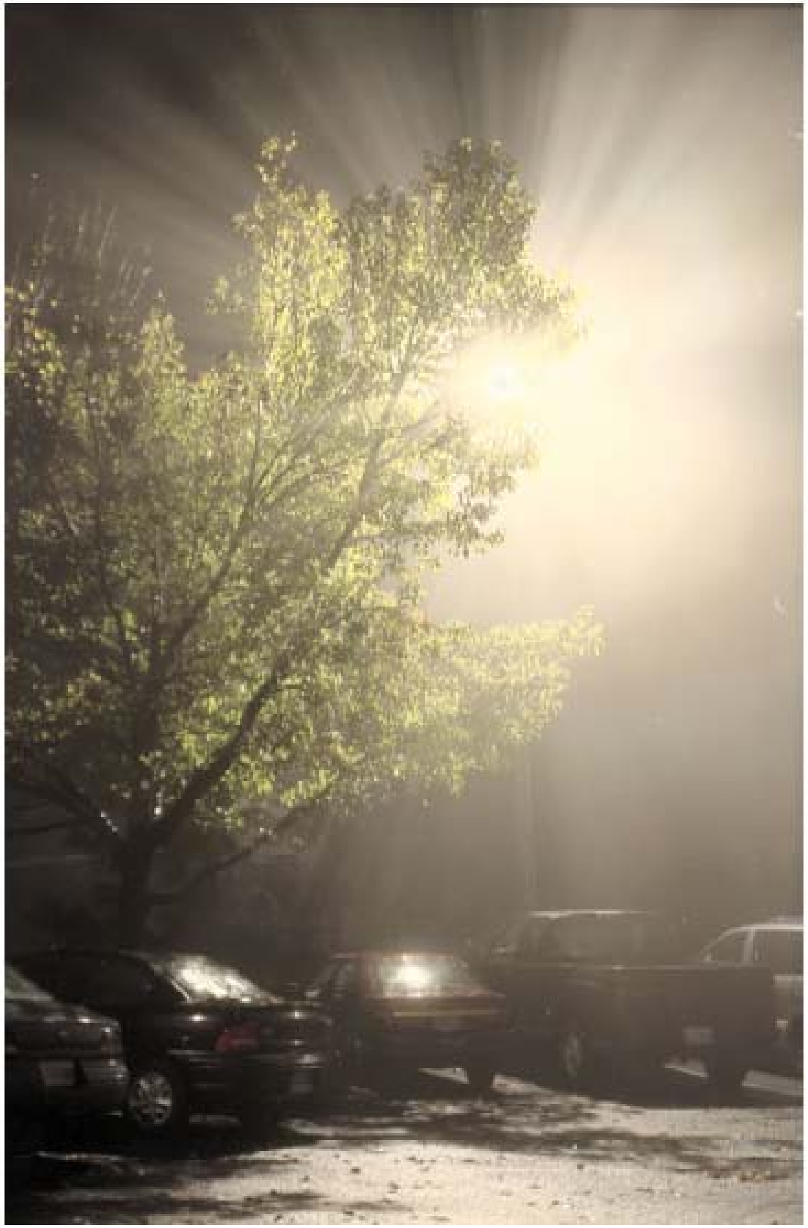
Slide from Siggraph 2005 by Raskar (Graphs by Fattal et al.)

Gradient attenuation



From Fattal et al.

Fattal et al. Gradient tone mapping



Poisson Matting

- Sun et al. Siggraph 2004
- Assume gradient of F & B is negligible
- Plus various image-editing tools to refine matte

$$I = \alpha F + (1 - \alpha) B$$

$$\nabla I = (F - B)\nabla\alpha + \alpha\nabla F + (1 - \alpha)\nabla B$$

$$\nabla\alpha \approx \frac{1}{F - B}\nabla I$$

Figure 1: Pulling of matte from a complex scene. From left to right: a complex natural image for existing matting techniques where the color background is complex, a high quality matte generated by Poisson matting, a composite image with the extracted koala and a constant-color background, and a composite image with the extracted koala and a different background.

Interactive Local Adjustment of Tonal Values

Dani Lischinski, Zeev Farbman
The Hebrew University

Matt Uyttendaele, Richard Szeliski
Microsoft Research

Background (1)

Darkroom

Camera shutter ---→ Photograph

Tool { Dodging / Burning brushes } Only!

But, …

It is tedious, time-consuming and painstaking!

Background (2)

[Adobe Photoshop CS2, 2005]

- A large arsenal of adjustment tools
- Hard to master these tools
  - To learn, use
- Tedious and time-consuming
  - Professional ability, experienced skill
  - Too many layer masks
- Incapable in some requirements

Background (2)

[Adobe Photoshop CS2, 2005]

Original image Layer mask Result

Related Work: Tone Mapping Operators

DigiVFX

- Global operators

  [Ward Larson et al. 1997; Reinhard et al. 2002; Drago et al. 2003]

  - Usually fast
- Local operators

  [Fattal et al. 2002; Reinhard et al. 2002;

  Li et al. 2005] …

  - Better at preserving local contrasts
  - Introduce visual artifacts sometimes

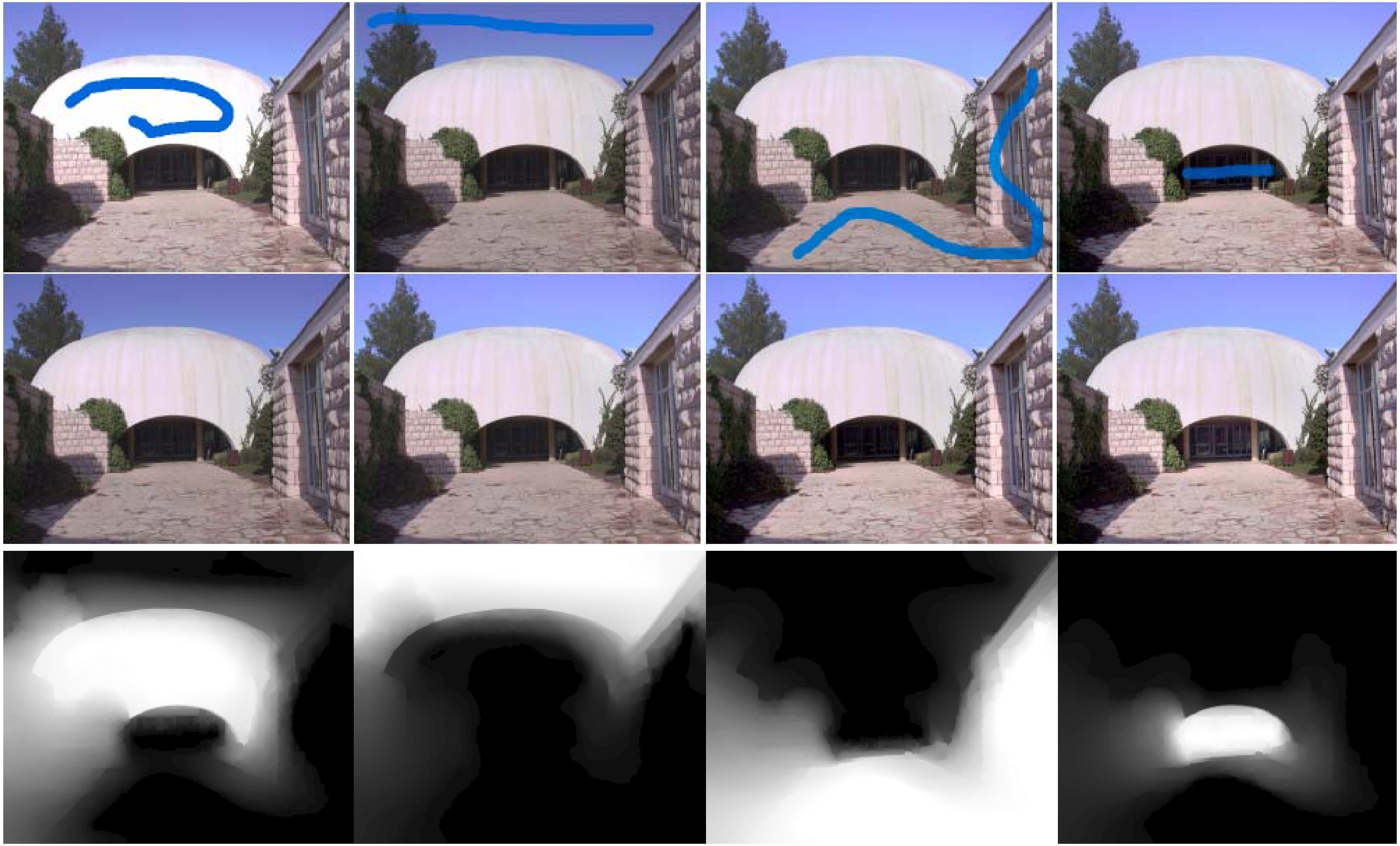
Limitations of Tone Mapping Operators

- Lack of direct local control
  - Can't directly manipulate a particular region
- Not guaranteed to converge to a subjectively satisfactory result
  - Involves several trial-and-error iterations
  - Change the entire image each iteration

Algorithm Overview

1. Load a digital negative, a camera RAW file, an HDR radiance map, or an ordinary image
2. Indicate regions in the image that require adjusting
3. Experiment with the available adjustment parameters until a satisfactory result is obtained in the desired regions
4. Iterate 2 and 3 until a satisfactory image


An Example



Region Selection: Strokes and Brushes

- Basic brush
- Luminance brush

weight=1, for the selected pixels in the brush;
weight=0, else

Region Selection: Luminance Brush

μ be the mean lightness (CIE L^*)

A pixel with a lightness value of ℓ is selected only if $|\mu - \ell| < \sigma$

the weight

$$w(\ell) = \exp(-|\ell - \mu|^2/\sigma^2)$$

Region Selection: Strokes and Brushes

- Basic brush
- Luminance brush
- Lumachrome brush (chromaticity)
  - the CIE $L^*a^*b^*$ color space
- Over-exposure brush
- Under-exposure brush

Constraint Propagation

User strokes　　　　Adjusted exposure

Image-guided Energy Minimization

$$f = \arg\min_{f} \left\{ \sum_{\mathbf{x}} w(\mathbf{x}) \left(f(\mathbf{x}) - g(\mathbf{x}) \right)^2 + \lambda \sum_{\mathbf{x}} h(\nabla f, \nabla L) \right\}$$

Data term + smoothing term

Image-guided Energy Minimization

$$f = \arg\min_{f} \left\{ \sum_{\mathbf{x}} w(\mathbf{x}) \left(f(\mathbf{x}) - g(\mathbf{x}) \right)^2 + \lambda \sum_{\mathbf{x}} h(\nabla f, \nabla L) \right\}$$

data term + smoothing term

$$h(\nabla f, \nabla L) = \frac{|f_x|^2}{|L_x|^\alpha + \varepsilon} + \frac{|f_y|^2}{|L_y|^\alpha + \varepsilon}$$

L : log-luminance channel
α : sensitivity factor
ε : a small zero-division constant
λ : a balance factor

Default:
$\alpha = 1$
$\varepsilon = 0.0001$
$\lambda = 0.2$

Standard Finite Differences

$$f = \arg\min_f \left\{ \sum_{\mathbf{x}} w(\mathbf{x}) \, (f(\mathbf{x}) - g(\mathbf{x}))^2 + \lambda \sum_{\mathbf{x}} h(\nabla f, \nabla L) \right\}$$

$$\mathbf{A} f = b,$$

where $\mathbf{A}_{ij} = \begin{cases} -\lambda \left(\left| L_i - L_j \right|^\alpha + \varepsilon \right)^{-1} & j \in N_4(i) \\ w_i - \sum_{k \in N_4(i)} \mathbf{A}_{ik} & i = j \\ 0 & \text{otherwise} \end{cases}$

and $b_i = w_i \, g_i.$

$N_4(i)$ are the 4-neighbors of pixel i

Fast Approximate Solution

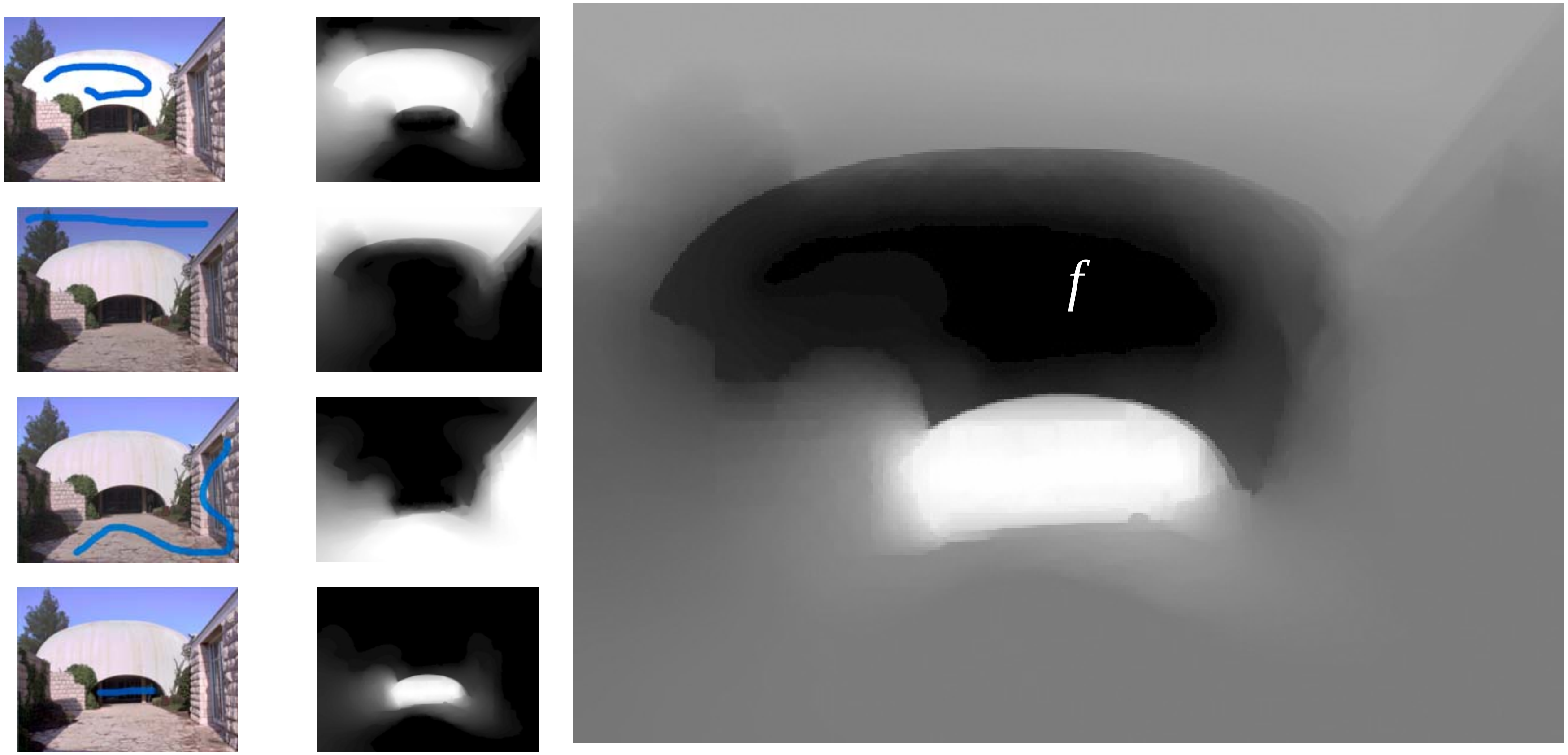
$$\mathbf{A}f = b$$

Solved iteratively by [Saad 2003]

preconditioned conjugate gradients (PCG)

Interactive Local Adjustment of Tonal Value

$$f = \arg\min_{f} \left\{ \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}) - g(\mathbf{x}))^2 + \lambda \sum_{\mathbf{x}} h(\nabla f, \nabla L) \right\}$$



Results

SIGGRAPH 2006

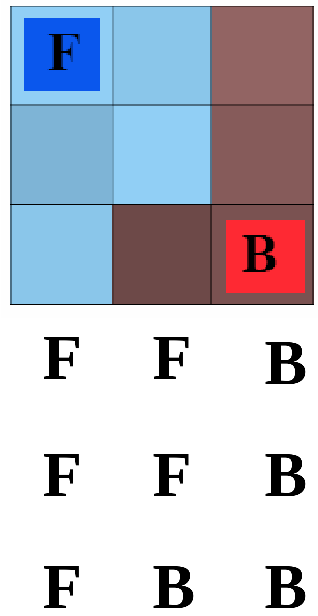
Interactive Local Adjustment of Tonal Values

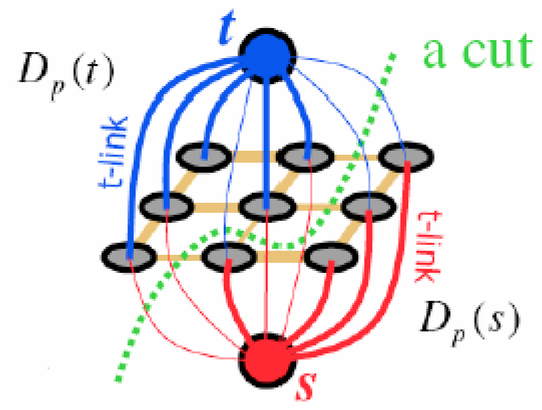
Dani Lischinski
Zeev Farbman
Matt Uyttendaele
Richard Szeliski

Graph cut

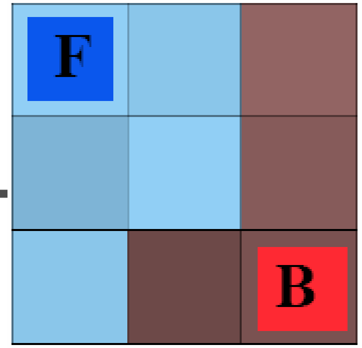
Graph cut

- Interactive image segmentation using graph cut
- Binary label: foreground vs. background
- User labels some pixels
  - similar to trimap, usually sparser
- Exploit
  - Statistics of known Fg & Bg
  - Smoothness of label
- Turn into discrete graph optimization
  - Graph cut (min cut / max flow)



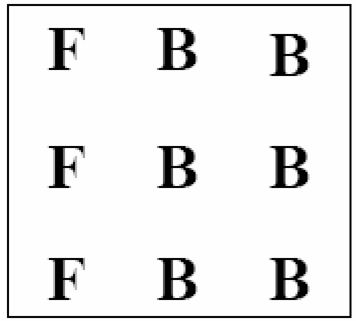


Energy function

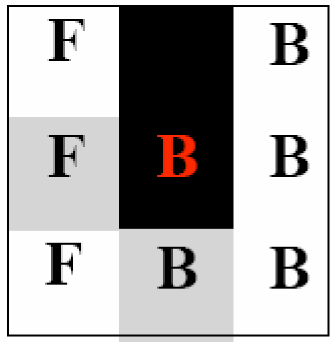


- Labeling: one value per pixel, F or B
- Energy(labeling) = data + smoothness
  - Very general situation
  - Will be minimized
- Data: for each pixel
  - Probability that this color belongs to F (resp. B)
  - Similar in spirit to Bayesian matting
- Smoothness (aka regularization): per neighboring pixel pair
  - Penalty for having different label
  - Penalty is downweighted if the two pixel colors are very different
  - Similar in spirit to bilateral filter

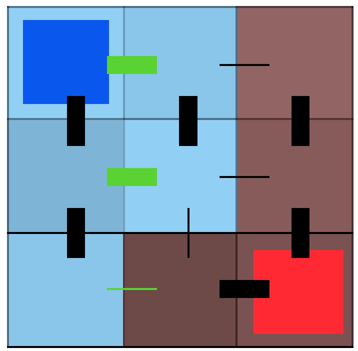
One labeling (ok, not best)



Data

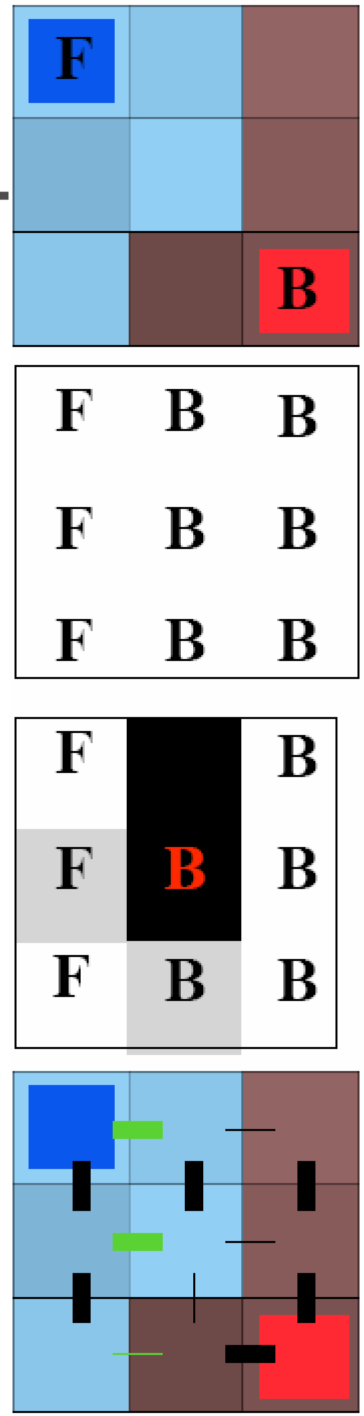


Smoothness



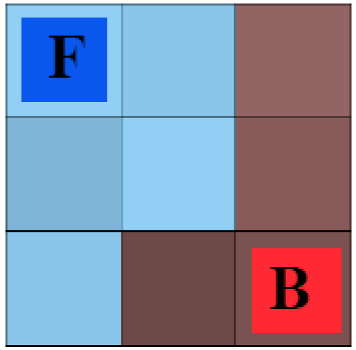
Data term

- A.k.a regional term
  (because integrated over full region)
- $D(L)=\sum_i -\log h[L_i](C_i)$
- Where *i* is a pixel
  L_i is the label at *i* (F or B),
  C_i is the pixel value
  $h[L_i]$ is the histogram of the observed Fg (resp Bg)
- Note the minus sign



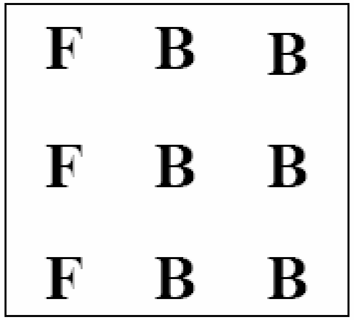
Hard constraints

- The user has provided some labels
- The quick and dirty way to include constraints into optimization is to replace the data term by a huge penalty if not respected.
- D(L_i)=0 if respected
- D(L_i) = K if not respected
  - e.g. K=- #pixels

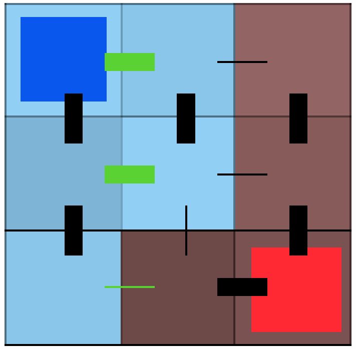


Smoothness term

- a.k.a boundary term, a.k.a. regularization
- $S(L)=\sum_{\{j,\ i\}\ \mathbf{in}\ N} B(C_i,C_j)\ \delta(L_i-L_j)$
- Where i,j are neighbors
  - e.g. 8-neighborhood (but I show 4 for simplicity)

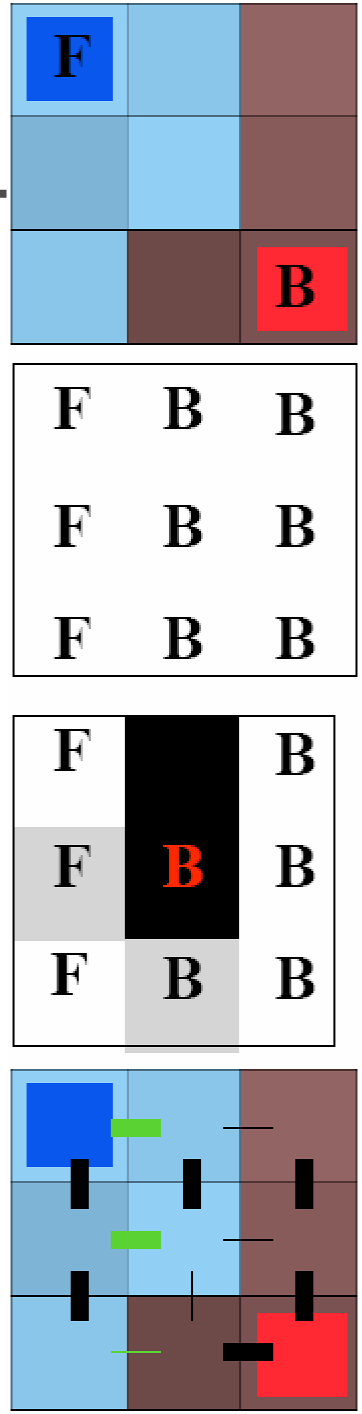


- $\delta(L_i-L_j)$ is 0 if $L_i=L_j$, 1 otherwise
- $B(C_i,C_j)$ is high when C_i and C_j are similar, low if there is a discontinuity between those two pixels
  - e.g. $\exp(-||C_i-Cj||^2/2\sigma^2)$
  - where σ can be a constant or the local variance
- Note positive sign



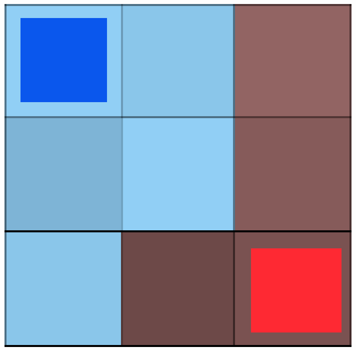
Optimization

- E(L)=D(L)+λ S(L)
- λ is a black-magic constant
- Find the labeling that minimizes E
- In this case, how many possibilities?
  - 2^9 (512)
  - We can try them all!
  - What about megapixel images?

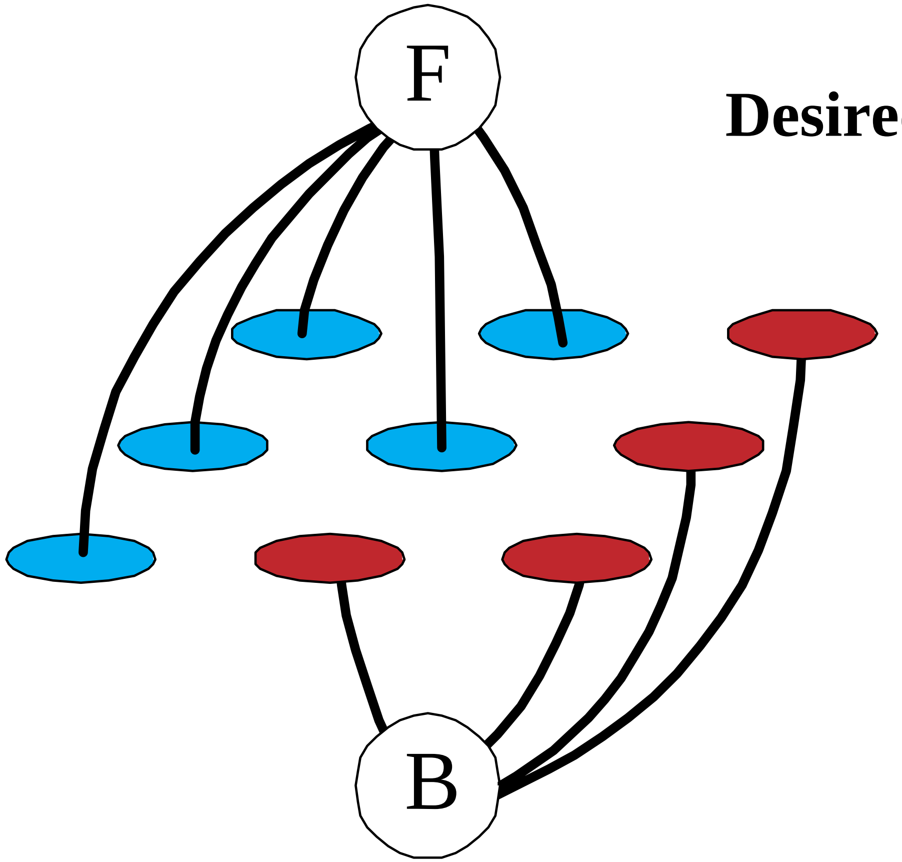


Labeling as a graph problem

- Each pixel = node
- Add two nodes F & B
- Labeling: link each pixel to either F or B



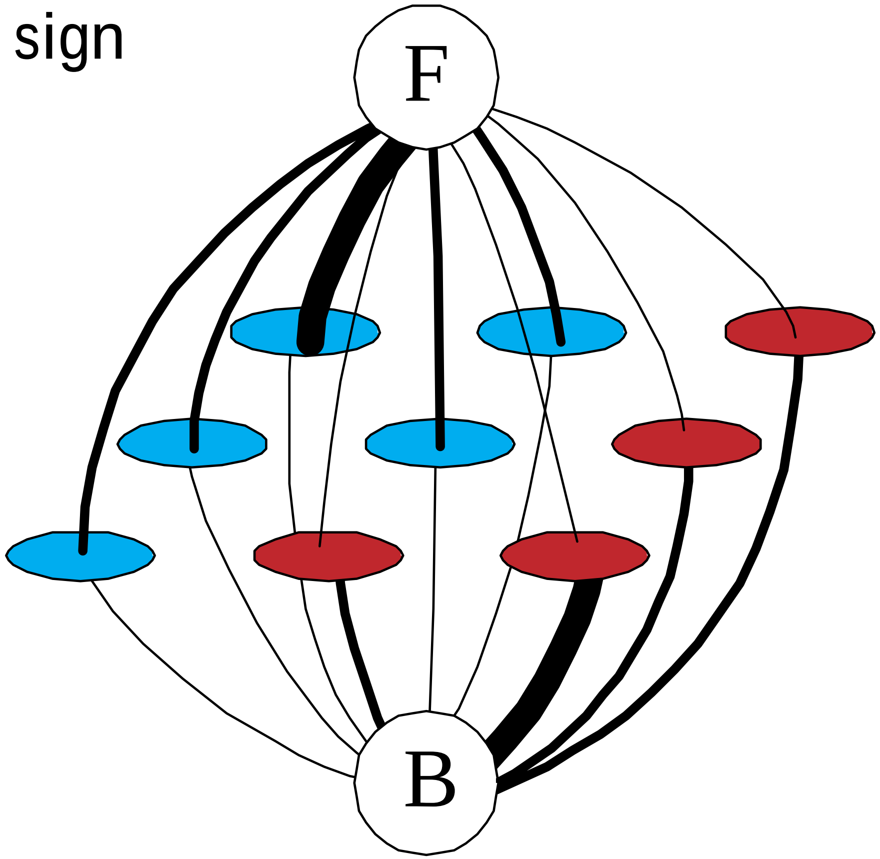
F	F	B
F	F	B
F	B	B



Desired result

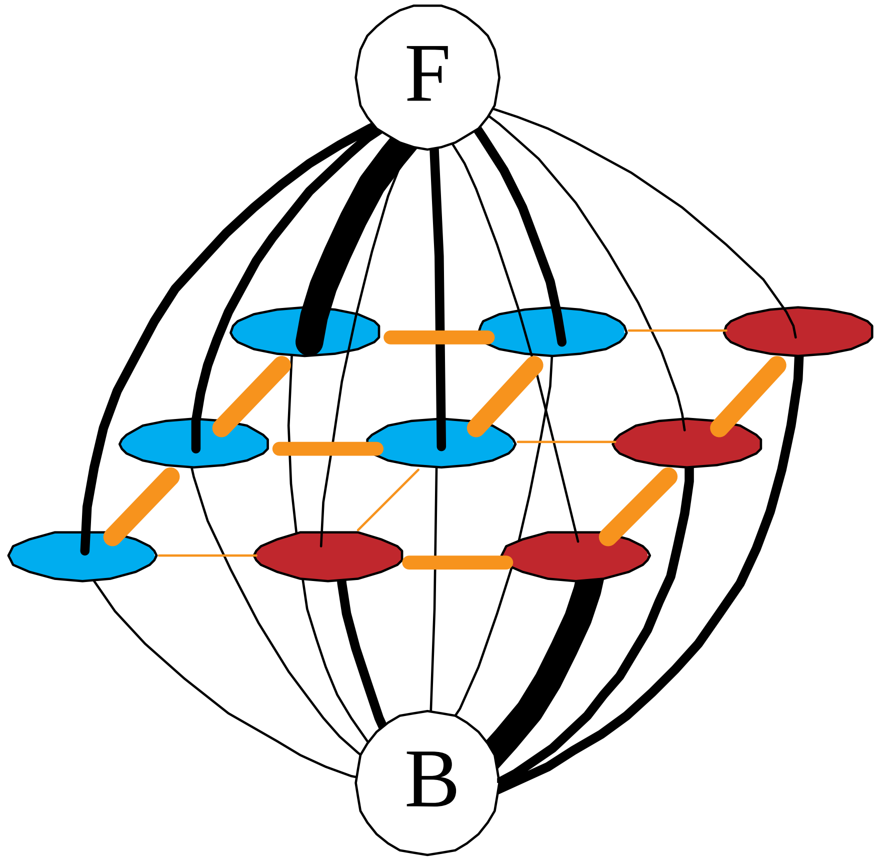
Data term

- Put one edge between each pixel and F & G
- Weight of edge = minus data term
  - Don’t forget huge weight for hard constraints
  - Careful with sign



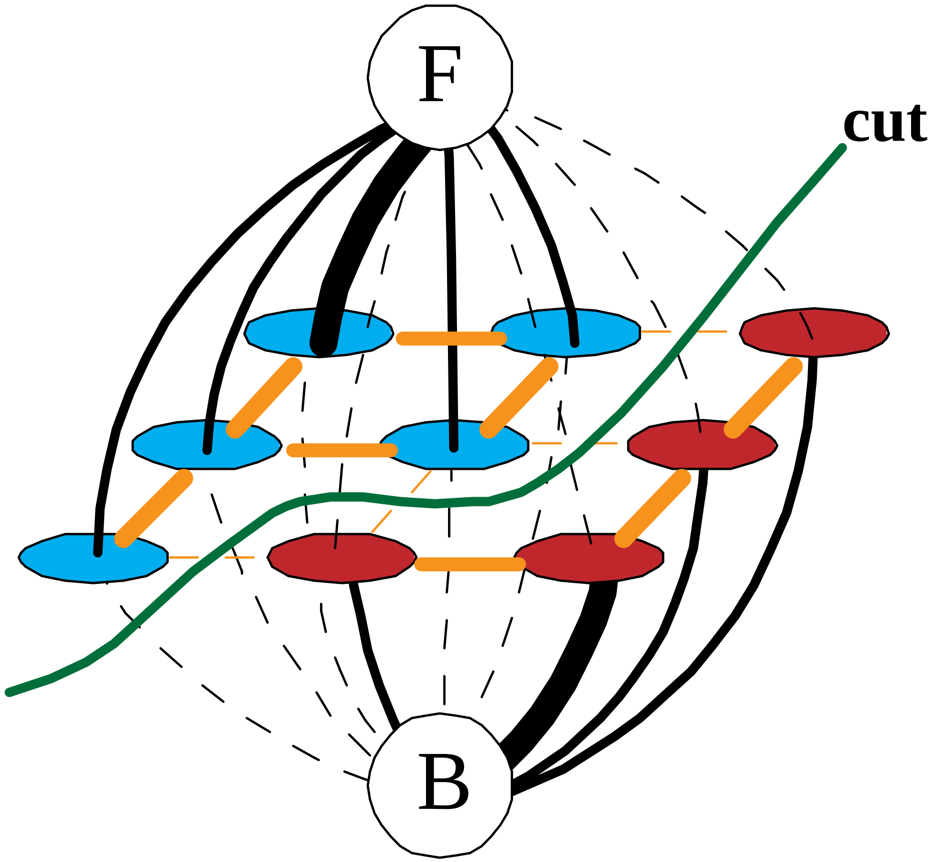
Smoothness term

- Add an edge between each neighbor pair
- Weight = smoothness term



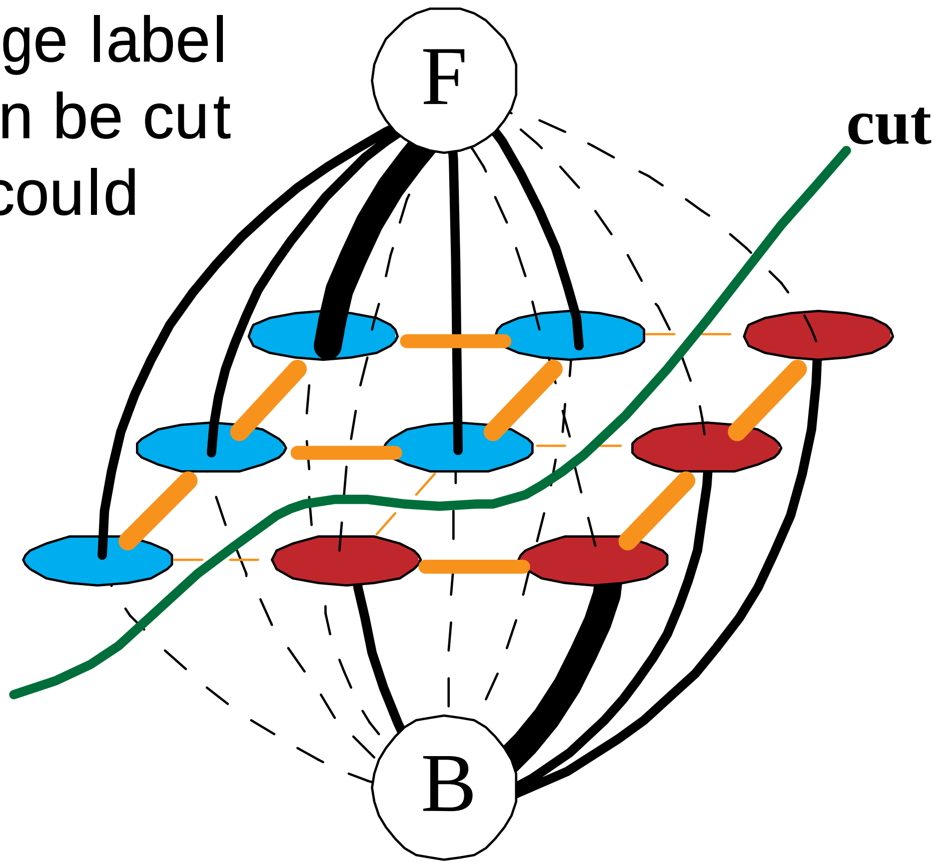
Min cut

- Energy optimization equivalent to min cut
- Cut: remove edges to disconnect F from B
- Minimum: minimize sum of cut edge weight



Min cut <=> labeling

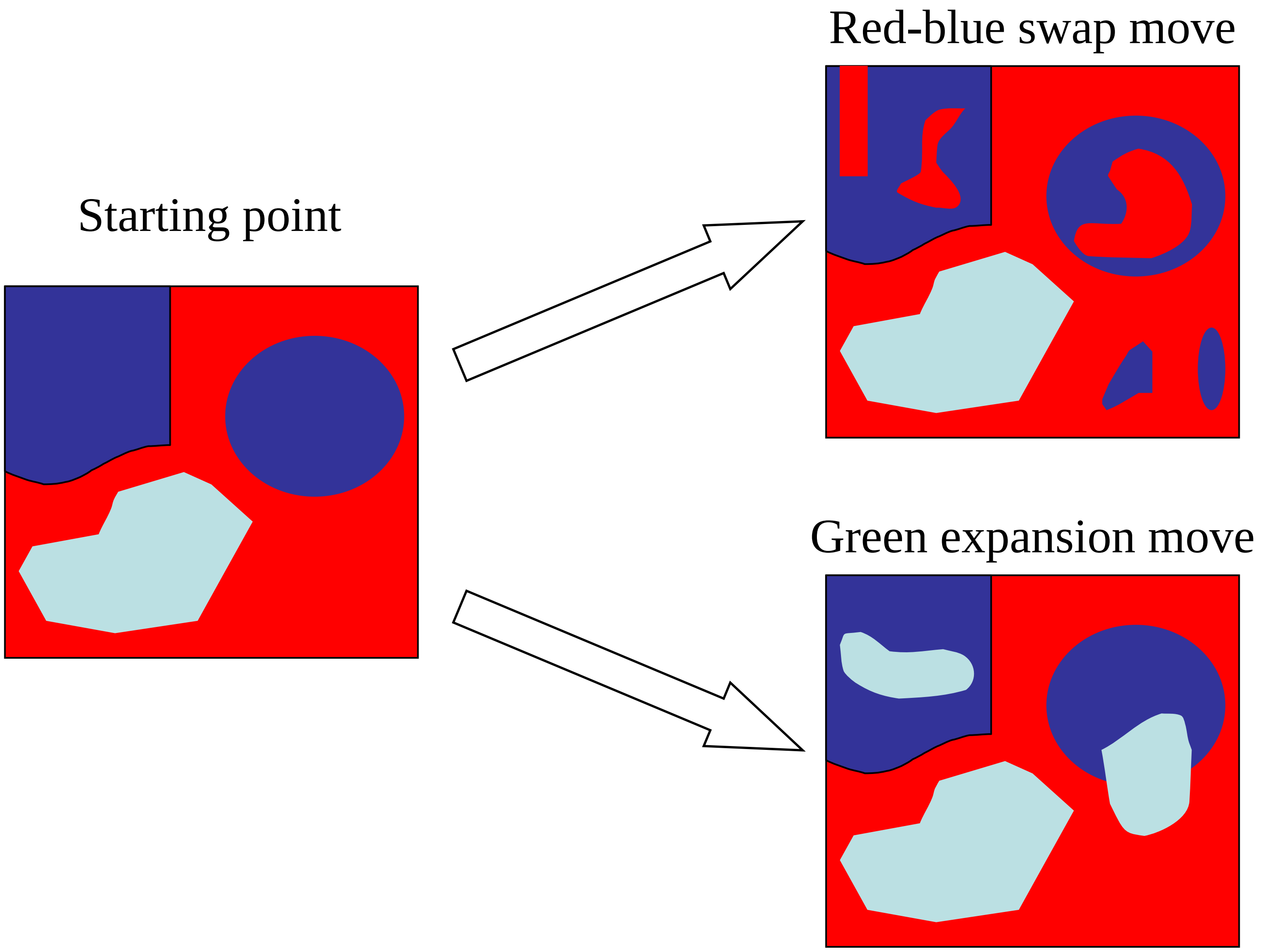
- In order to be a cut:
  - For each pixel, either the F or G edge has to be cut
- In order to be minimal
  - Only one edge label per pixel can be cut (otherwise could be added)



Computing a multiway cut

- With 2 labels: classical min-cut problem
  - Solvable by standard flow algorithms
    - polynomial time in theory, nearly linear in practice
  - More than 2 terminals: NP-hard
    [Dahlhaus *et al.*, STOC '92]
- Efficient approximation algorithms exist
  - Within a factor of 2 of optimal
  - Computes local minimum in a strong sense
    - even very large moves will not improve the energy
  - Yuri Boykov, Olga Veksler and Ramin Zabih, Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

Move examples

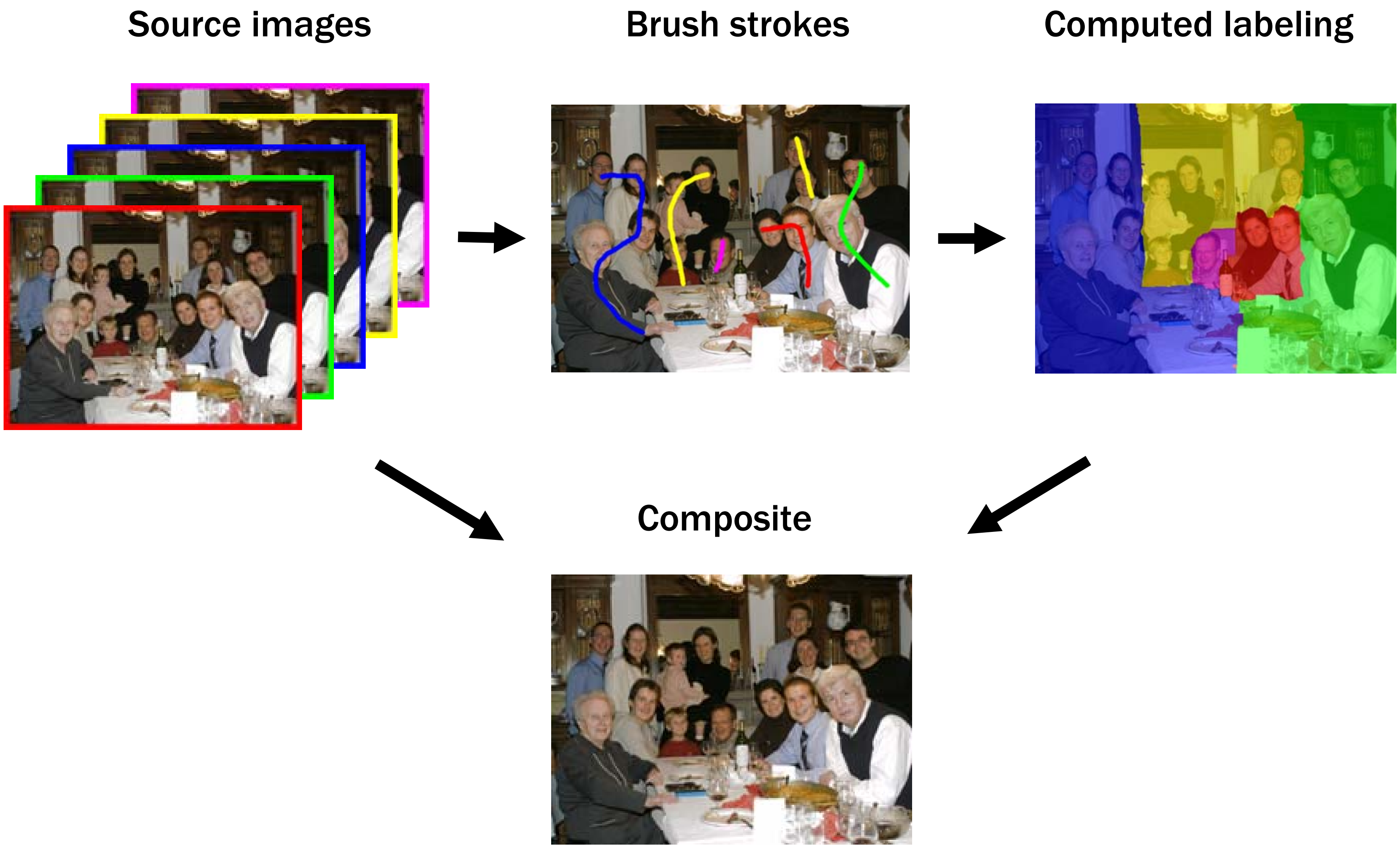


GrabCut
Interactive Foreground Extraction using Iterated Graph Cuts

Carsten Rother
Vladimir Kolmogorov
Andrew Blake

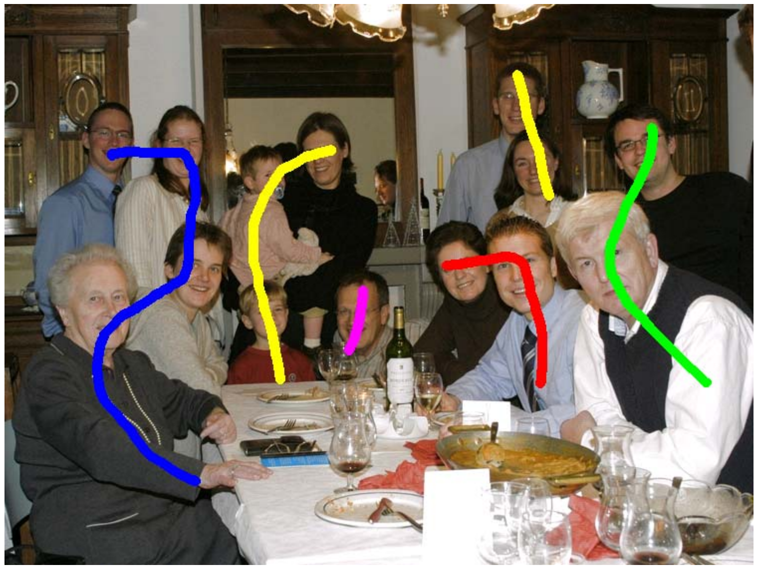
Microsoft Research Cambridge-UK

Agrawala et al, Digital Photomontage, Siggraph 2004


Source images
Brush strokes
Computed labeling
Composite

Graph Cuts for Segmentation and Mosaicing

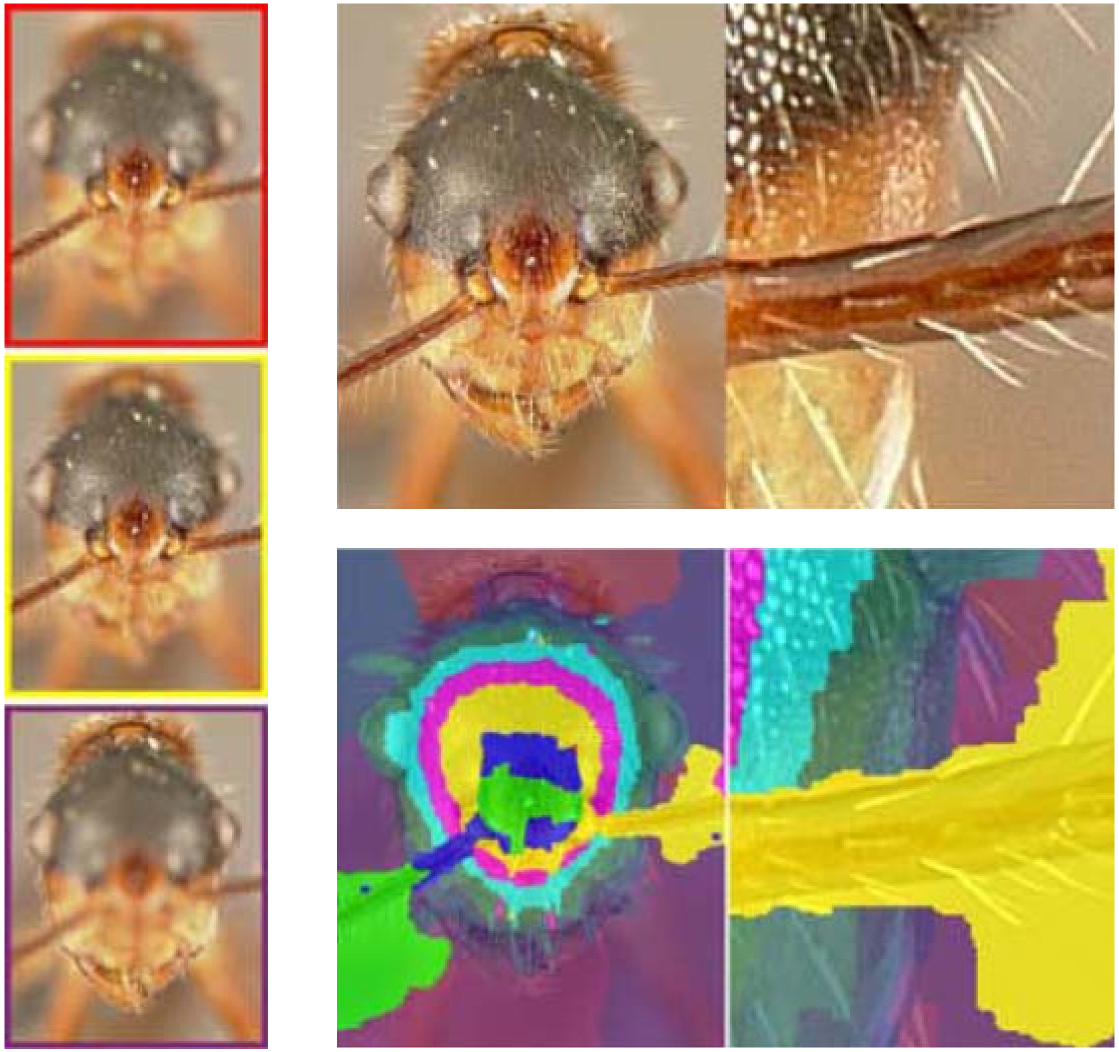
Brush strokes



Computed labeling

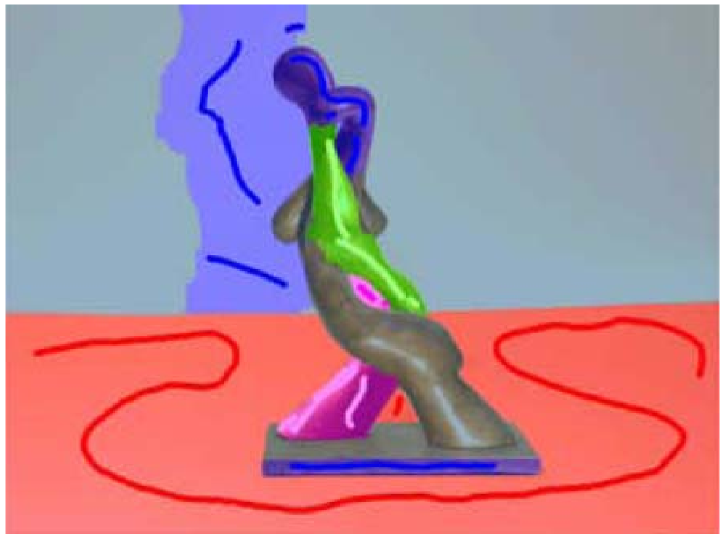
Interactive Digital Photomontage

- Extended depth of field



Interactive Digital Photomontage

- Relighting



Interactive Digital Photomontage

Bilateral filtering

[Ben Weiss, Siggraph 2006]

Image Denoising

noisy image

naïve denoising
Gaussian blur

better denoising
edge-preserving filter

Smoothing an image without blurring its edges.

A Wide Range of Options

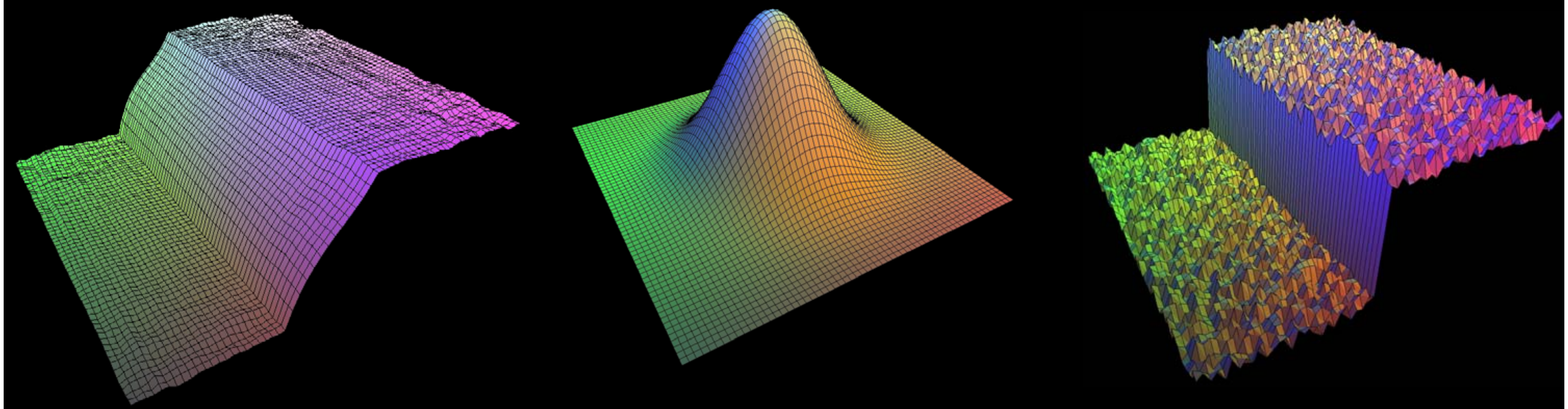
- Diffusion, Bayesian, Wavelets…
  - All have their pros and cons.

- Bilateral filter
  - not always the best result [Buades 05] but often good
  - easy to understand, adapt and set up

Start with Gaussian filtering

- Here, input is a step function + noise

$$J = f \otimes I$$

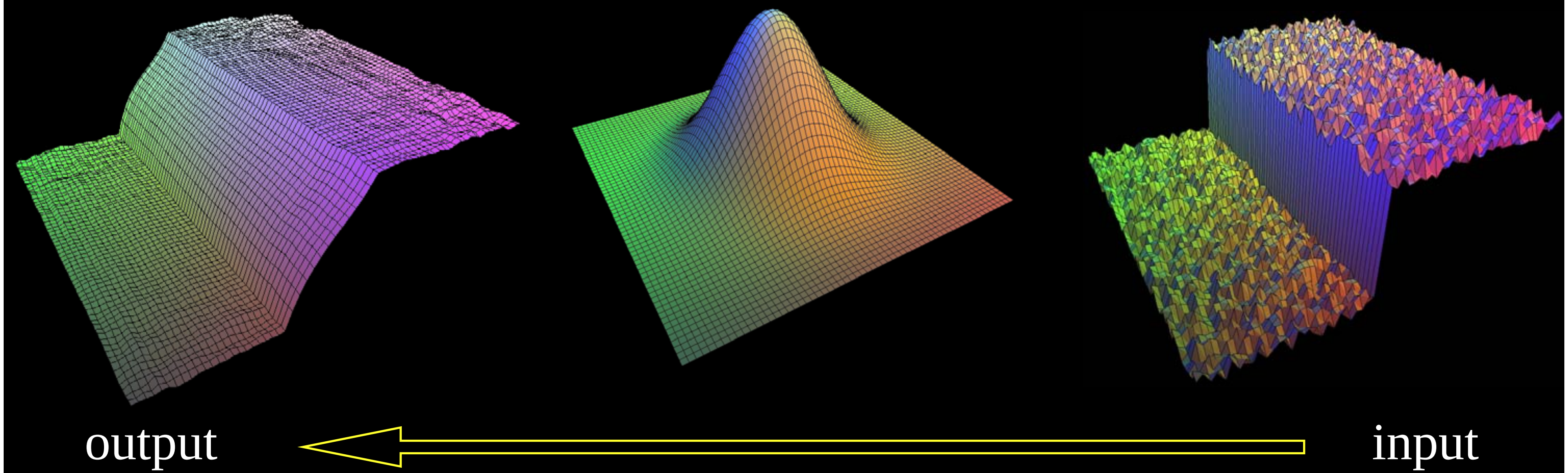


output ← input

Start with Gaussian filtering

- Spatial Gaussian f

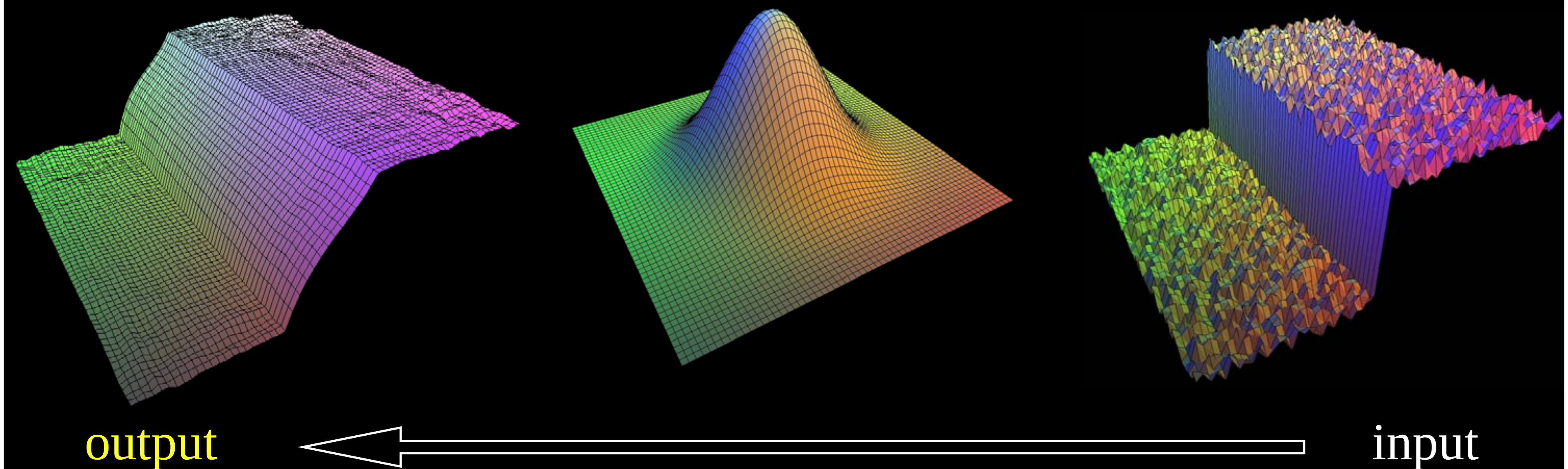
$$J = f \otimes I$$



Start with Gaussian filtering

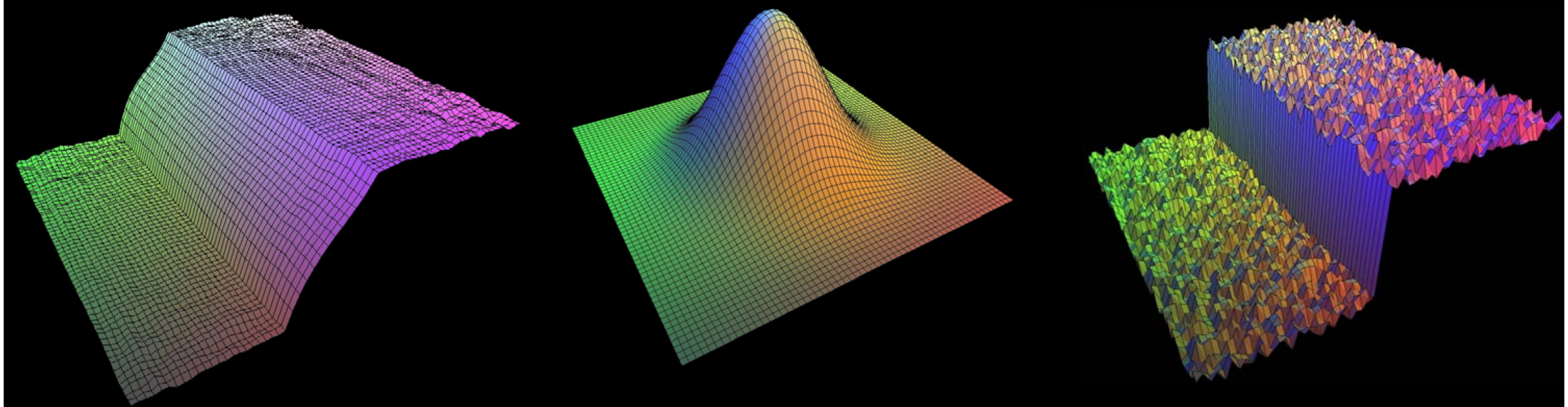
- Output is blurred

$$J = f \otimes I$$



Gaussian filter as weighted average

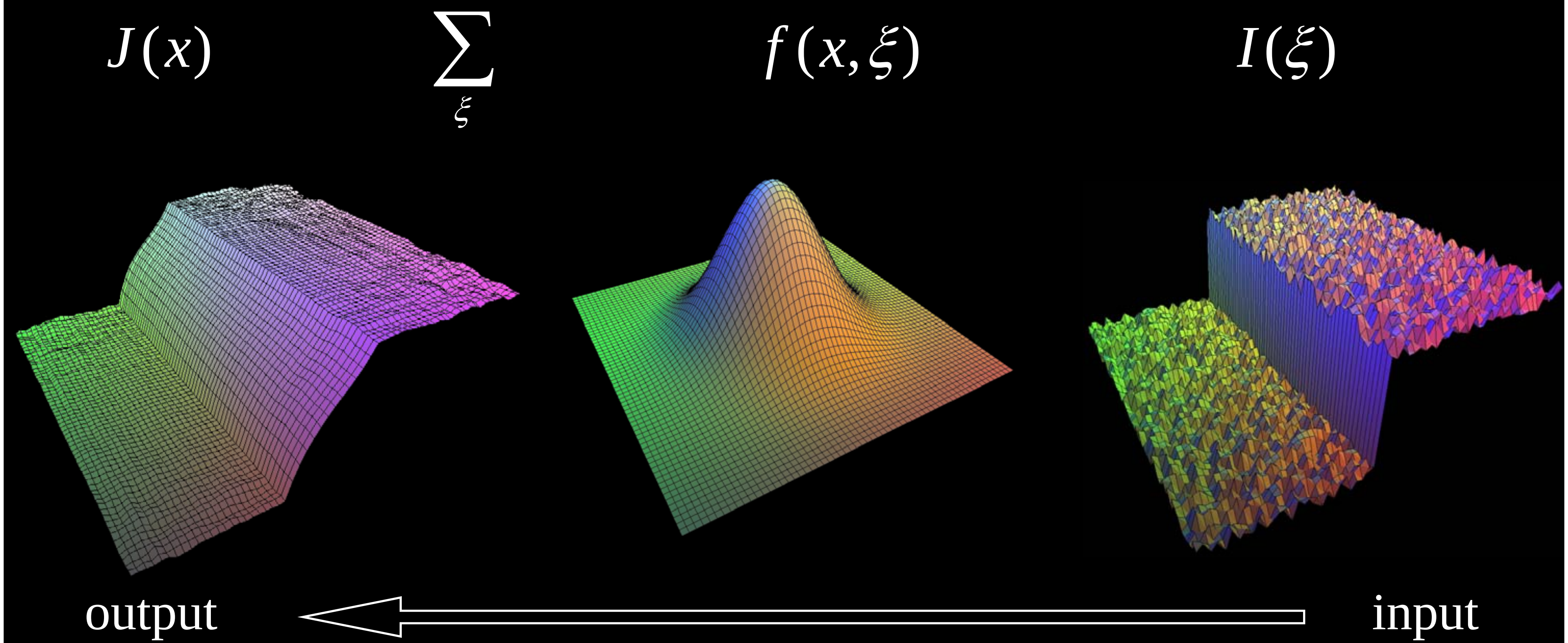
$$J(x) \quad \sum_{\xi} \quad f(x,\xi) \quad I(\xi)$$



output ⟵ input

The problem of edges

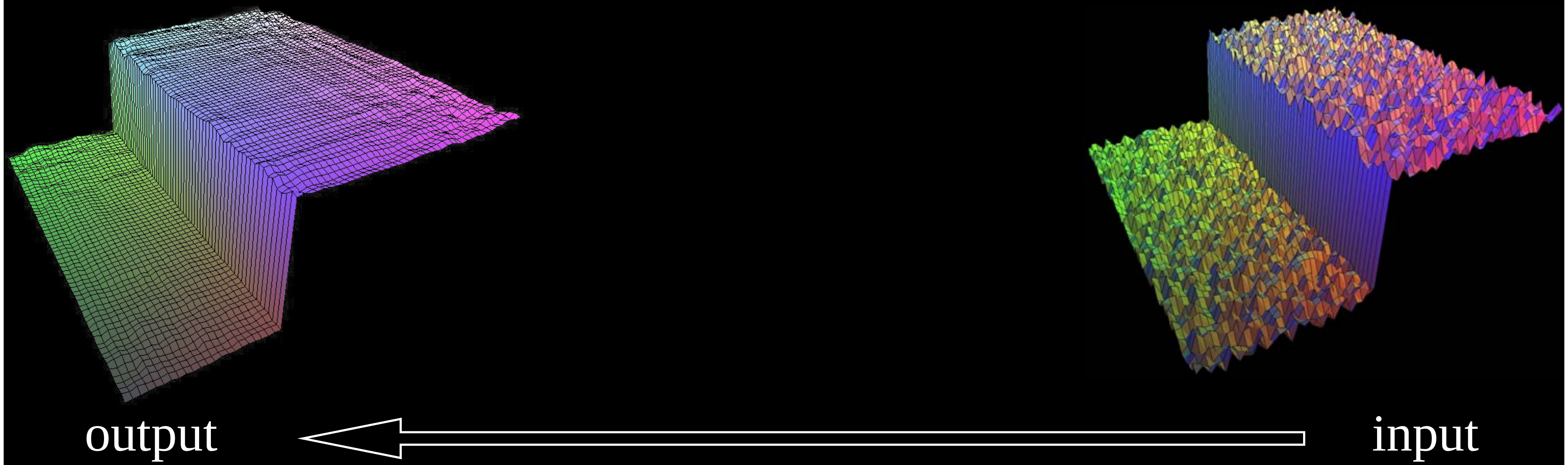
- Here, $I(\xi)$ "pollutes" our estimate J(x)
- It is too different



Principle of Bilateral filtering

- [Tomasi and Manduchi 1998]
- Penalty g on the intensity difference

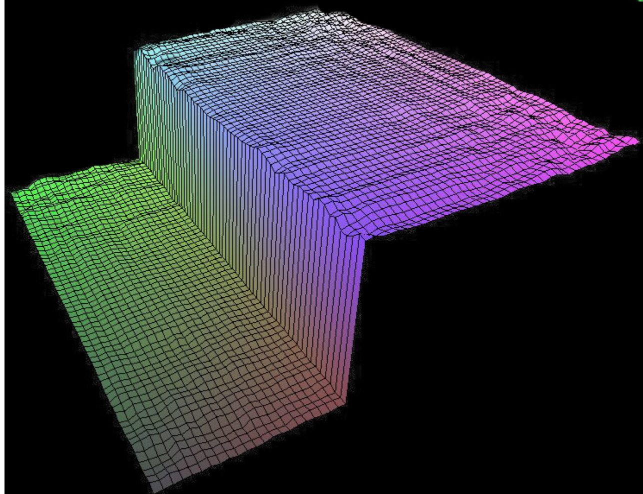
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi) - I(x)) \quad I(\xi)$$

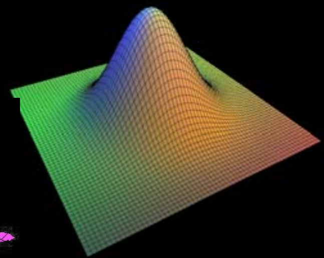


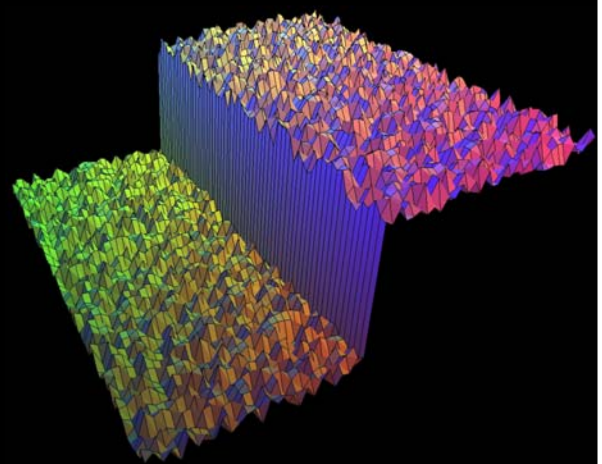
Bilateral filtering

- [Tomasi and Manduchi 1998]
- Spatial Gaussian f

$$J(x) \quad \frac{1}{k(x)} \quad f(x,\xi) \quad g(I(\xi)-I(x)) \quad I(\xi)$$



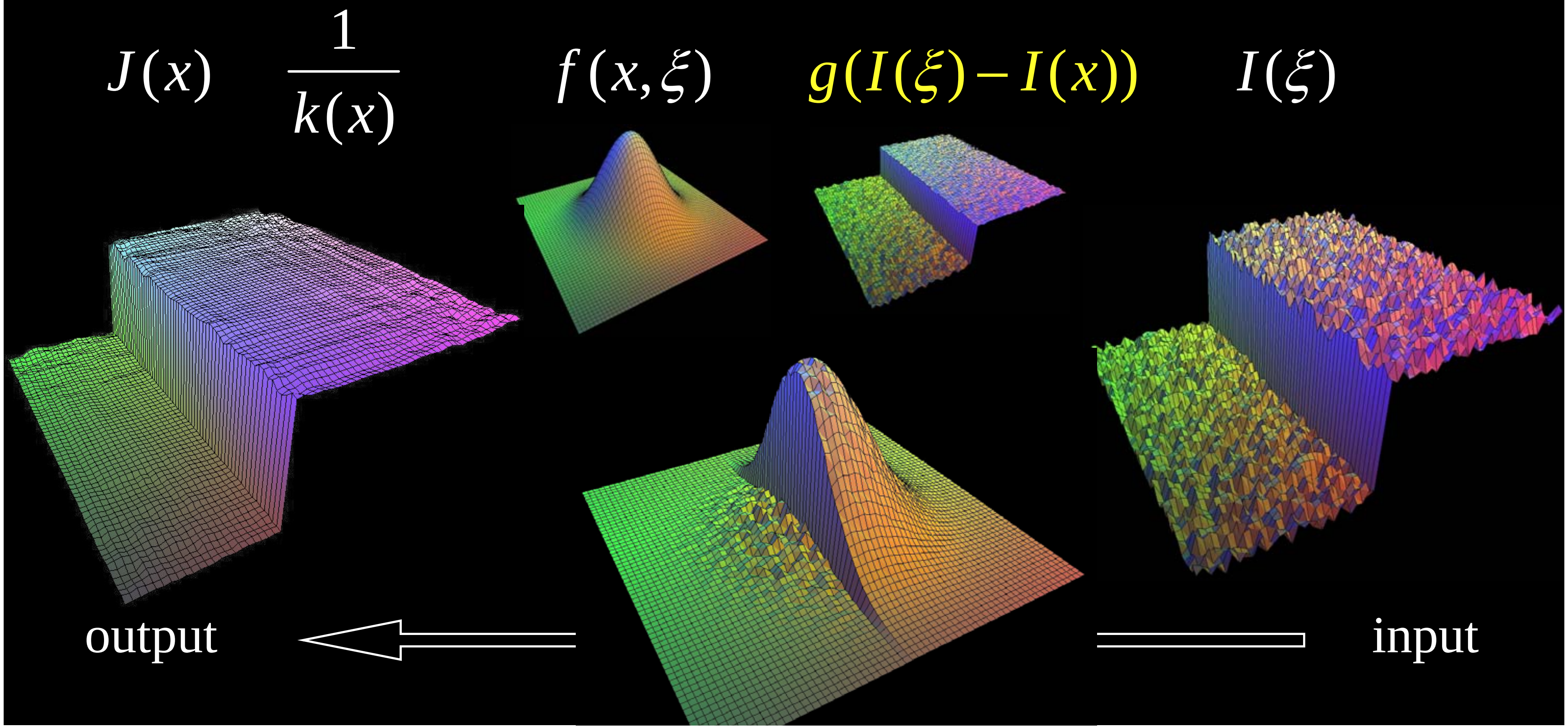




output ⟵ input

Bilateral filtering

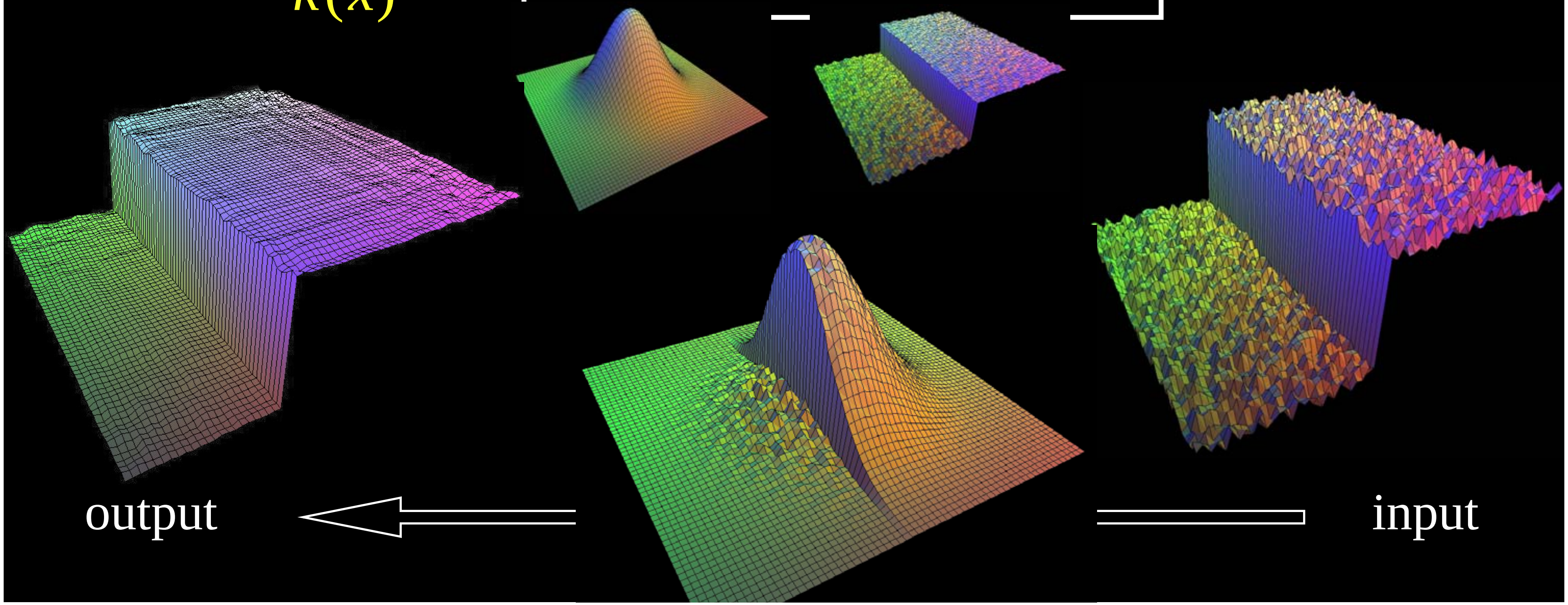
- [Tomasi and Manduchi 1998]
- Spatial Gaussian f
- Gaussian g on the intensity difference



Normalization factor

- [Tomasi and Manduchi 1998]
- k(x)= $\sum_{\xi} \boxed{f(x,\xi) \quad g(I(\xi)-I(x))}$

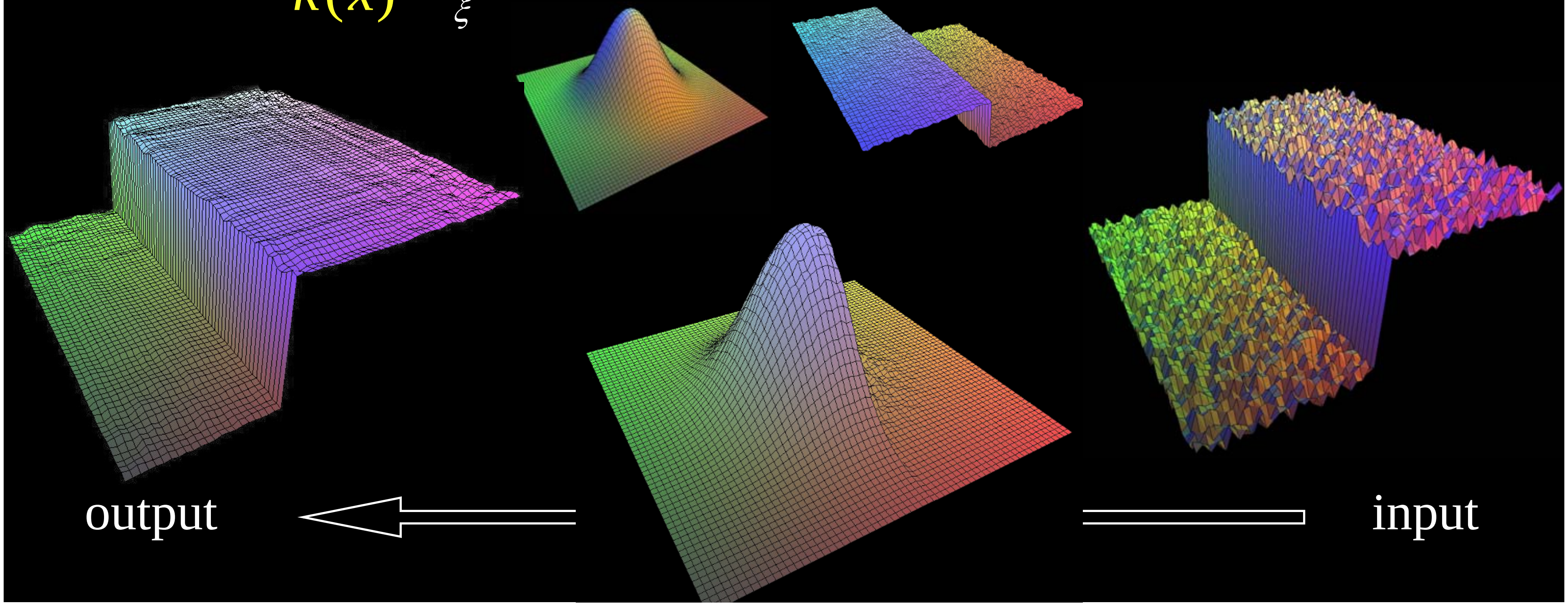
$$J(x) \quad \frac{1}{k(x)} \quad \sum_{\xi} \boxed{f(x,\xi) \quad g(I(\xi)-I(x))} \quad I(\xi)$$



Bilateral filtering is non-linear

- [Tomasi and Manduchi 1998]
- The weights are different for each output pixel

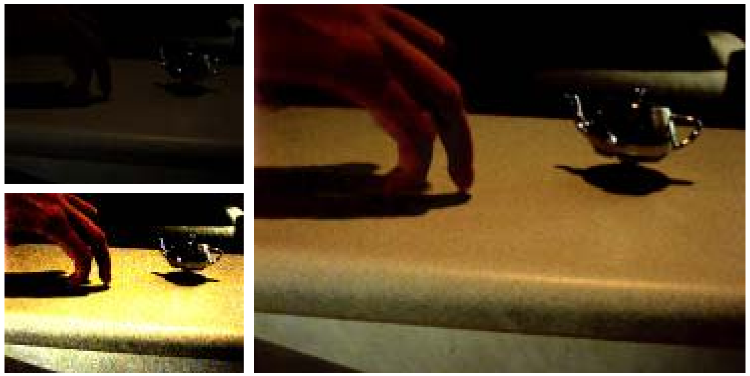
$$J(x) \quad \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi)-I(x)) \quad I(\xi)$$



Many Applications based on Bilateral Filter

Tone Mapping [Durand 02]

Flash / No-Flash [Eisemann 04, Petschnigg 04]



Virtual Video Exposure [Bennett 05]

Tone Management [Bae 06]

And many others…

Advantages of Bilateral Filter

- Easy to understand
  - Weighted mean of nearby pixels
- Easy to adapt
  - Distance between pixel values
- Easy to set up
  - Non-iterative

But Bilateral Filter is Nonlinear

- Slow but some accelerations exist:
  - [Elad 02]: Gauss-Seidel iterations
    - **Only for many iterations**
  - [Durand 02, Weiss 06]: fast approximation
    - **No formal understanding** of accuracy versus speed
    - [Weiss 06]: Only **box function** as spatial kernel

A Fast Approximation of the Bilateral Filter using a Signal Processing Approach

Sylvain Paris and Frédo Durand

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology

CSAIL

Definition of Bilateral Filter

- [Smith 97, Tomasi 98]
- Smoothes an image and preserves edges
- Weighted average of neighbors
- Weights
  - Gaussian on *space* distance
  - Gaussian on *range* distance
  - sum to 1

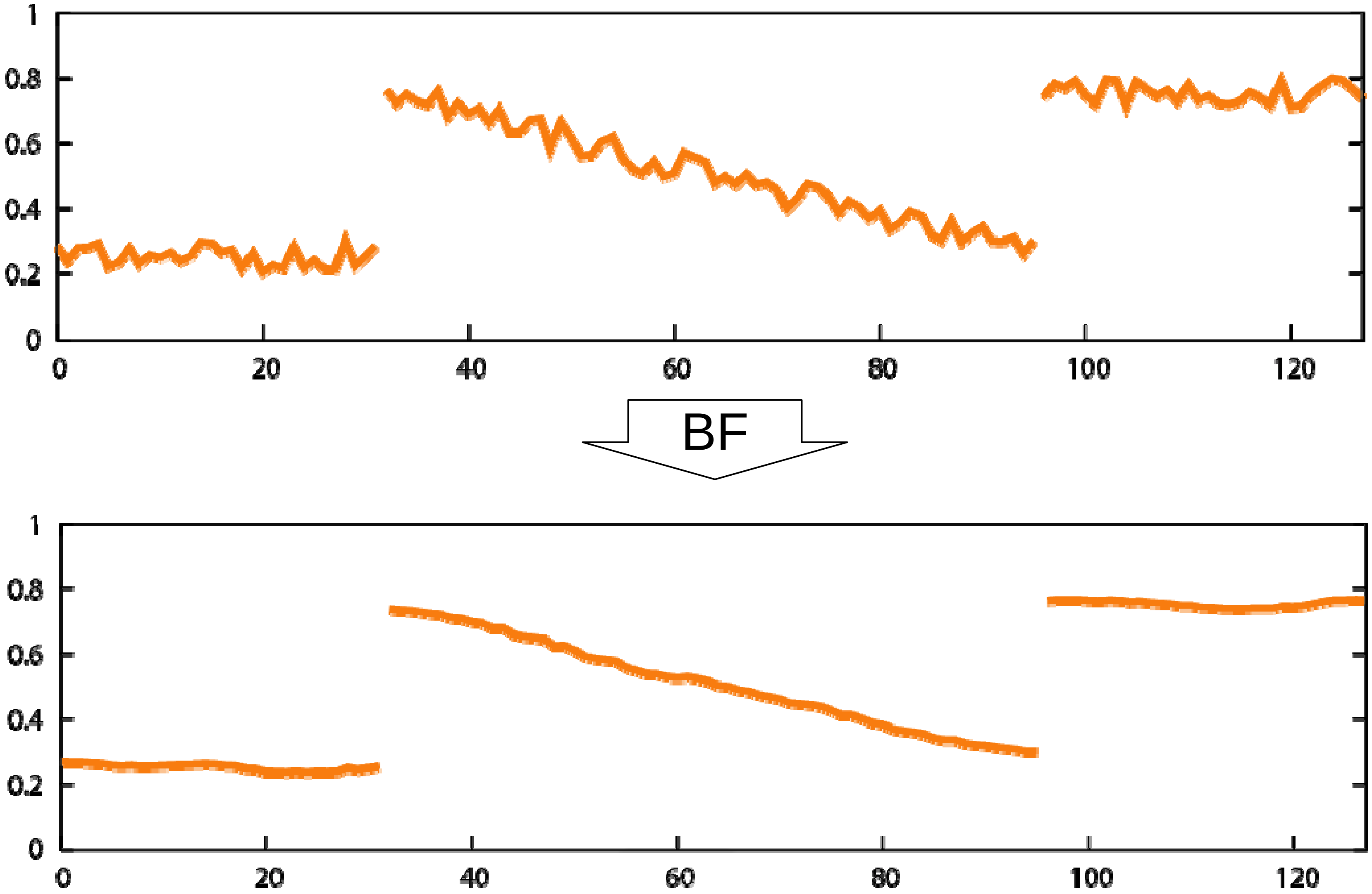
$$I^{\mathrm{bf}}_{\mathbf{p}} = \frac{1}{W^{\mathrm{bf}}_{\mathbf{p}}} \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)\, I_{\mathbf{q}}$$

space range

Contributions

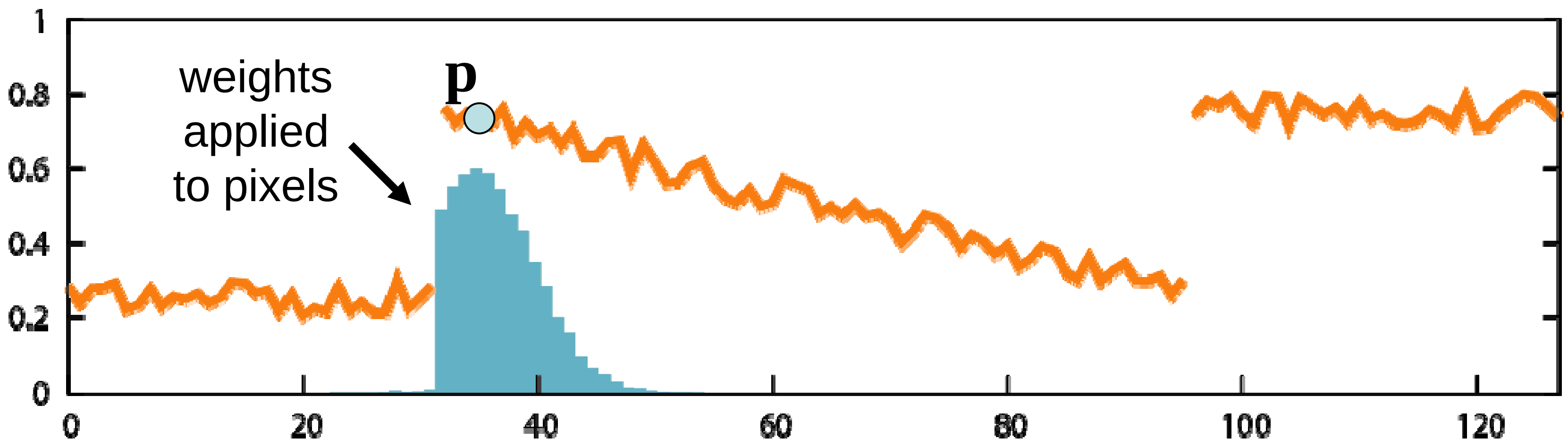
- Link with **linear filtering**
- **Fast** and **accurate** approximation

Intuition on 1D Signal



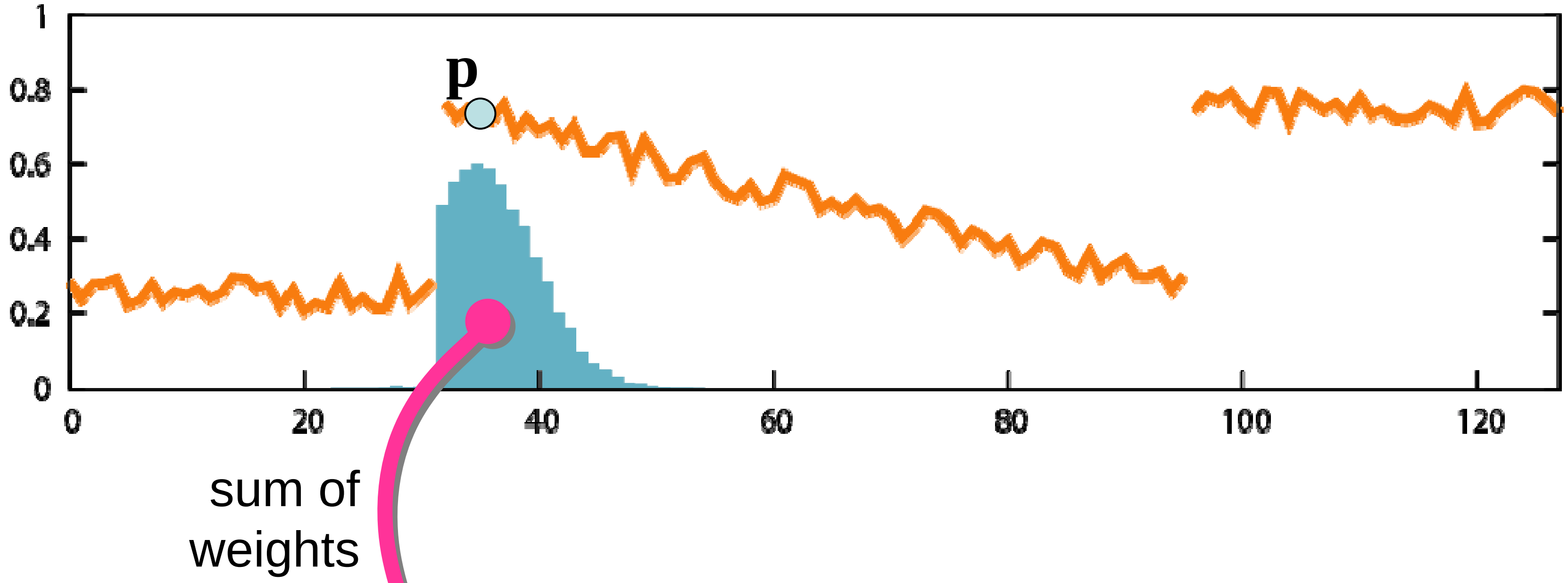
Intuition on 1D Signal

Weighted Average of Neighbors



- Near and similar pixels have influence.
- Far pixels have no influence.
- Pixels with different value have no influence.

Link with Linear Filtering
1. Handling the Division



$$I_{\mathbf{p}}^{\mathrm{bf}} = \frac{1}{W_{\mathbf{p}}^{\mathrm{bf}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)\, I_{\mathbf{q}}$$

Handling the division with a **projective space**.

Formalization: Handling the Division

$$I_{\mathbf{p}}^{\mathrm{bf}} = \frac{1}{W_{\mathbf{p}}^{\mathrm{bf}}} \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)\, I_{\mathbf{q}}$$

$$W_{\mathbf{p}}^{\mathrm{bf}} = \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)$$

- Normalizing factor as homogeneous coordinate
  - Multiply both sides by $W_{\mathbf{p}}^{\mathrm{bf}}$

$$\begin{pmatrix} W_{\mathbf{p}}^{\mathrm{bf}}\, I_{\mathbf{p}}^{\mathrm{bf}} \\ W_{\mathbf{p}}^{\mathrm{bf}} \end{pmatrix} = \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}}-I_{\mathbf{q}}|) \begin{pmatrix} I_{\mathbf{q}} \\ 1 \end{pmatrix}$$

Formalization: Handling the Division

$$\begin{pmatrix} W_{\mathbf{p}}^{\mathrm{bf}}\, I_{\mathbf{p}}^{\mathrm{bf}} \\ W_{\mathbf{p}}^{\mathrm{bf}} \end{pmatrix} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) \begin{pmatrix} W_{\mathbf{q}}\, I_{\mathbf{q}} \\ W_{\mathbf{q}} \end{pmatrix} \quad \text{with } W_{\mathbf{q}}=1$$

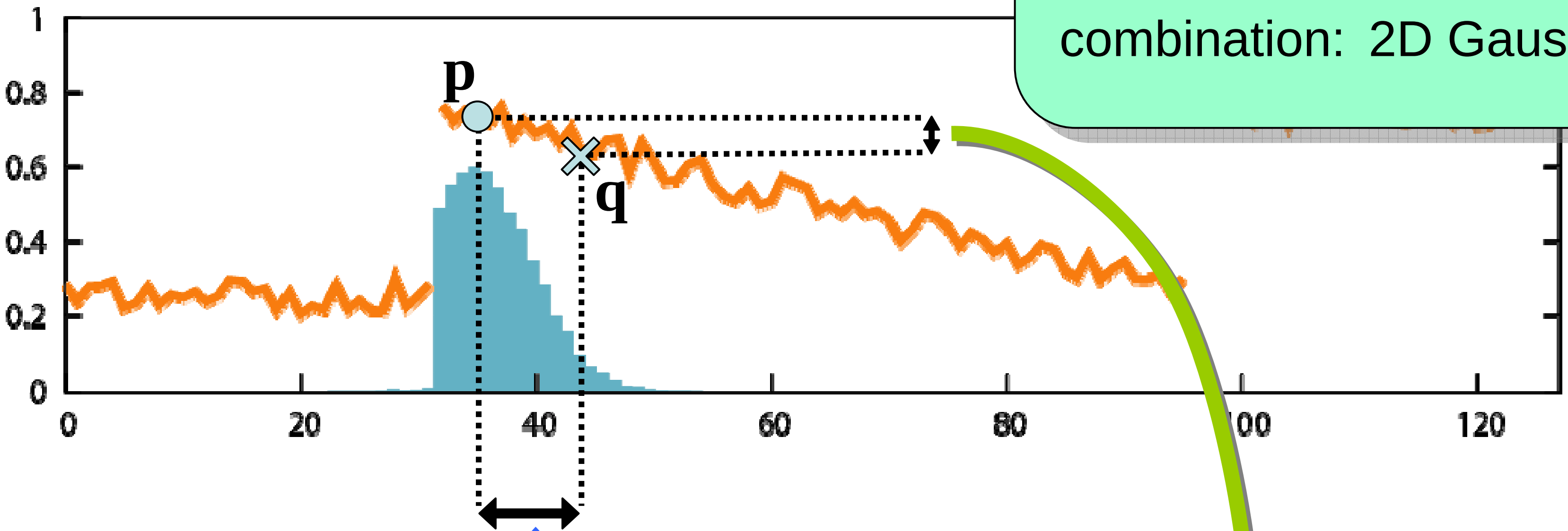
- Similar to homogeneous coordinates in projective space
- Division delayed until the end
- Next step: Adding a dimension to make a convolution appear

Link with Linear Filtering

2. Introducing a Convolution

space: 1D Gaussian
× range: 1D Gaussian

combination: 2D Gaussian



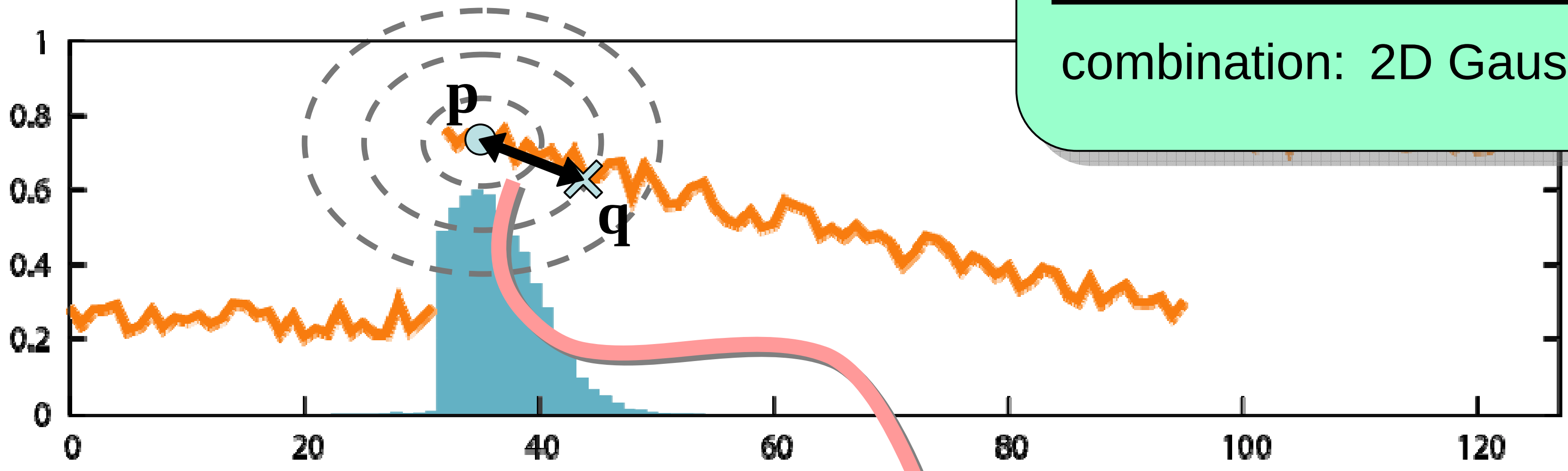
$$\begin{pmatrix} W^{bf}_{\mathbf{p}} I^{bf}_{\mathbf{p}} \\ W^{bf}_{\mathbf{p}} \end{pmatrix} = \sum_{\mathbf{q} \in \mathcal{S}} \underbrace{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)}_{\text{space}} \underbrace{G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)}_{\text{range}} \begin{pmatrix} W_{\mathbf{q}} I_{\mathbf{q}} \\ W_{\mathbf{q}} \end{pmatrix}$$

Link with Linear Filtering

2. Introducing a Convolution

space: 1D Gaussian
× range: 1D Gaussian

combination: 2D Gaussian



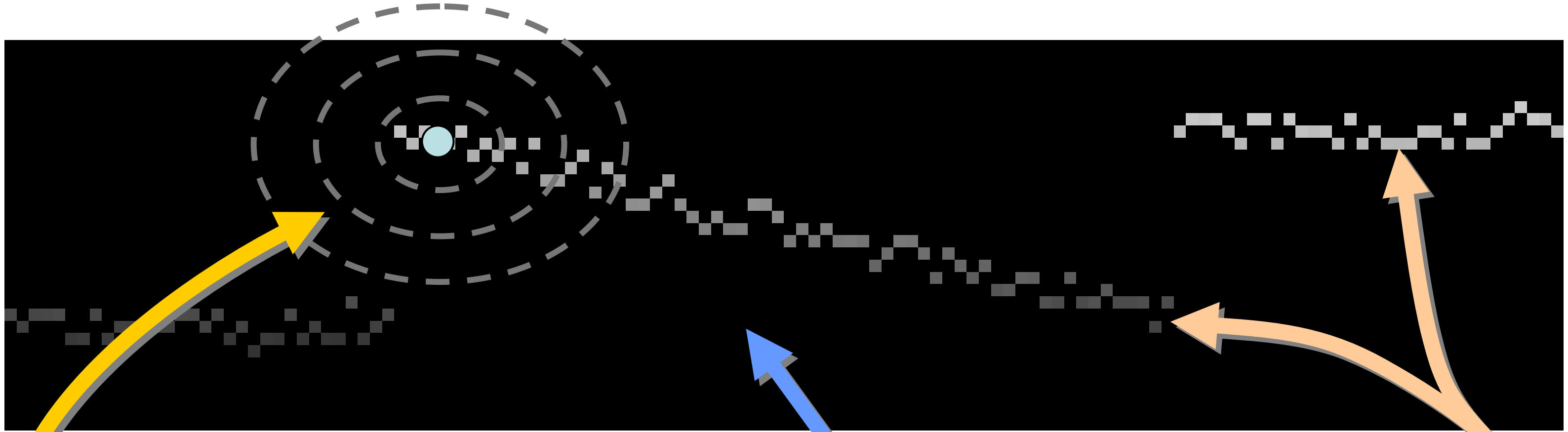
$$\begin{pmatrix} W_{\mathbf{p}}^{\text{bf}} I_{\mathbf{p}}^{\text{bf}} \\ W_{\mathbf{p}}^{\text{bf}} \end{pmatrix} = \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|) \begin{pmatrix} W_{\mathbf{q}}\, I_{\mathbf{q}} \\ W_{\mathbf{q}} \end{pmatrix}$$

space x range

Corresponds to a 3D Gaussian on a 2D image.

Link with Linear Filtering

2. Introducing a Convolution



sum all values

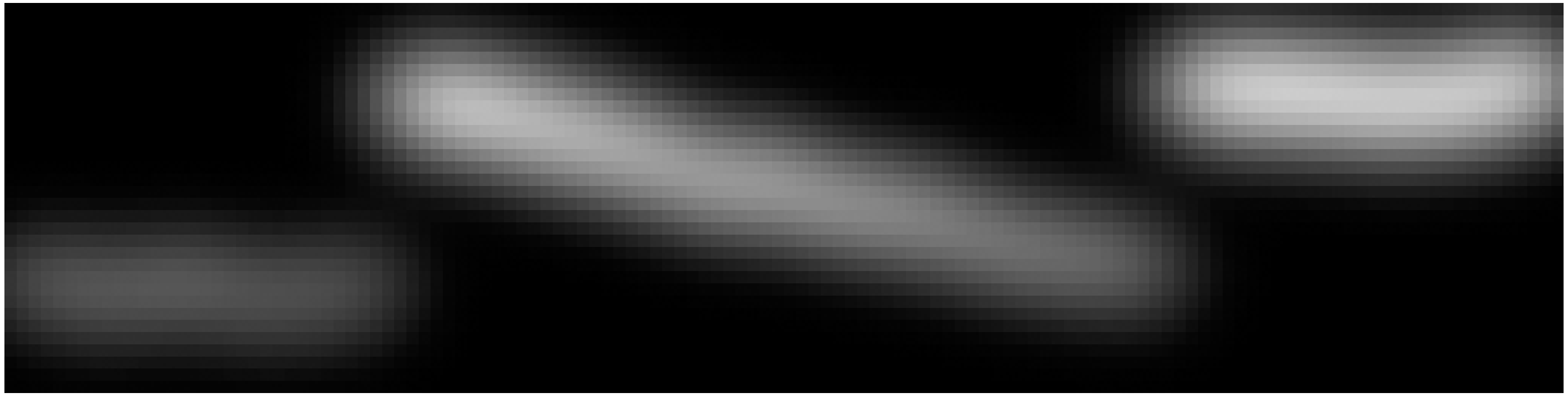
black = zero

$$\begin{pmatrix} W^{\mathrm{bf}}_{\mathbf{p}}\, I^{\mathrm{bf}}_{\mathbf{p}} \\ W^{\mathrm{bf}}_{\mathbf{p}} \end{pmatrix} = \sum_{(\mathbf{q},\zeta)\in\mathcal{S}\times\mathcal{R}} \text{space-range Gaussian} \begin{pmatrix} W_{\mathbf{q}}\, I_{\mathbf{q}} \\ W_{\mathbf{q}} \end{pmatrix}$$

sum all values multiplied by kernel ⇨ convolution

Link with Linear Filtering

2. Introducing a Convolution

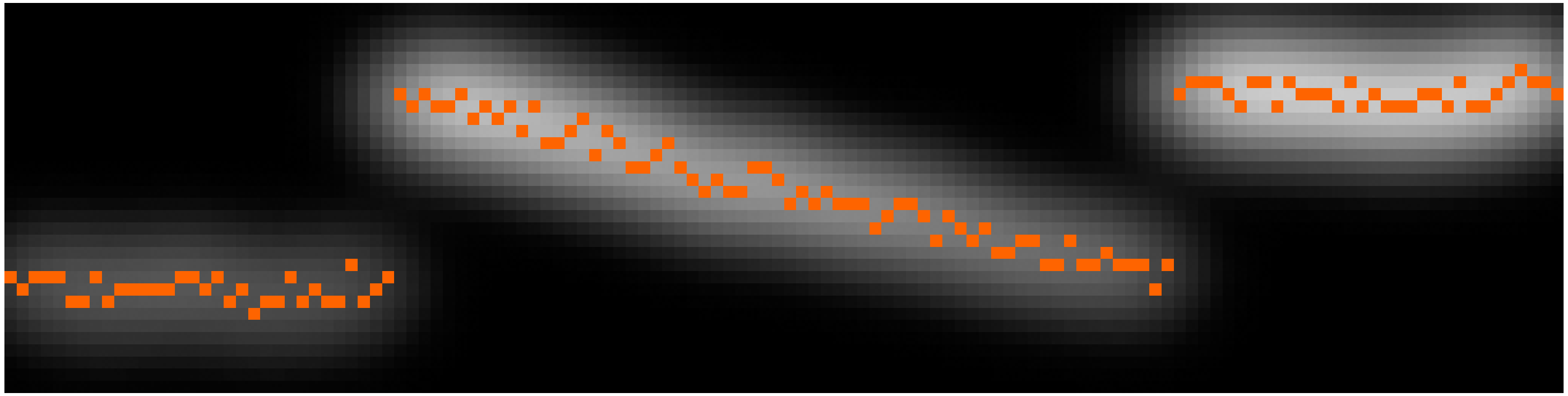


result of the convolution

$$\begin{pmatrix} W^{\mathrm{bf}}_{\mathbf{p}} \; I^{\mathrm{bf}}_{\mathbf{p}} \\ W^{\mathrm{bf}}_{\mathbf{p}} \end{pmatrix} = \sum_{(\mathbf{q},\zeta)\in\mathcal{S}\times\mathcal{R}} \text{space-range Gaussian} \begin{pmatrix} W_{\mathbf{q}} \; I_{\mathbf{q}} \\ W_{\mathbf{q}} \end{pmatrix}$$

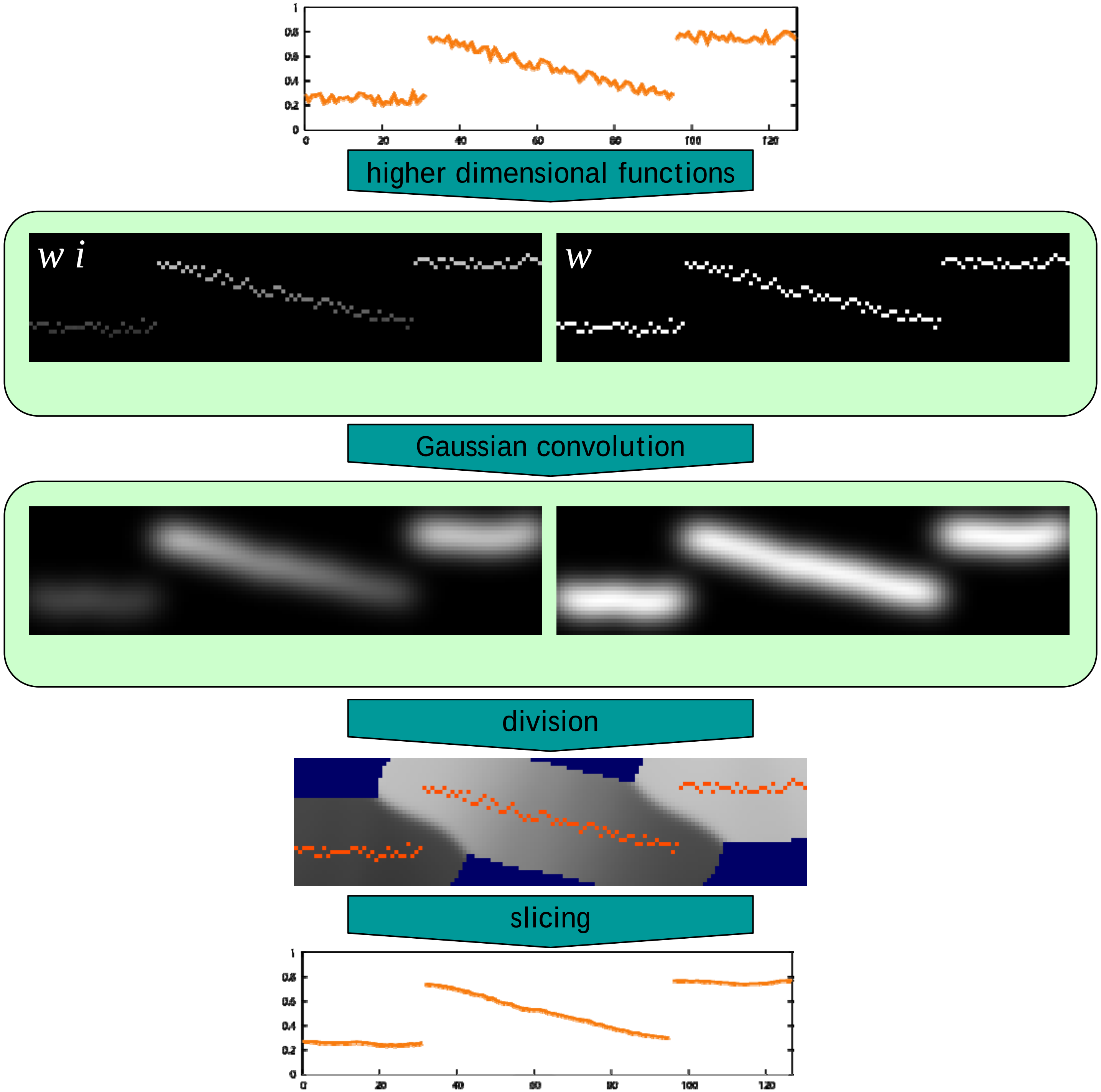
Link with Linear Filtering

2. Introducing a Convolution



result of the convolution

$$\begin{pmatrix} W^{\text{bf}}_{\mathbf{p}} \, I^{\text{bf}}_{\mathbf{p}} \\ W^{\text{bf}}_{\mathbf{p}} \end{pmatrix} = \sum_{(\mathbf{q},\zeta)\in\mathcal{S}\times\mathcal{R}} \text{space-range Gaussian} \begin{pmatrix} W_{\mathbf{q}} \, I_{\mathbf{q}} \\ W_{\mathbf{q}} \end{pmatrix}$$


higher dimensional functions
w i
w
Gaussian convolution
division
slicing

Reformulation: Summary

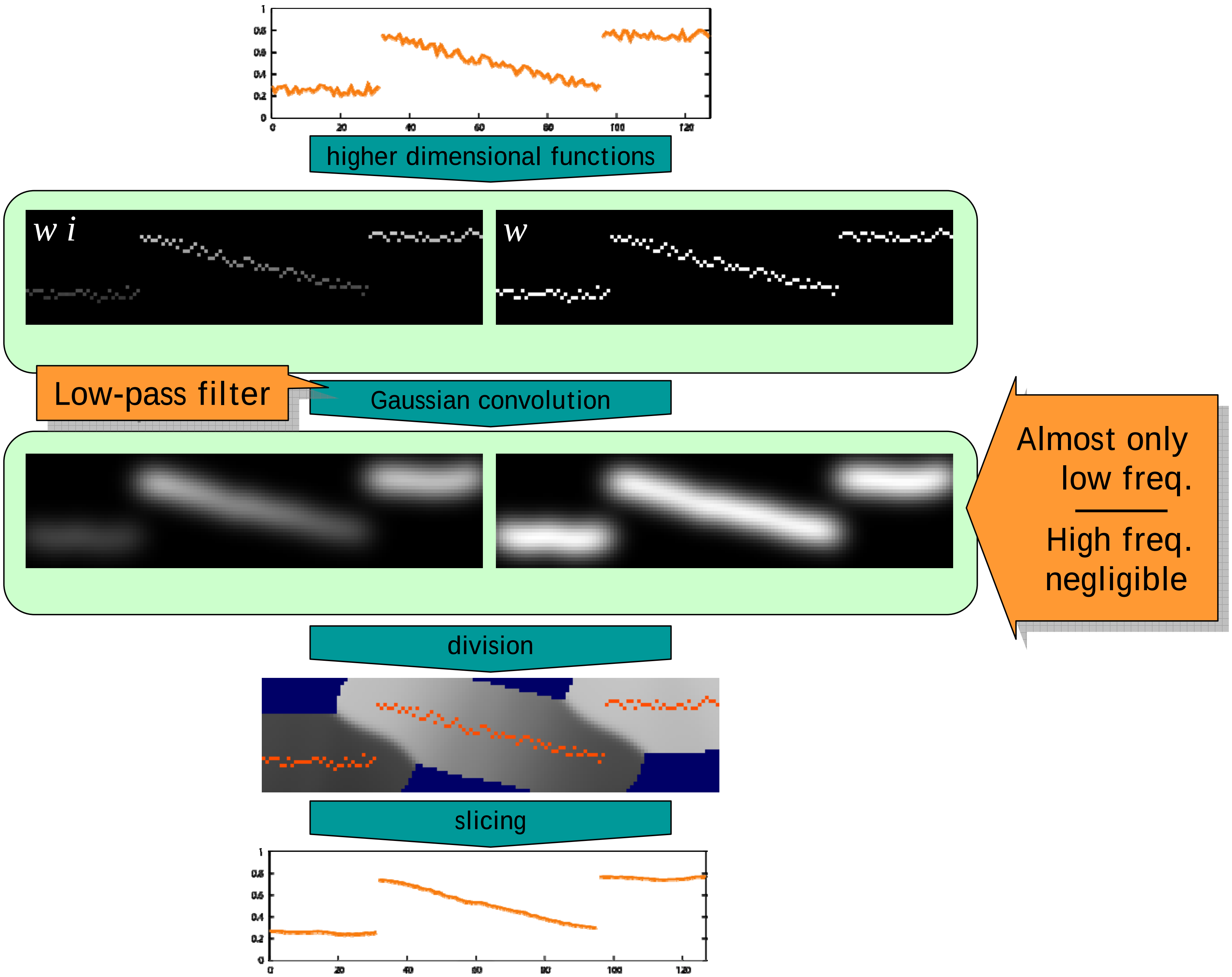
DigiVFX

$$\textbf{linear:} \qquad (w^{\mathrm{bf}}\, i^{\mathrm{bf}}, w^{\mathrm{bf}}) = g_{\sigma_{\mathrm{s}},\sigma_{\mathrm{r}}} \otimes (wi, w)$$

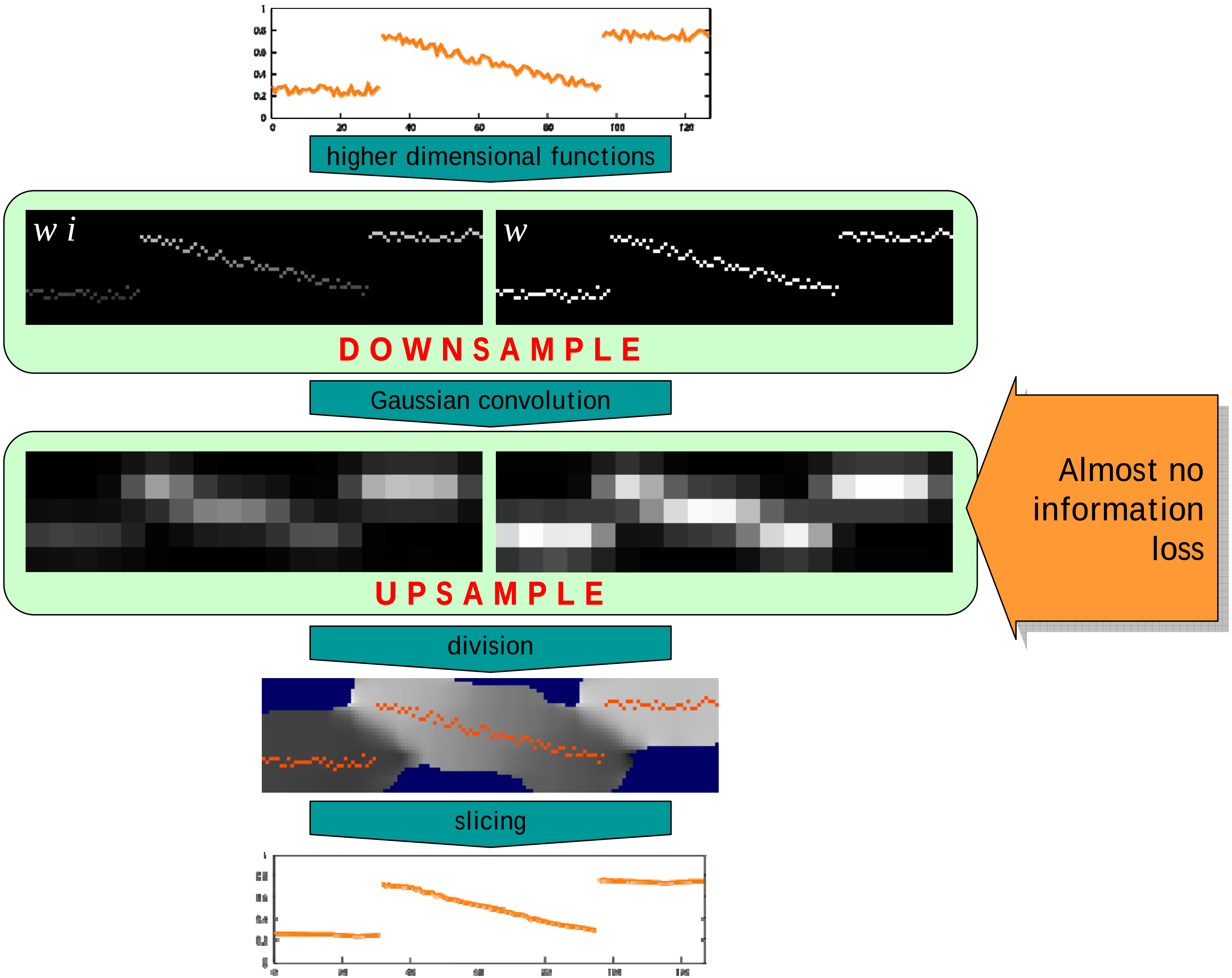
$$\textbf{nonlinear:} \qquad I_{\mathbf{p}}^{\mathrm{bf}} = \frac{w^{\mathrm{bf}}(\mathbf{p}, I_{\mathbf{p}})\; i^{\mathrm{bf}}(\mathbf{p}, I_{\mathbf{p}})}{w^{\mathrm{bf}}(\mathbf{p}, I_{\mathbf{p}})}$$

1. Convolution in higher dimension
   - expensive but well understood (linear, FFT, etc)
2. Division and slicing
   - nonlinear but simple and pixel-wise

Exact reformulation



higher dimensional functions
w i
w
Low-pass filter
Gaussian convolution
Almost only low freq.
High freq. negligible
division
slicing



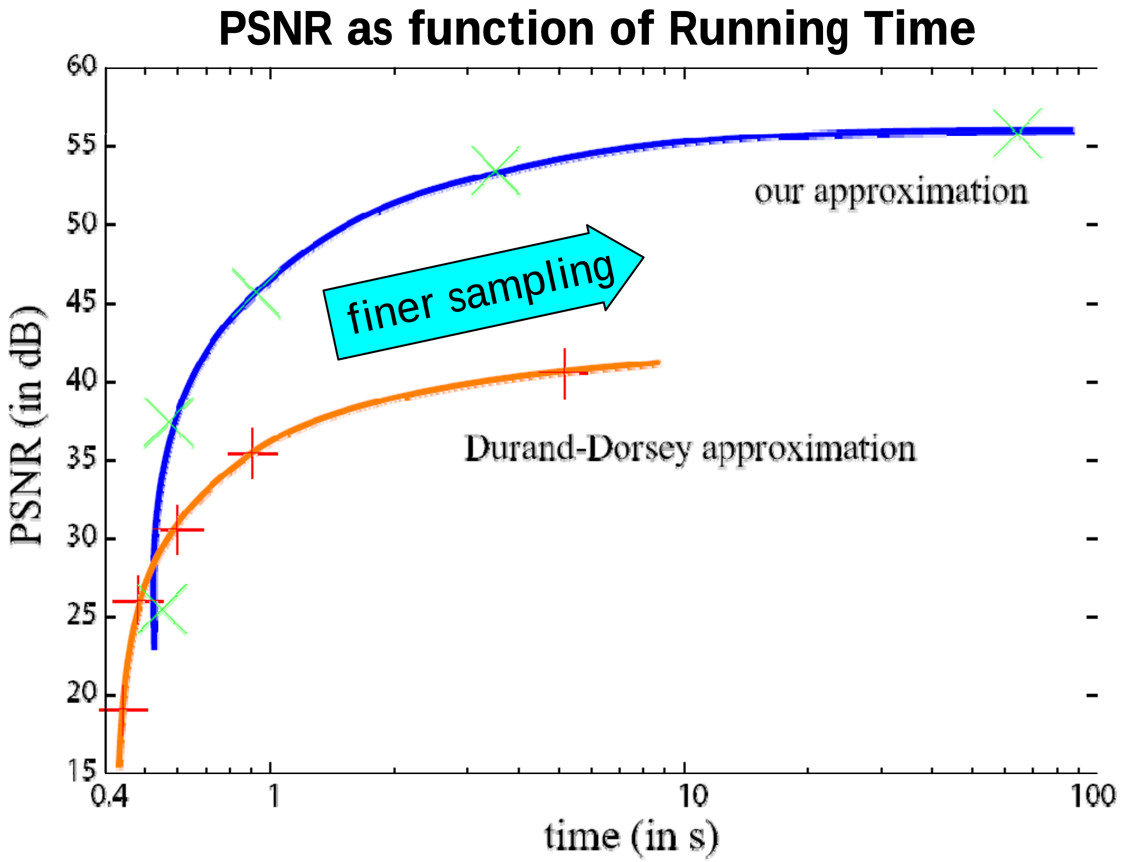
higher dimensional functions
w i
w
DOWNSAMPLE
Gaussian convolution
UPSAMPLE
Almost no information loss
division
slicing

Fast Convolution by Downsampling

- Downsampling cuts frequencies above Nyquist limit
  - Less data to process
  - But induces error
- Evaluation of the approximation
  - Precision versus running time
  - Visual accuracy

Accuracy versus Running Time

- Finer sampling increases accuracy.
- More precise than previous work.



Digital photograph 1200×1600

Straightforward implementation is over 10 minutes.

Visual Results

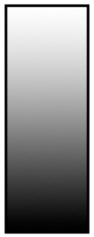
1200 × 1600

- Comparison with previous work [Durand 02]
  - running time = 1s for both techniques

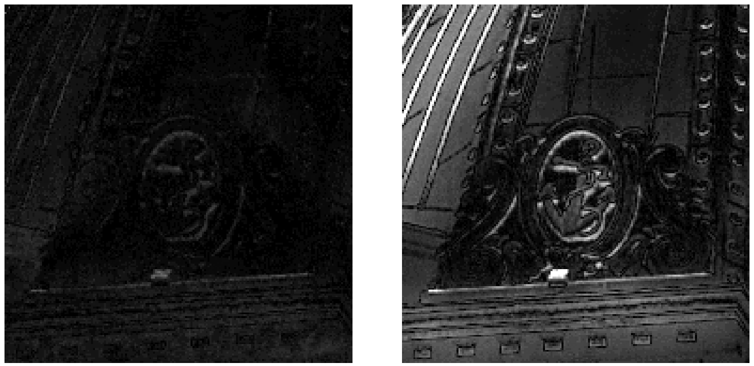
input | exact BF | our result | prev. work

difference with exact computation (intensities in [0:1])

0.1

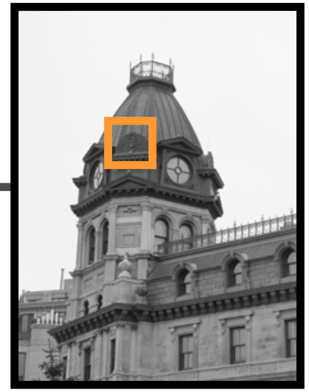


0

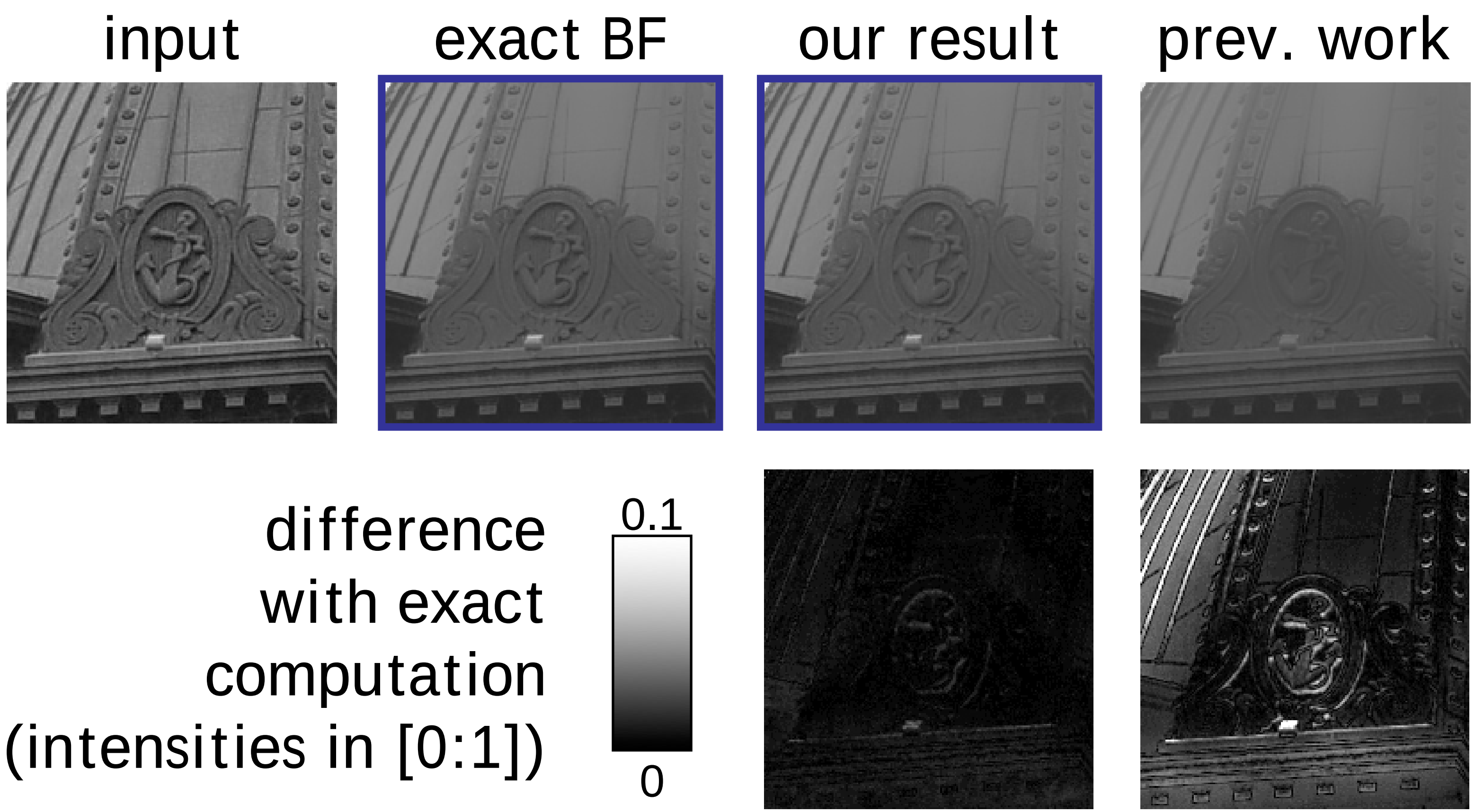


Visual Results

- Comparison with previous work [Durand 02]
  - running time = 1s for both techniques

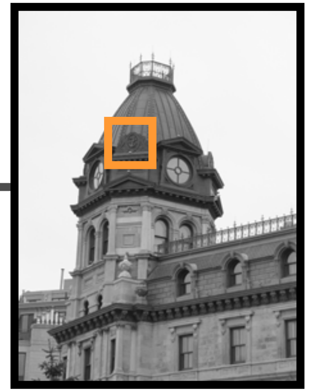


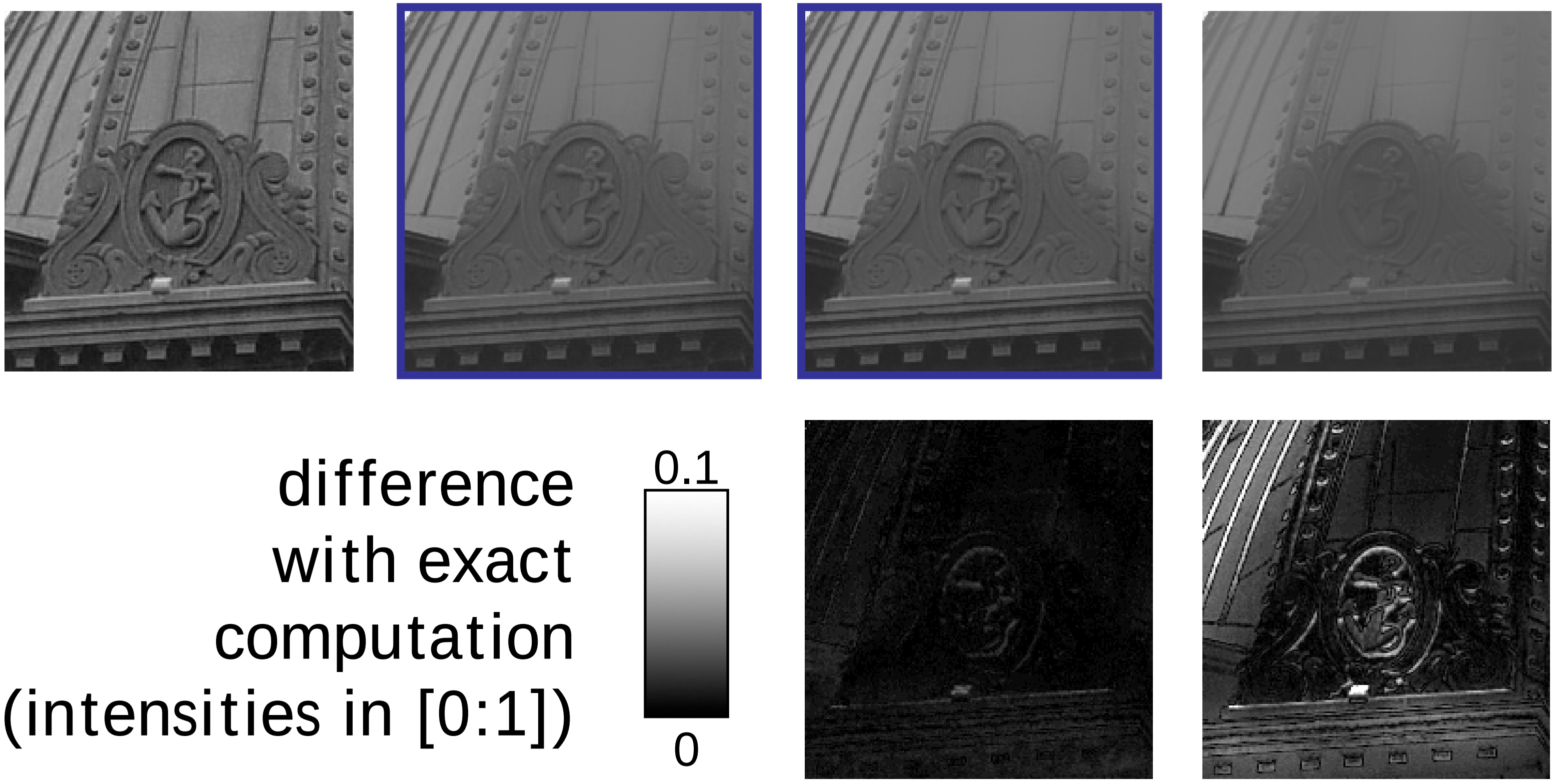
1200 × 1600



Visual Results

- Comparison with previous work [Durand 02]
  - running time = 1s for both techniques

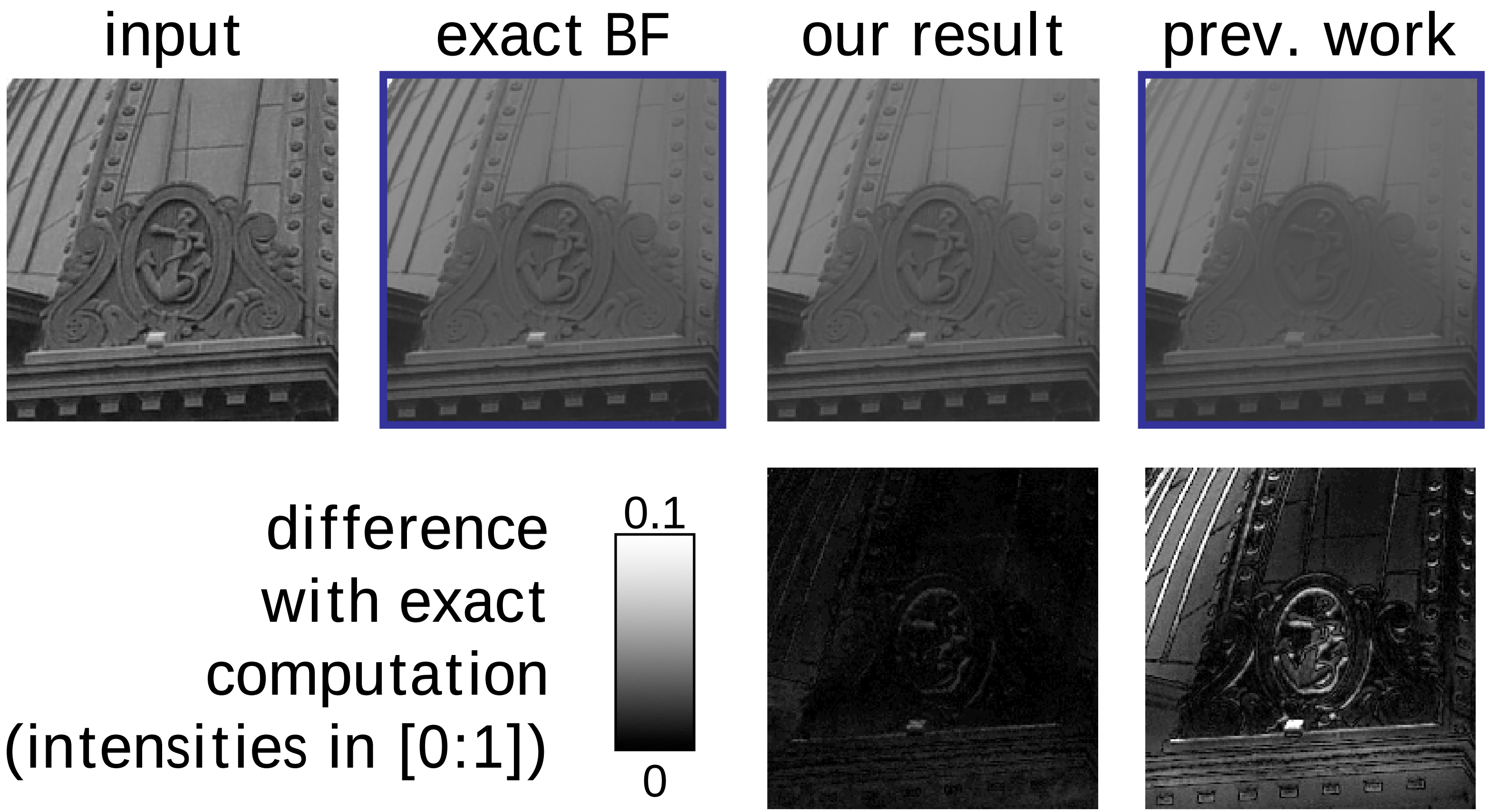




Visual Results

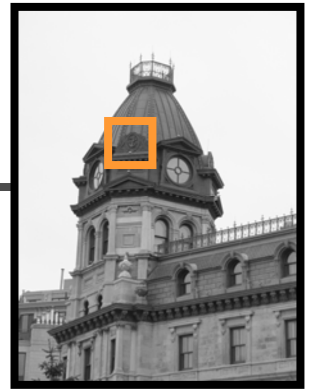
1200 × 1600

- Comparison with previous work [Durand 02]
  - running time = 1s for both techniques

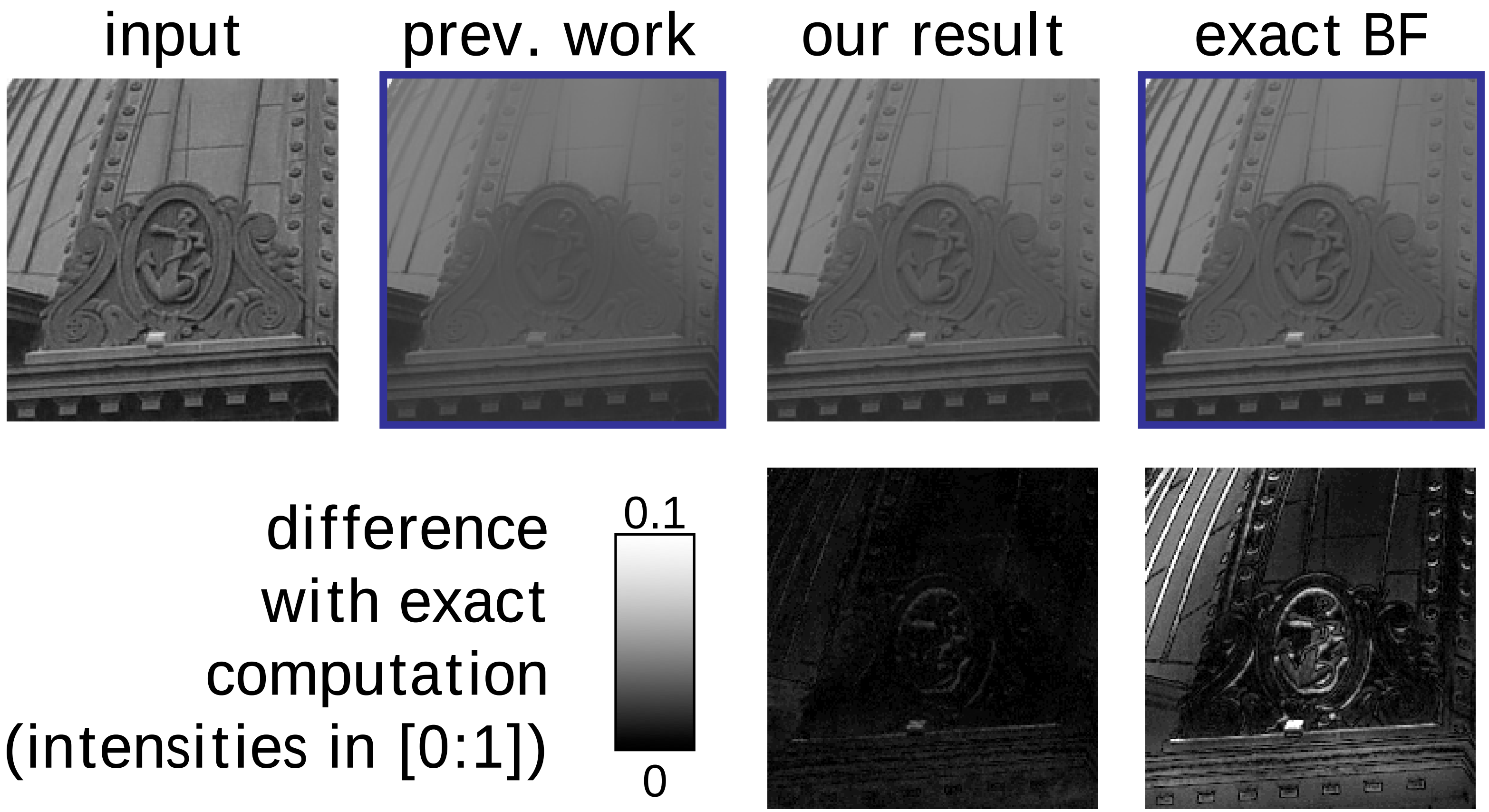


Visual Results

- Comparison with previous work [Durand 02]
  - running time = 1s for both techniques



1200 × 1600



Discussion

DigiVFX

- Higher dimension ⇨ advantageous formulation
  - akin to Level Sets with topology
  - our approach: isolate nonlinearities
  - dimension increase largely offset by downsampling

- Space-range domain already appeared
  - [Sochen 98, Barash 02]: image as an embedded manifold
  - new in our approach: image as a dense function

Conclusions

higher dimension ⇨ "better" computation

Practical gain

- Interactive running time
- Visually similar results
- Simple to code (100 lines)

Theoretical gain

- Link with linear filters
- Separation linear/nonlinear
- Signal processing framework

Two-scale Tone Management for Photographic Look

Soonmin Bae, Sylvain Paris, and Frédo Durand

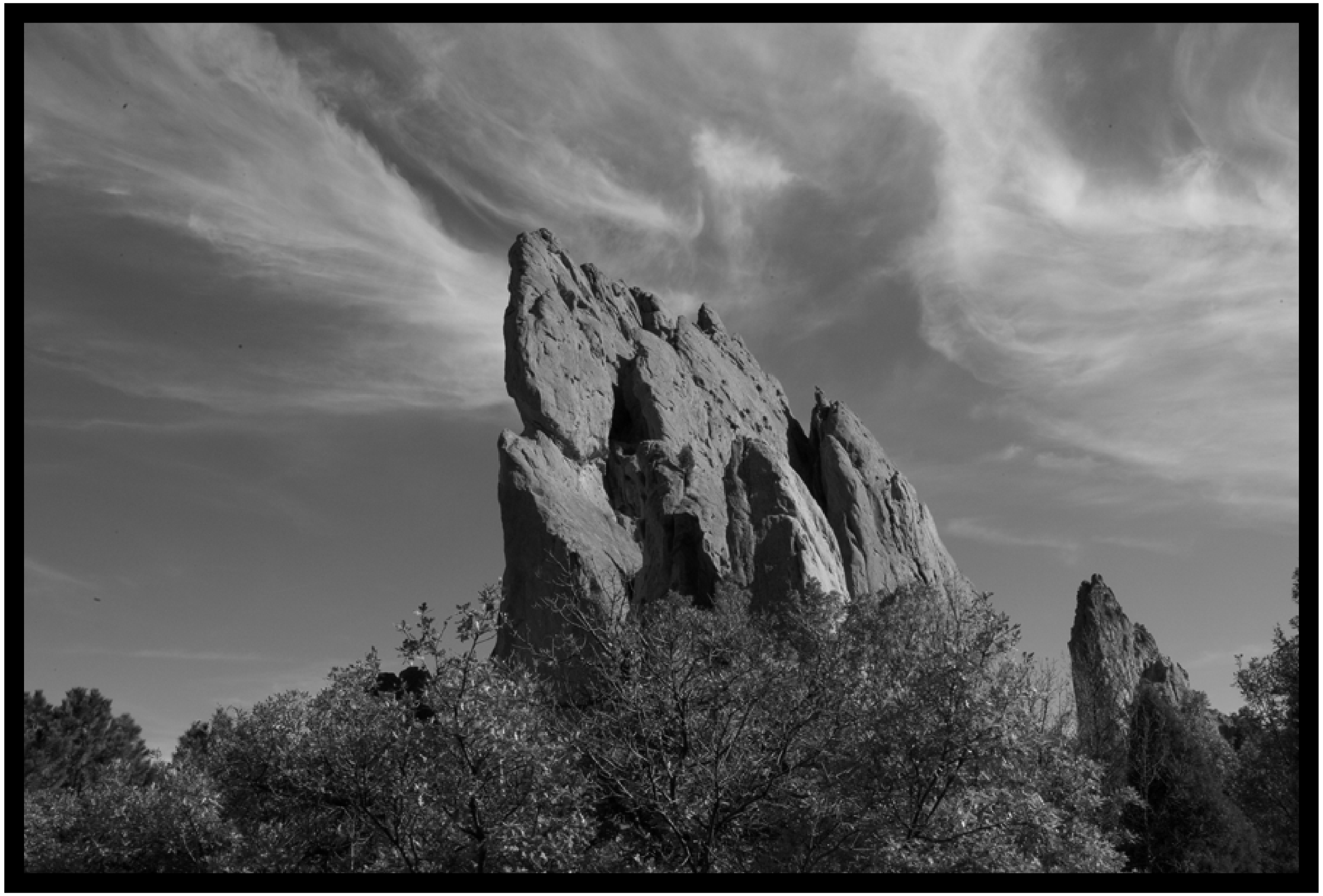
MIT CSAIL

SIGGRAPH2006

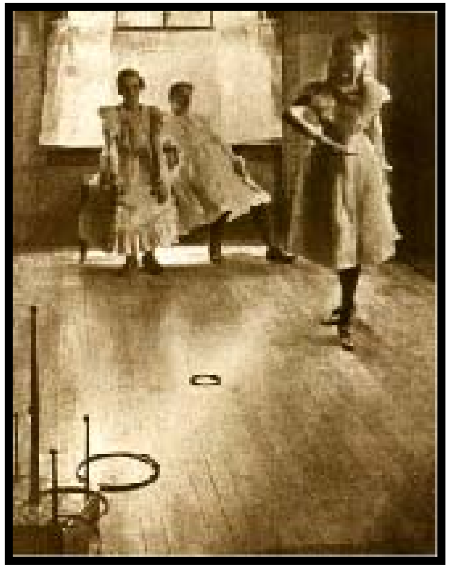
Ansel Adams

Ansel Adams, *Clearing Winter Storm*

An Amateur Photographer



A Variety of Looks



Goals

- Control over photographic look
- Transfer “look” from a model photo

For example,

we want

with the look of

Aspects of Photographic Look

- Subject choice
- Framing and composition

➔ Specified by input photos

Input

- Tone distribution and contrast

➔Modified based on model photos

Model

Tonal Aspects of Look


Ansel Adams


Kenro Izu

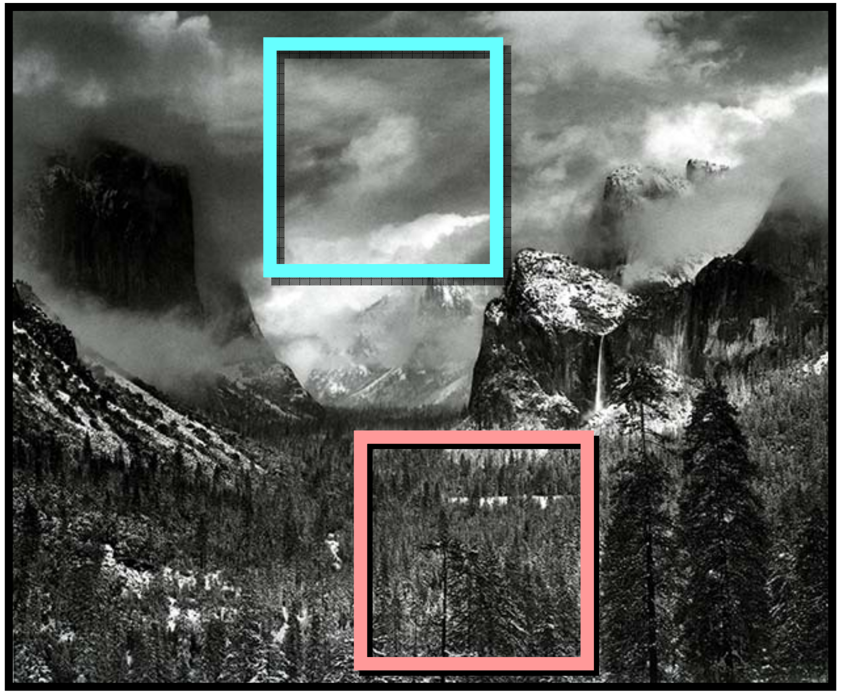
Tonal aspects of Look - Global Contrast

Ansel Adams

Kenro Izu

High Global Contrast **Low Global Contrast**

Tonal aspects of Look - Local Contrast

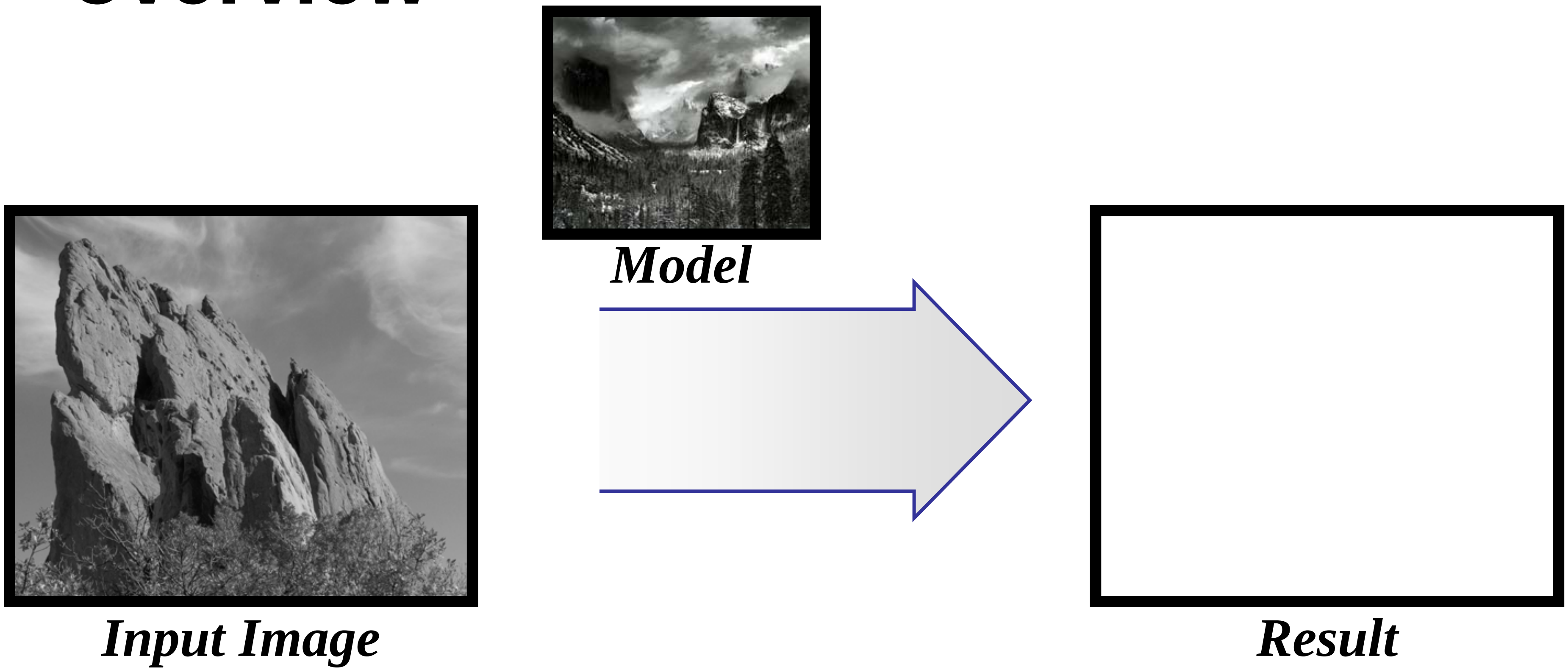


Ansel Adams

Kenro Izu

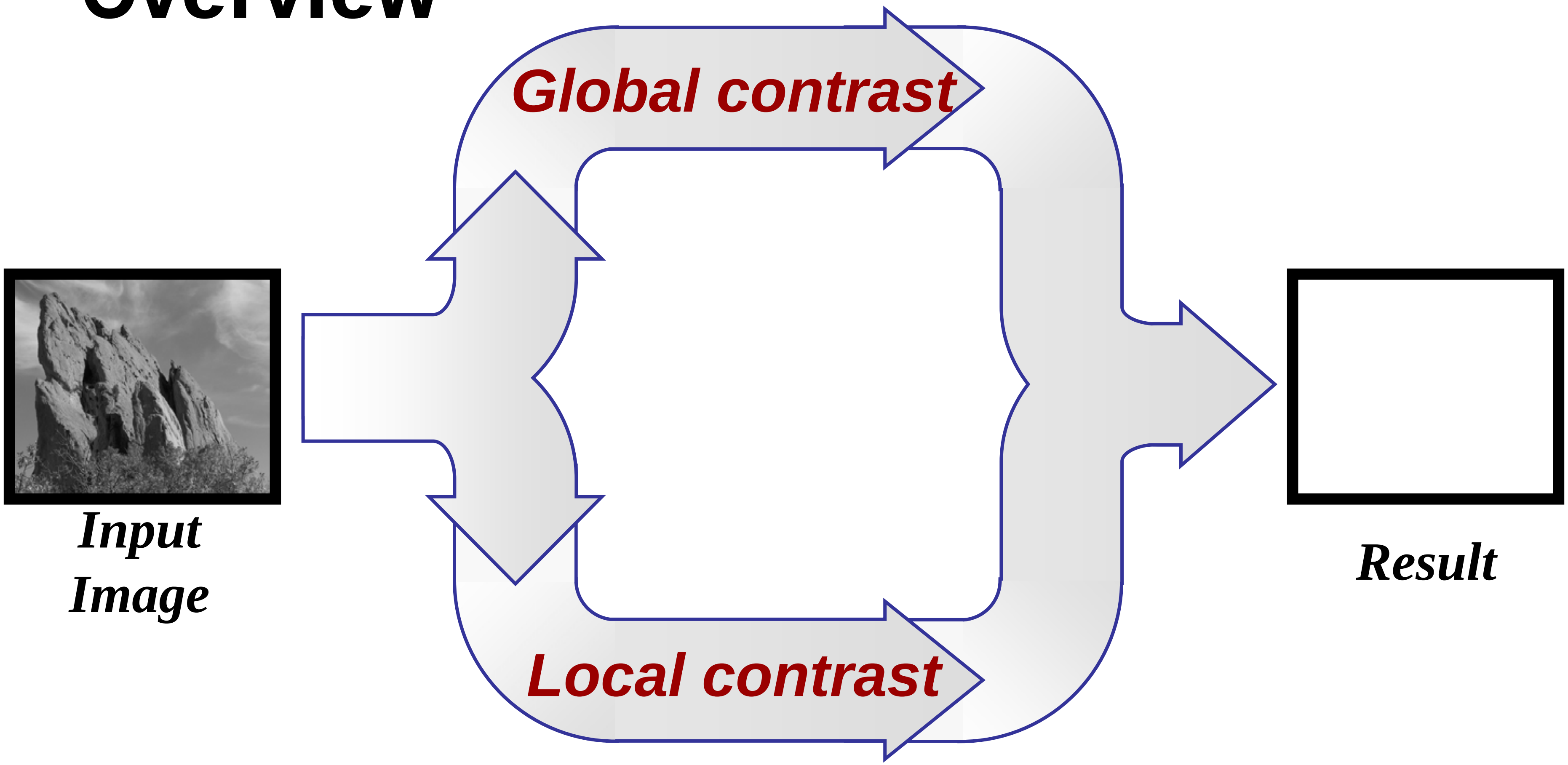
Variable amount of texture **Texture everywhere**

Overview



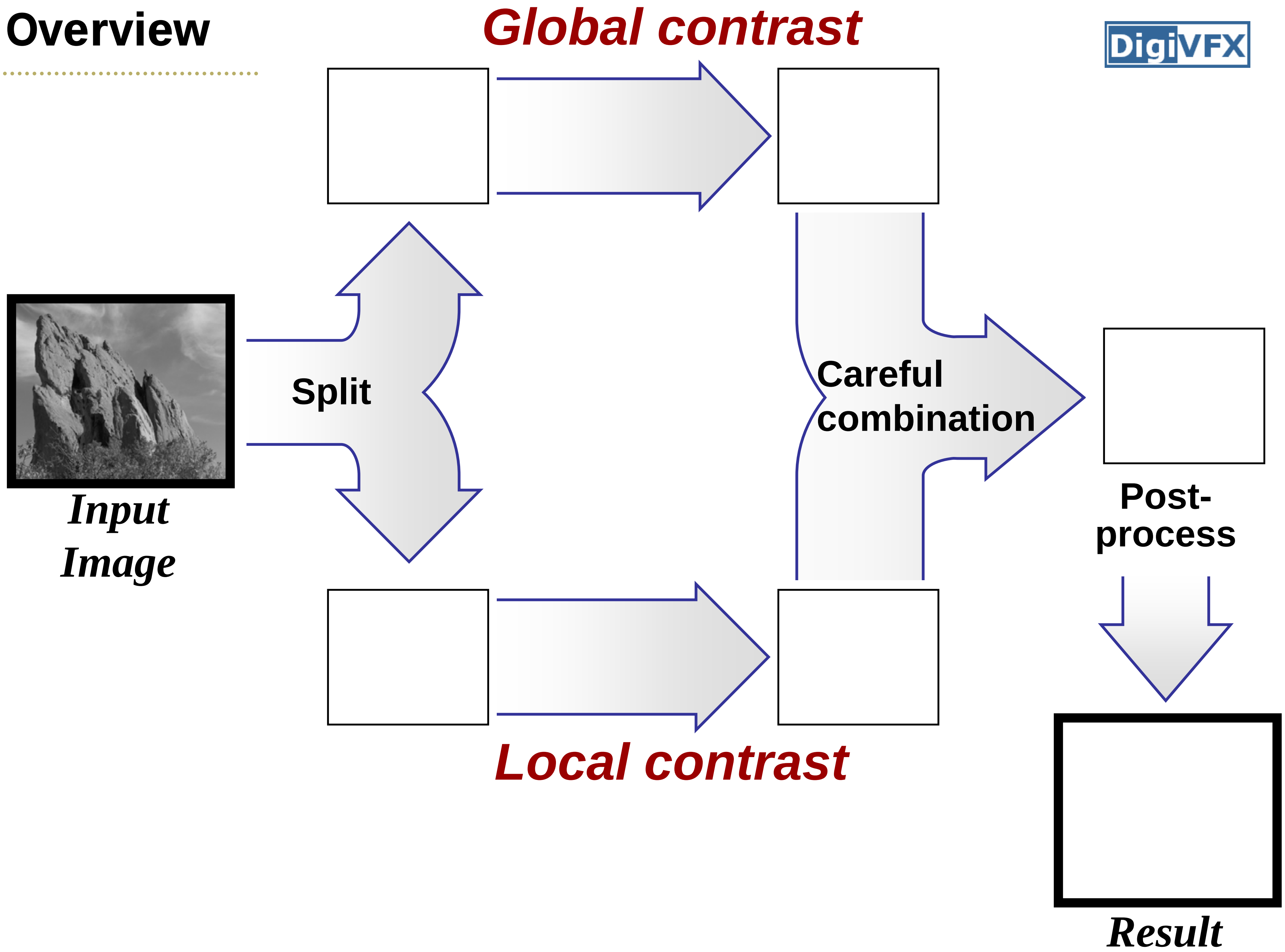
- Transfer look between photographs
  - Tonal aspects

Overview

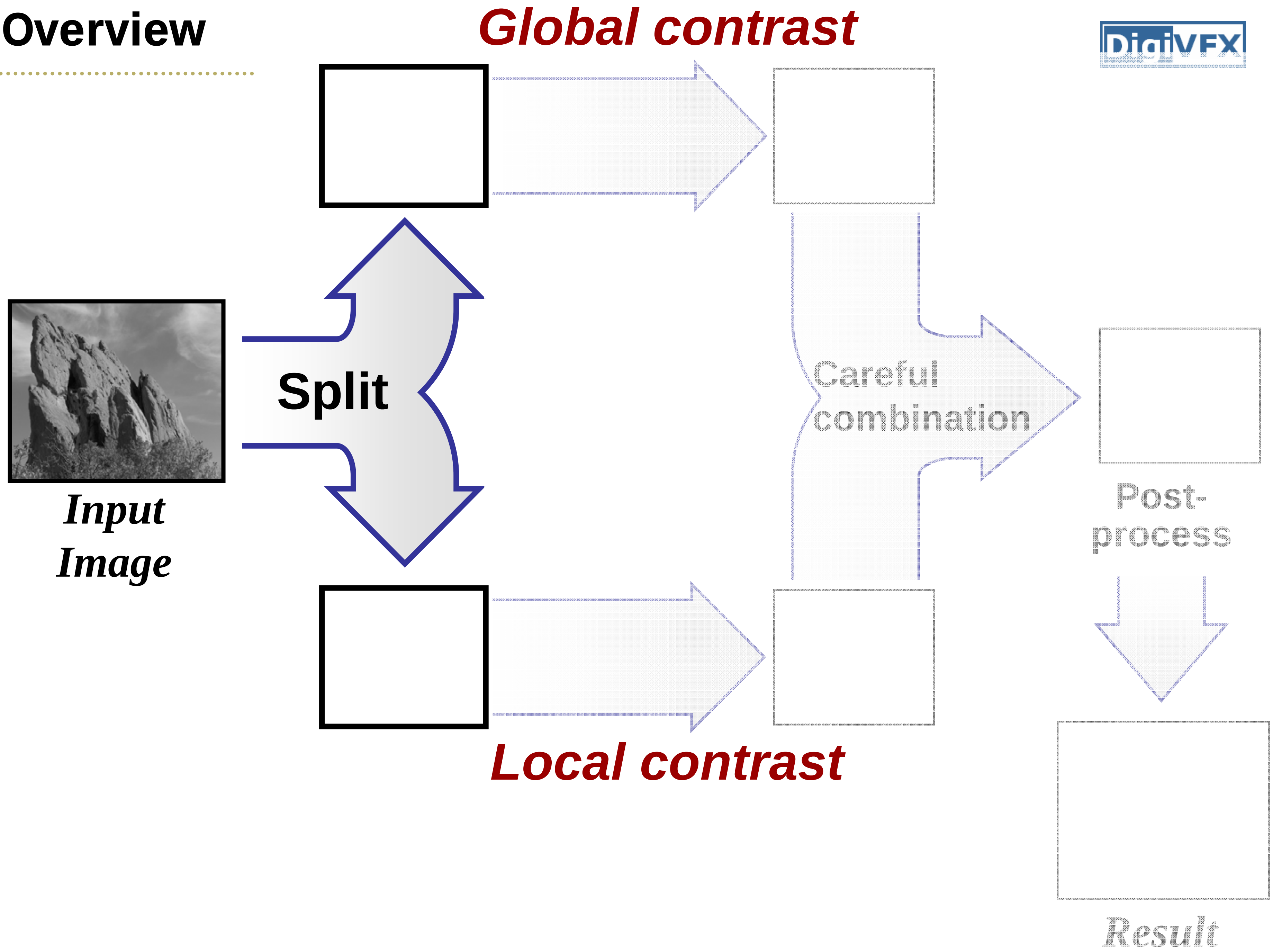


- Separate global and local contrast

Overview


Global contrast
Input Image
Split
Careful combination
Post-process
Local contrast
Result

Overview

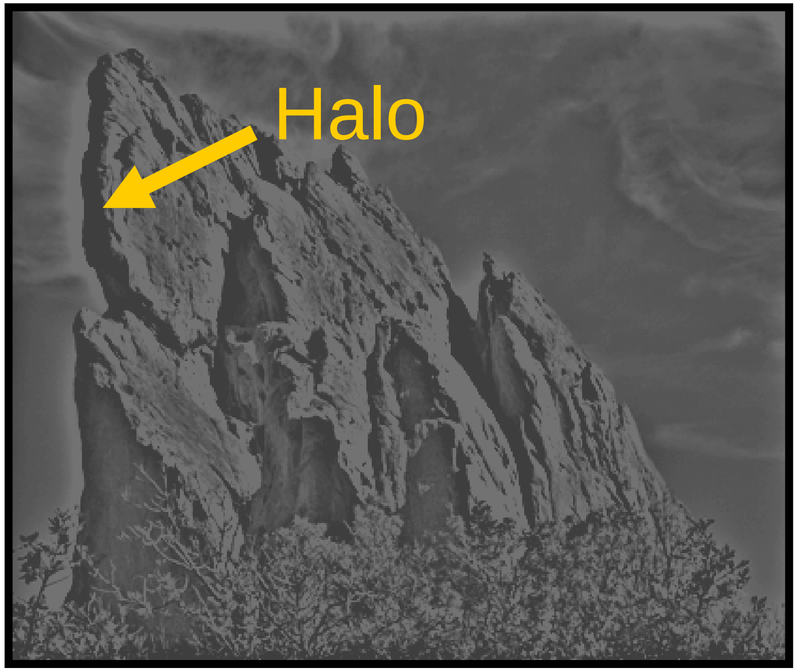

Global contrast
Input Image
Split
Careful combination
Post-process
Local contrast
Result

Split Global vs. Local Contrast

- Naïve decomposition: low vs. high frequency
  - Problem: introduce blur & halos

Low frequency

Global contrast



High frequency

Local contrast

Bilateral Filter

- Edge-preserving smoothing [Tomasi 98]
- We build upon tone mapping [Durand 02]

After bilateral filtering

Global contrast

Residual after filtering

Local contrast

Bilateral Filter

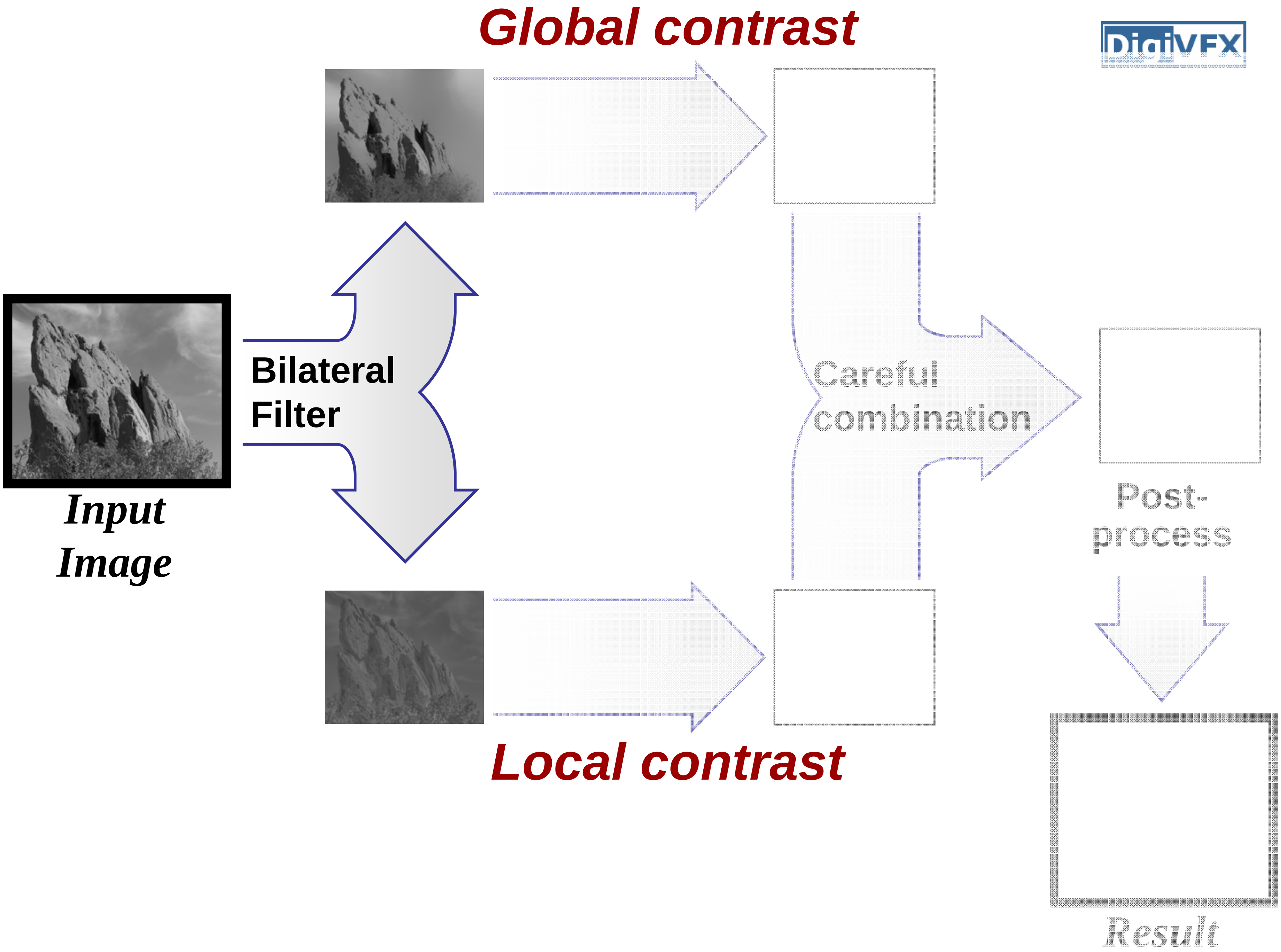
- Edge-preserving smoothing [Tomasi 98]
- We build upon tone mapping [Durand 02]

After bilateral filtering

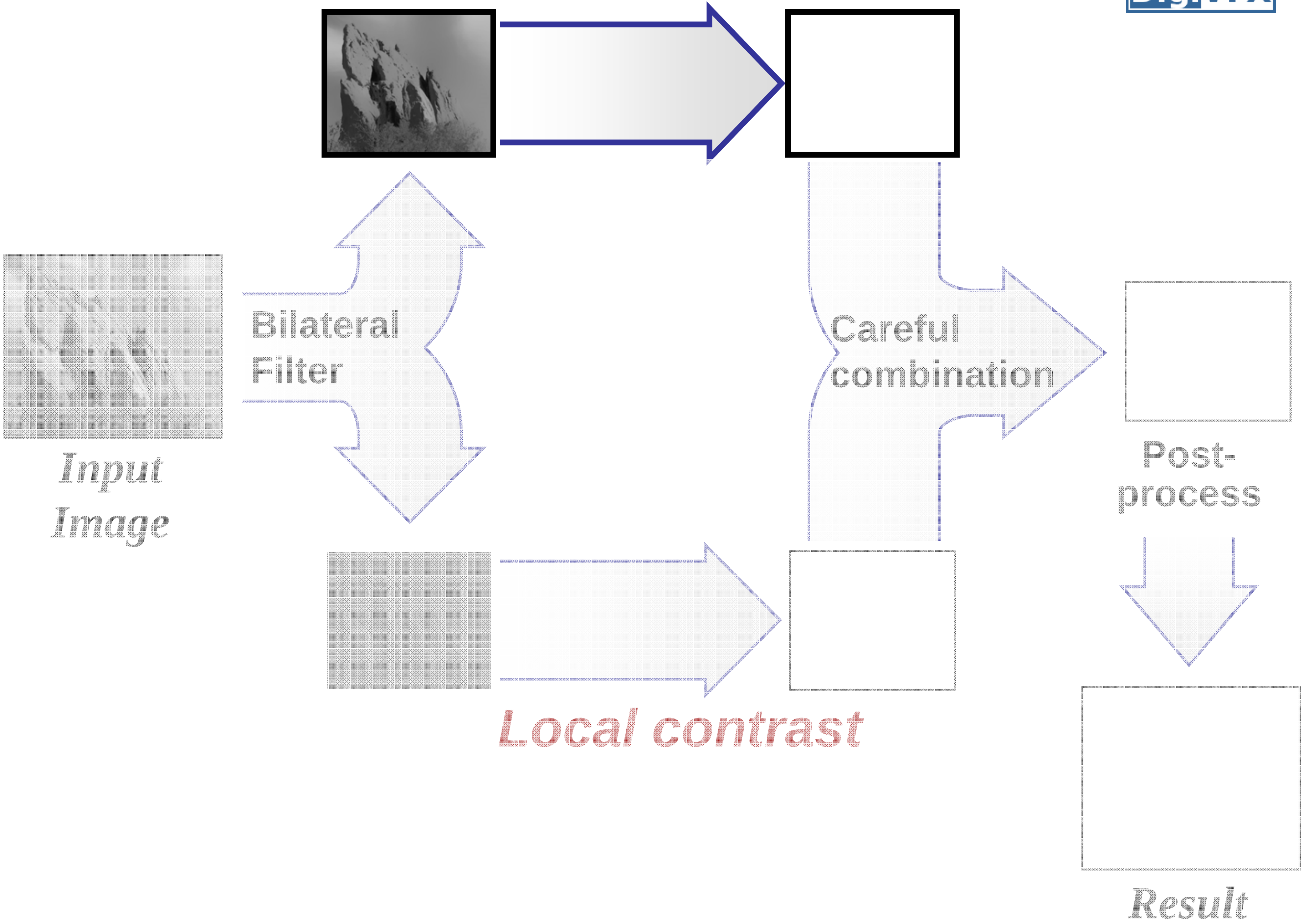
Global contrast

Residual after filtering

Local contrast


Global contrast
DigiVFX
Bilateral Filter
Input Image
Careful combination
Post-process
Local contrast
Result

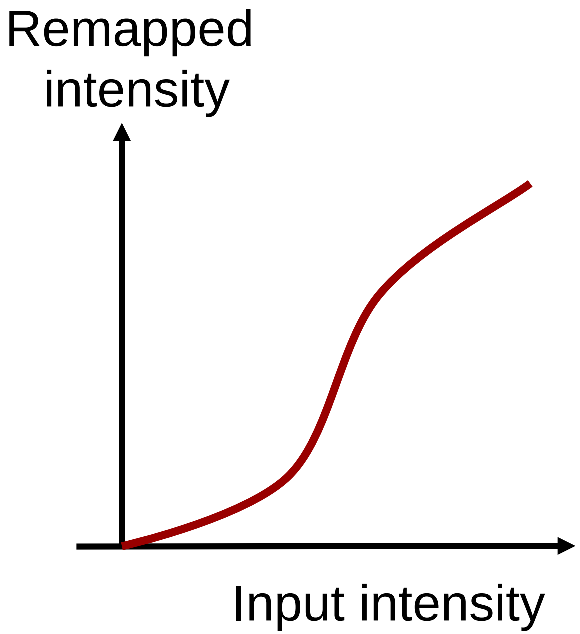
Global contrast



Global Contrast

- Intensity remapping of base layer

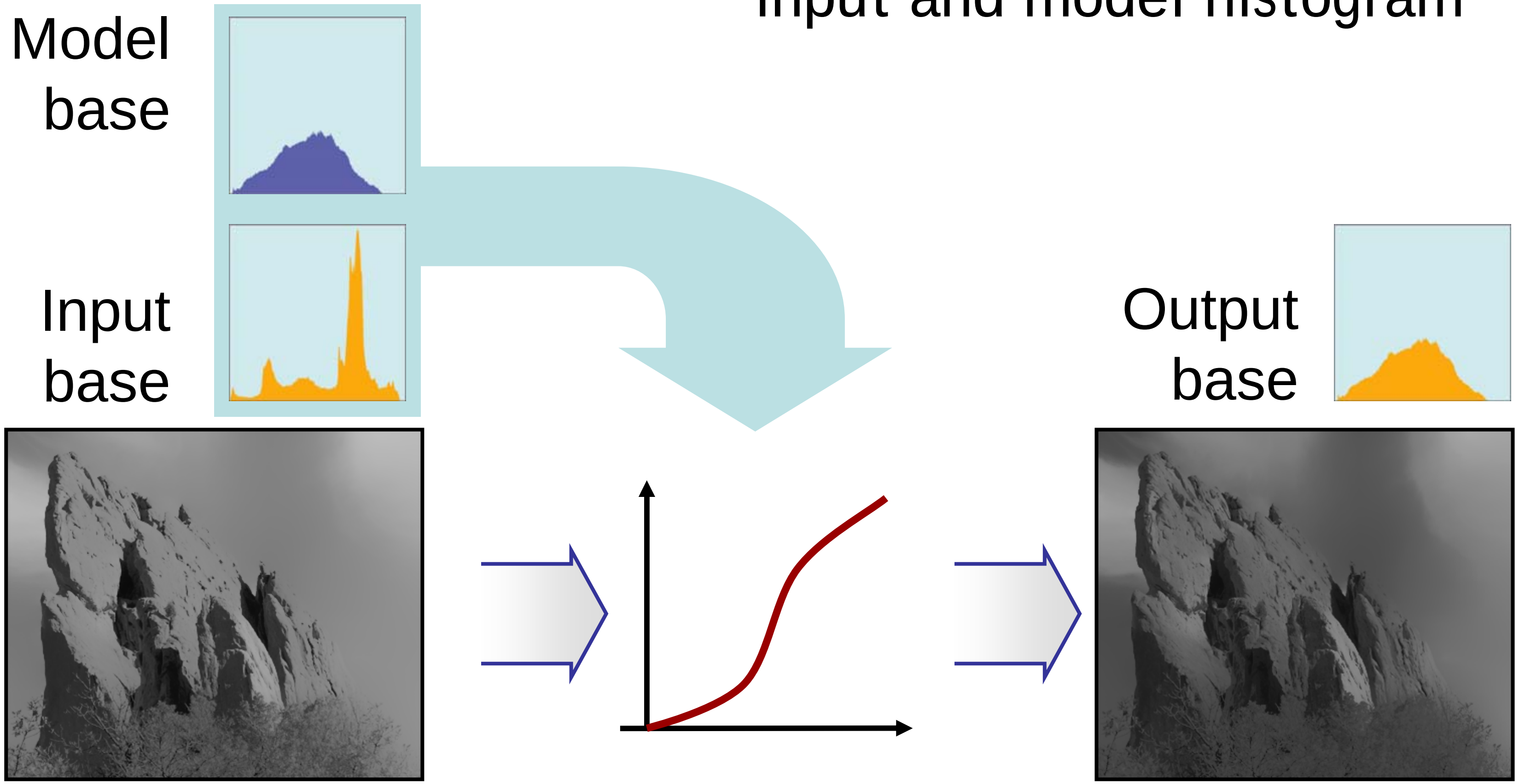
Input base

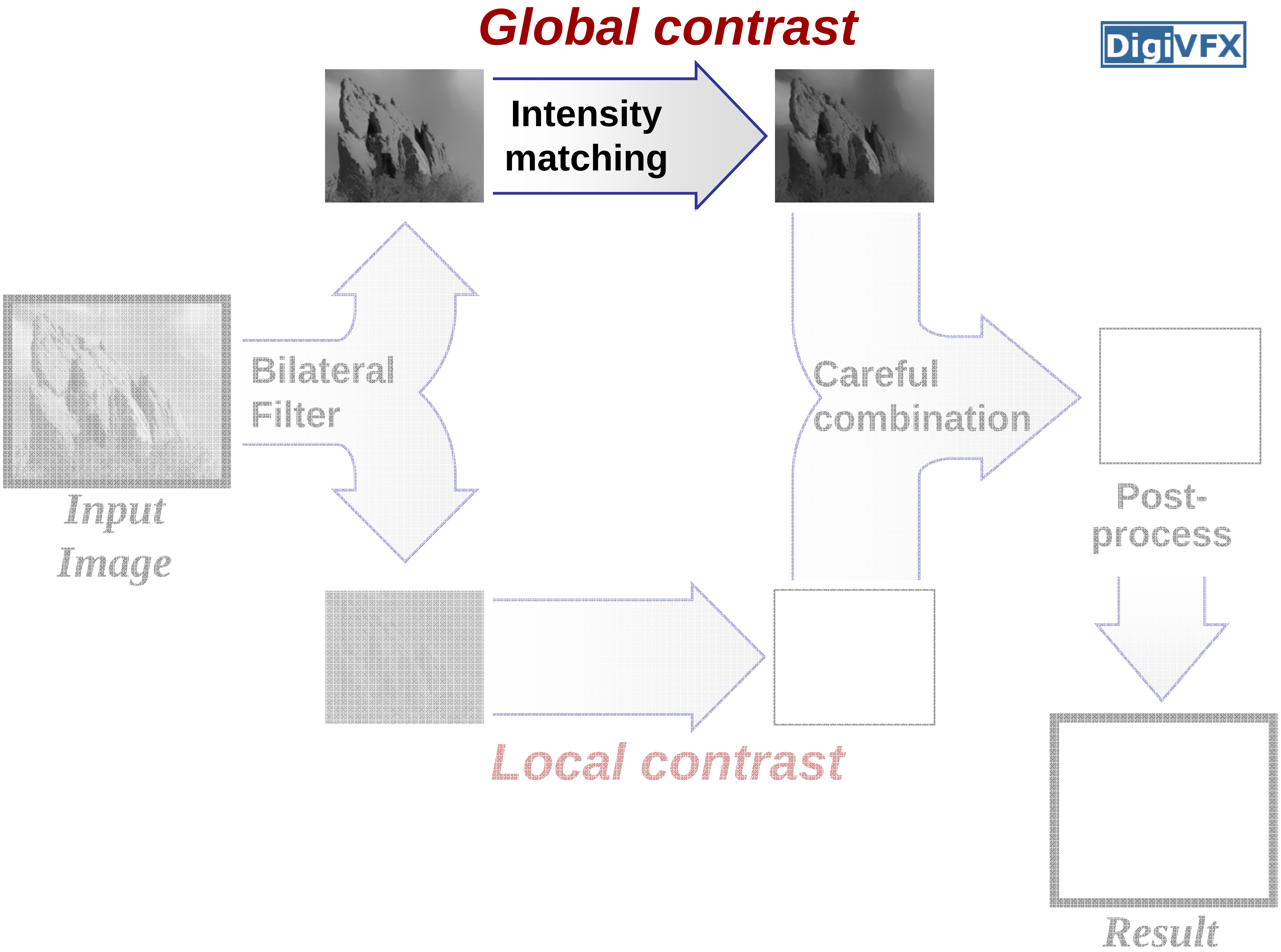


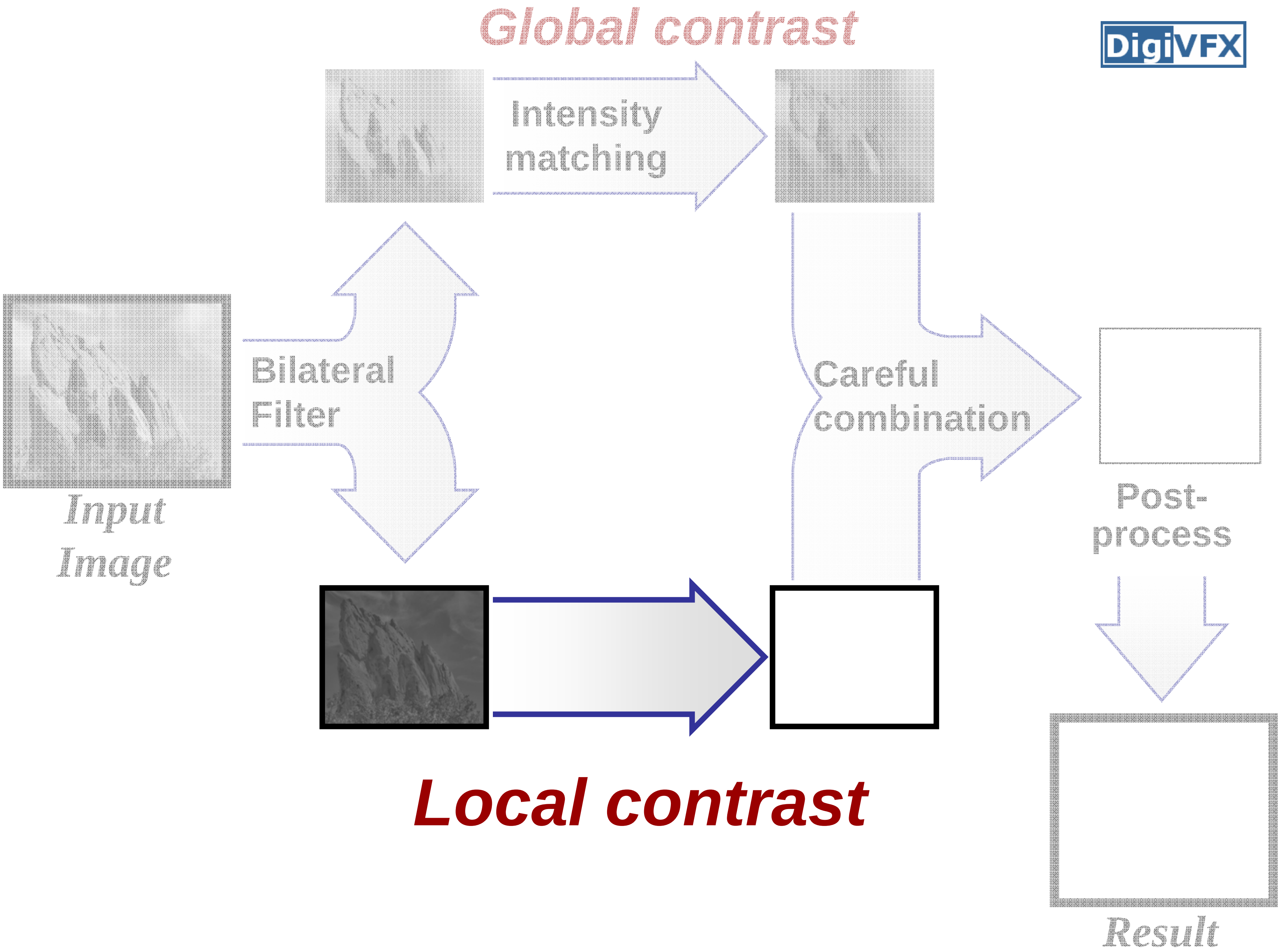
After remapping

Global Contrast (Model Transfer)

- Histogram matching
  - Remapping function given input and model histogram



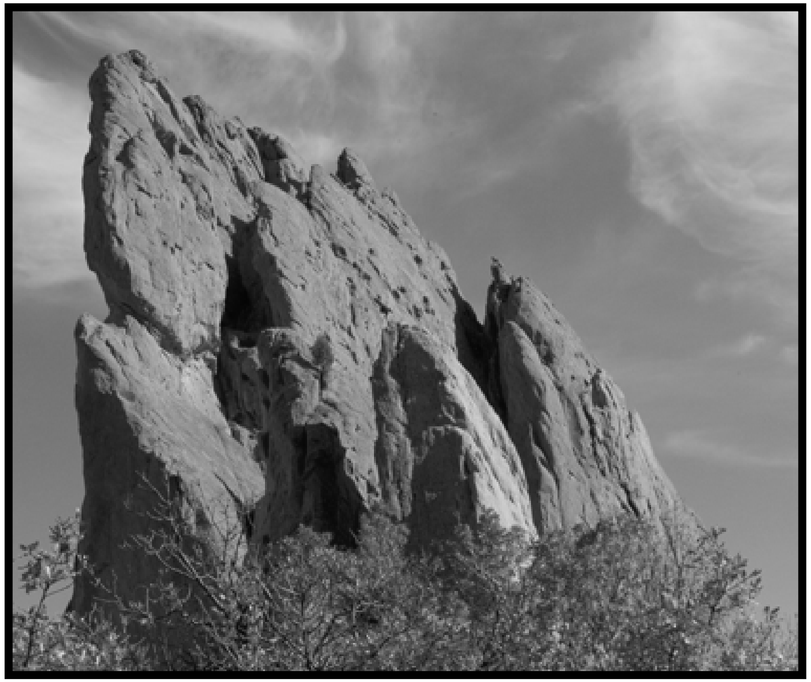

Global contrast
DigiVFX
Intensity matching
Bilateral Filter
Input Image
Careful combination
Post-process
Local contrast
Result

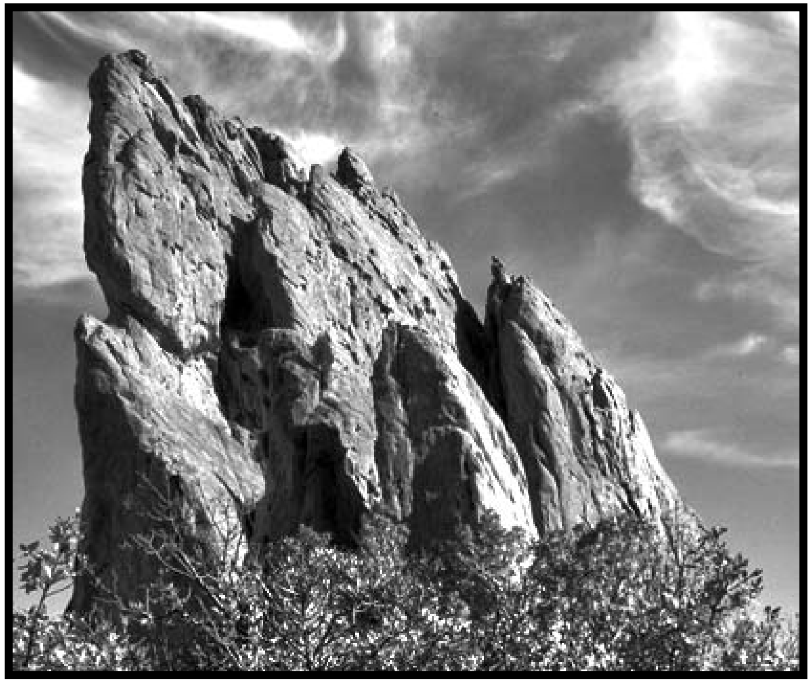


Global contrast
DigiVFX
Intensity matching
Bilateral Filter
Careful combination
Input Image
Post-process
Local contrast
Result

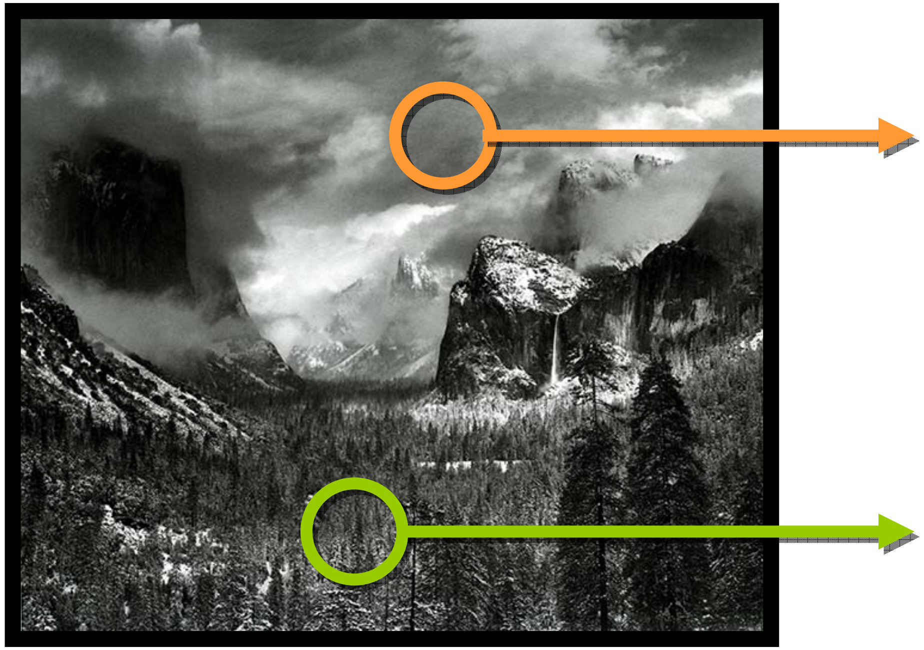
Local Contrast: Detail Layer

- Uniform control:
  - Multiply all values in the detail layer


Input


Base + 3 × Detail

The amount of local contrast is not uniform

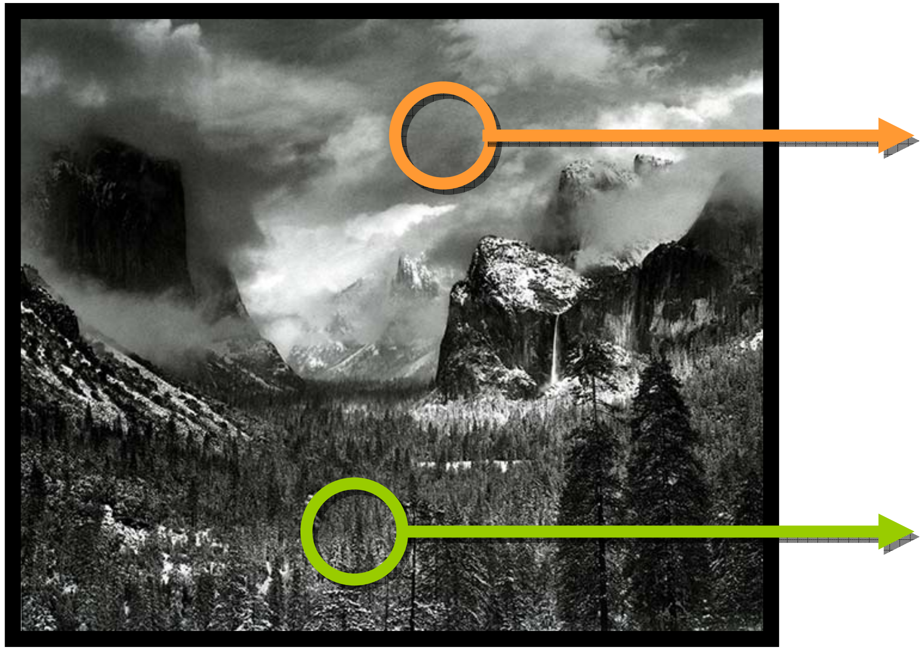


Smooth region

Textured region

Local Contrast Variation

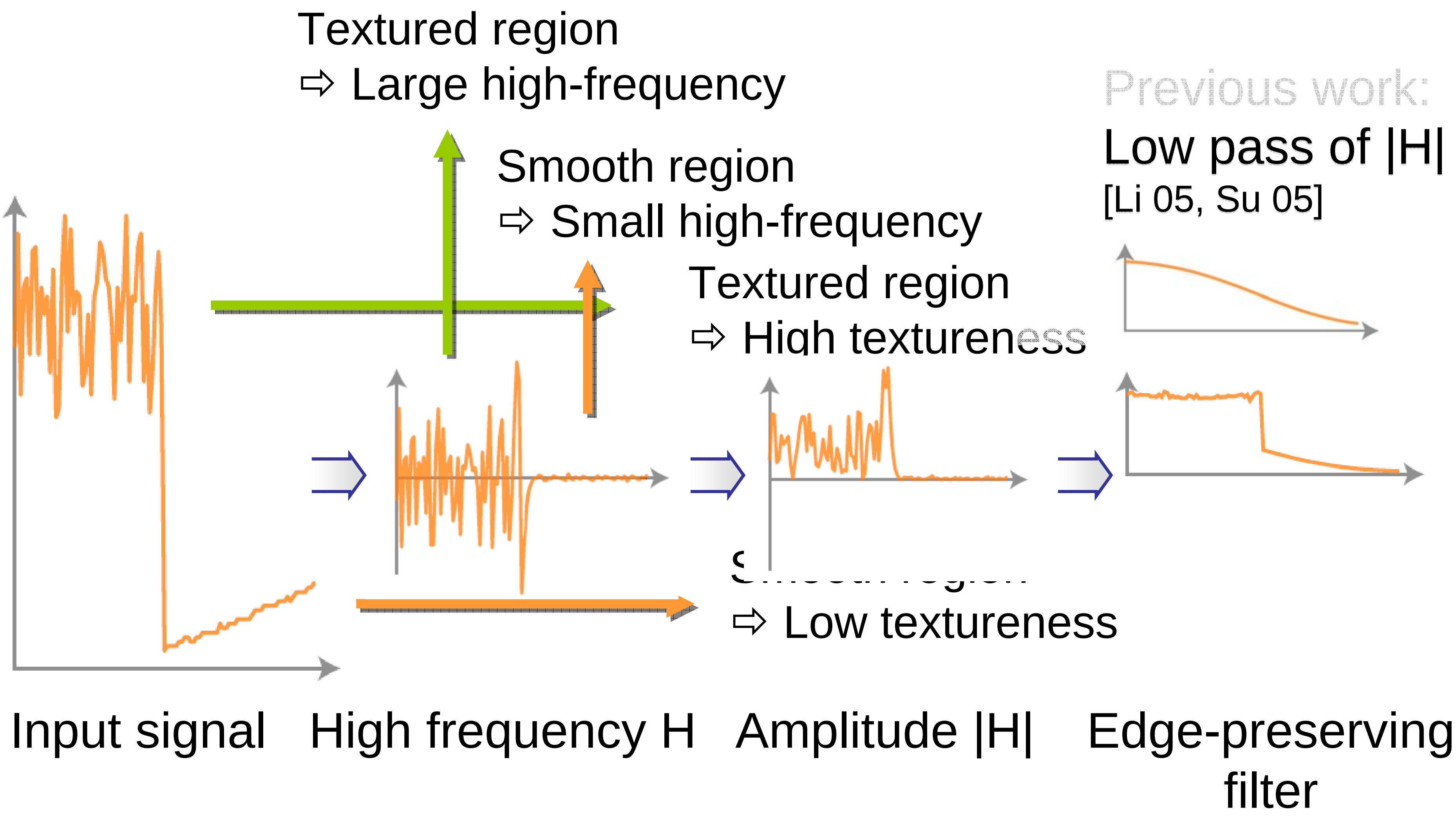
- We define "textureness": amount of local contrast
  - at each pixel based on surrounding region



Smooth region
⇨ Low textureness

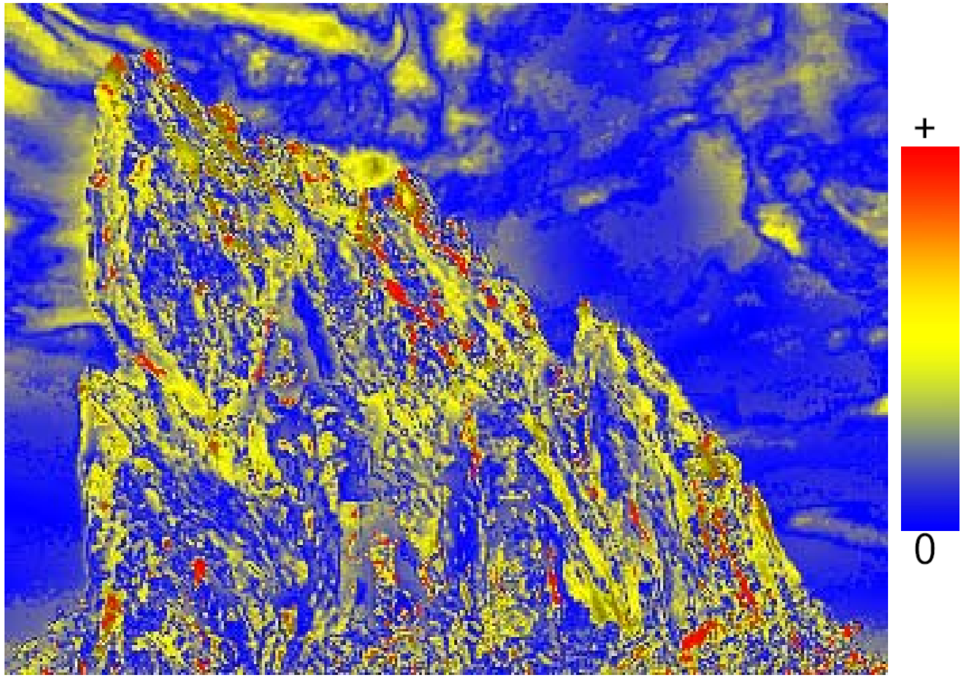
Textured region
⇨ High textureness

“Textureness”: 1D Example



Textureness

Input



Textureness

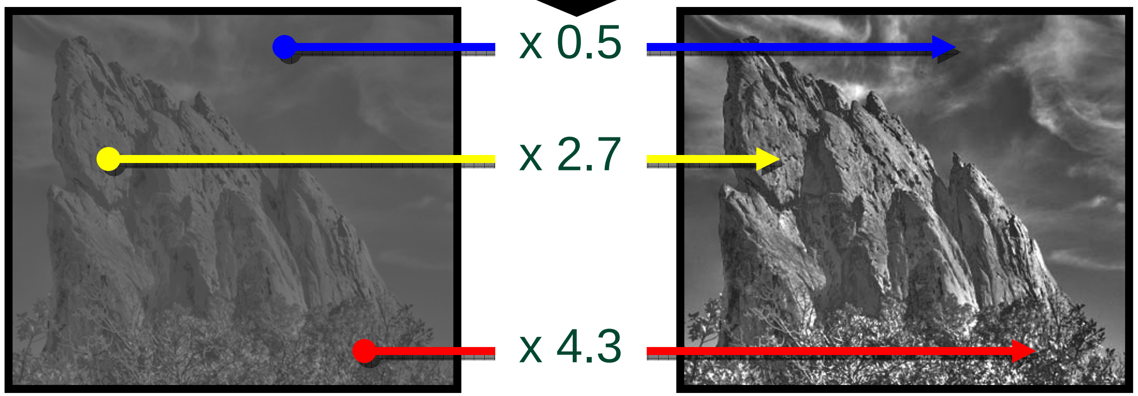
Textureness Transfer

Step 1:
Histogram transfer

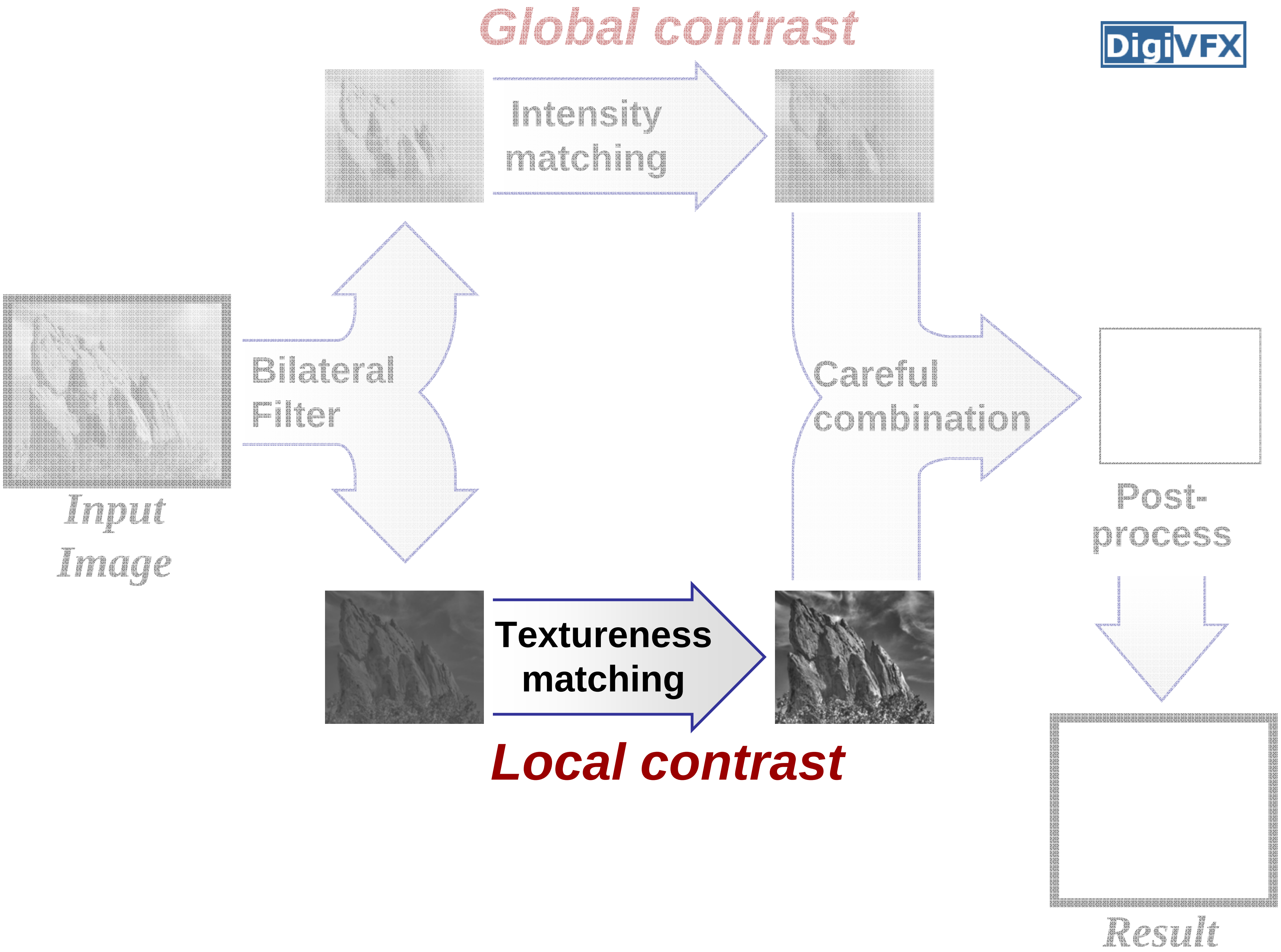
Model textureness

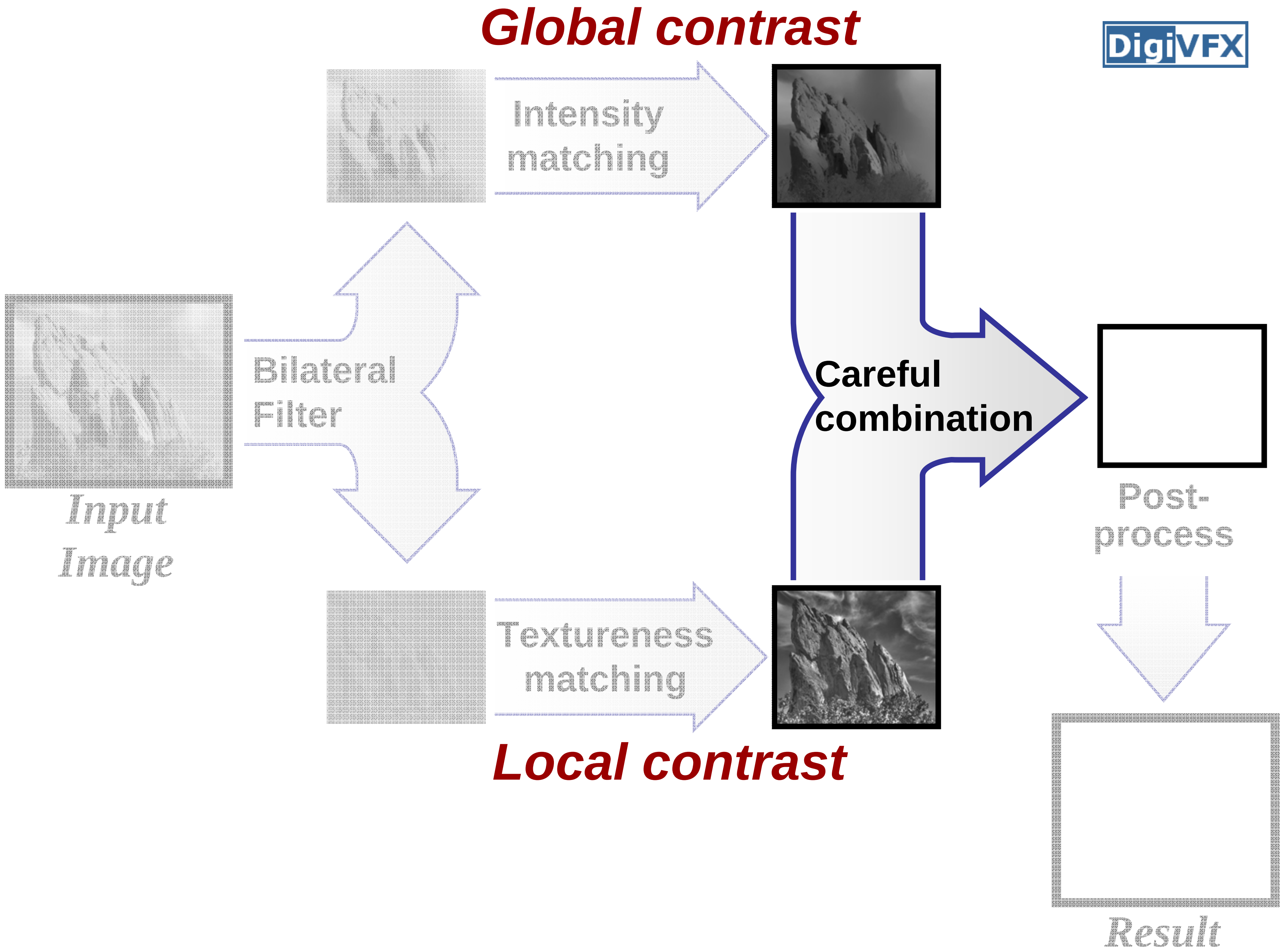
Input textureness → **Hist. transfer** → **Desired textureness**

Step 2:
Scaling detail layer (per pixel) to match desired textureness



Input detail   Output detail


Global contrast
Intensity matching
Bilateral Filter
Input Image
Careful combination
Post-process
Textureness matching
Local contrast
Result


Global contrast
Intensity matching
Input Image
Bilateral Filter
Careful combination
Post-process
Textureness matching
Local contrast
Result

A Non Perfect Result

- Decoupled and large modifications (up to 6x)
  - ➔Limited defects may appear

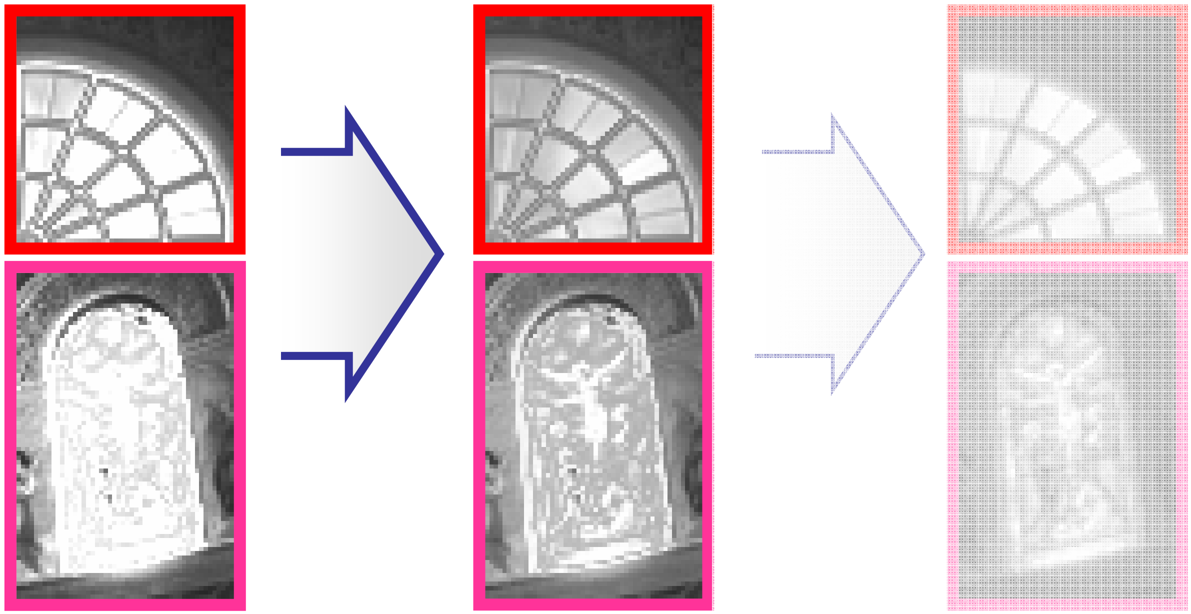
input (HDR)

result after
global and local adjustments

Intensity Remapping

- Some intensities may be outside displayable range.

→ Compress histogram to fit visible range.



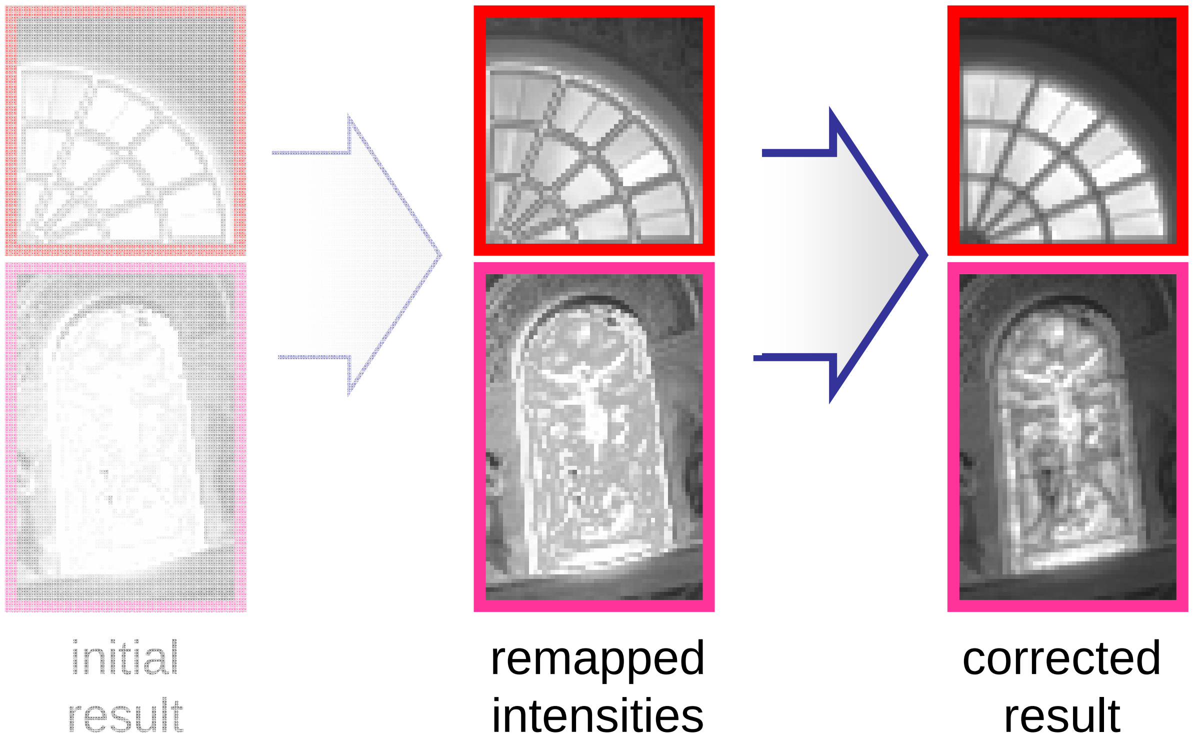
initial result

remapped intensities

corrected result

Preserving Details

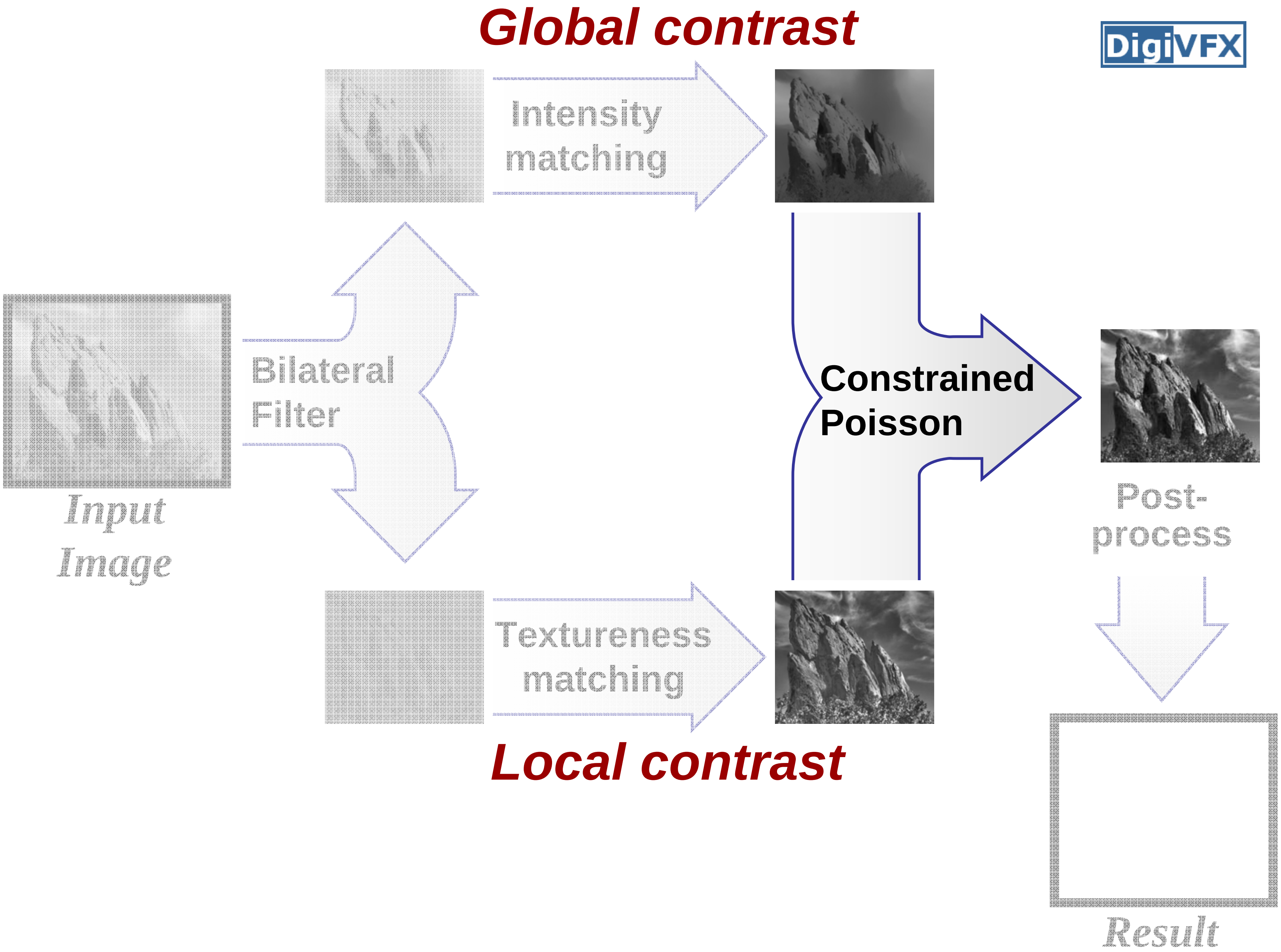
1. In the **gradient domain**:
   - Compare gradient amplitudes of input and current
   - Prevent extreme reduction & extreme increase
2. Solve the **Poisson equation**.

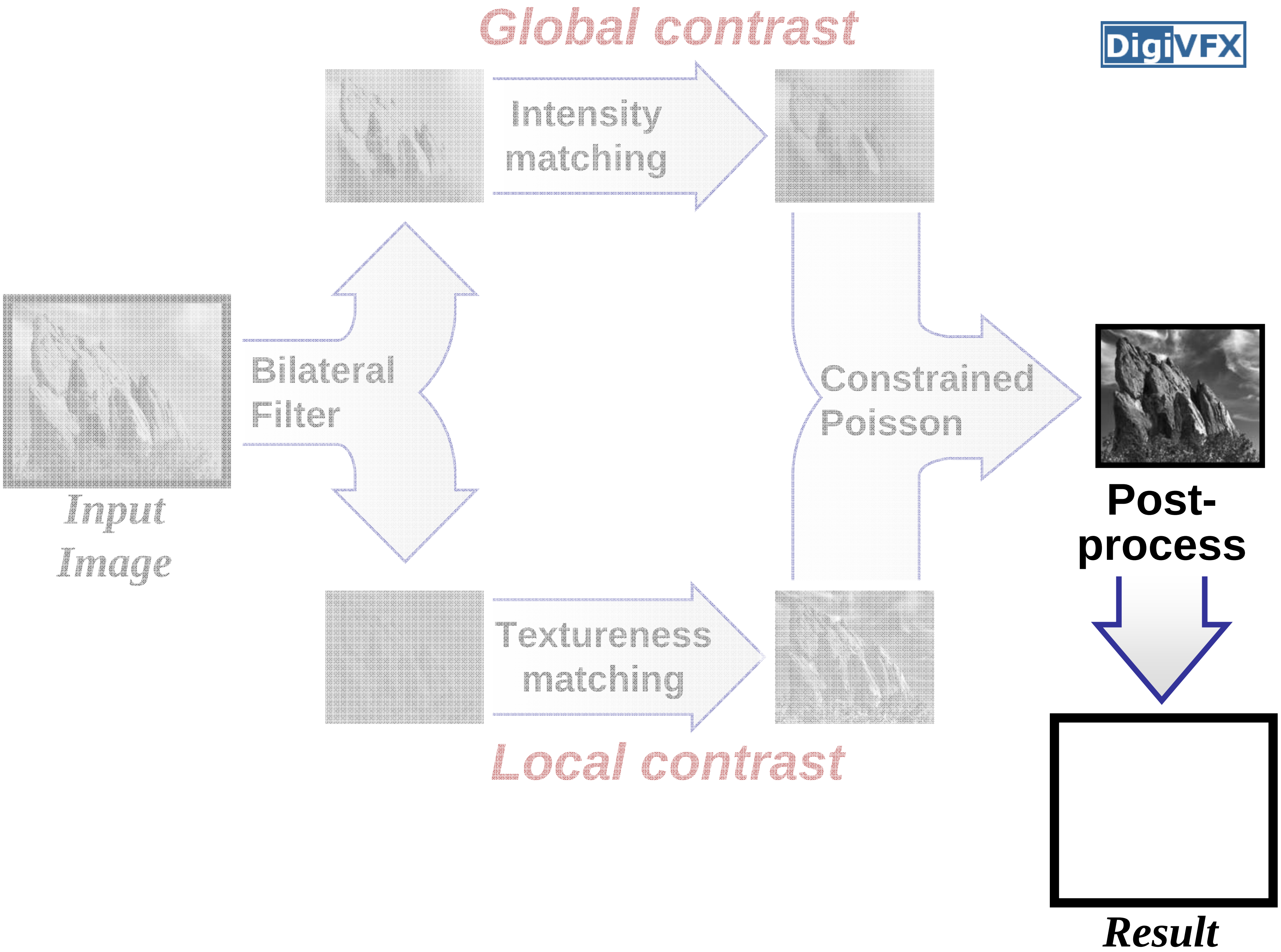


Effect of Detail Preservation

uncorrected result

corrected result


Global contrast
Intensity matching
Input Image
Bilateral Filter
Constrained Poisson
Post-process
Textureness matching
Local contrast
Result



Global contrast
Intensity matching
Bilateral Filter
Input Image
Constrained Poisson
Post-process
Textureness matching
Local contrast
Result

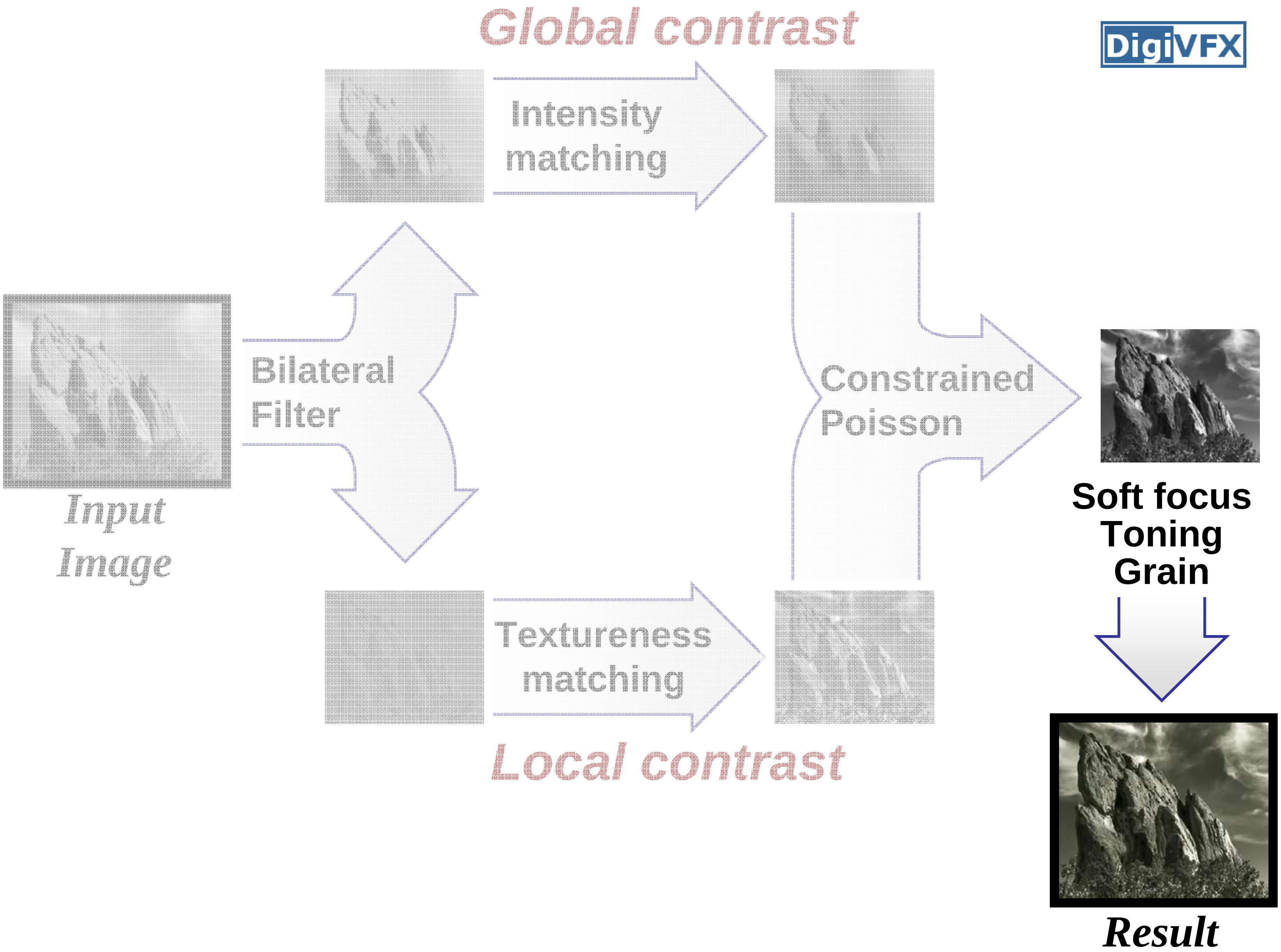
Additional Effects

model

- **Soft focus** (high frequency manipulation)
- **Film grain** (texture synthesis [Heeger 95])
- **Color toning** (chrominance = f (luminance))

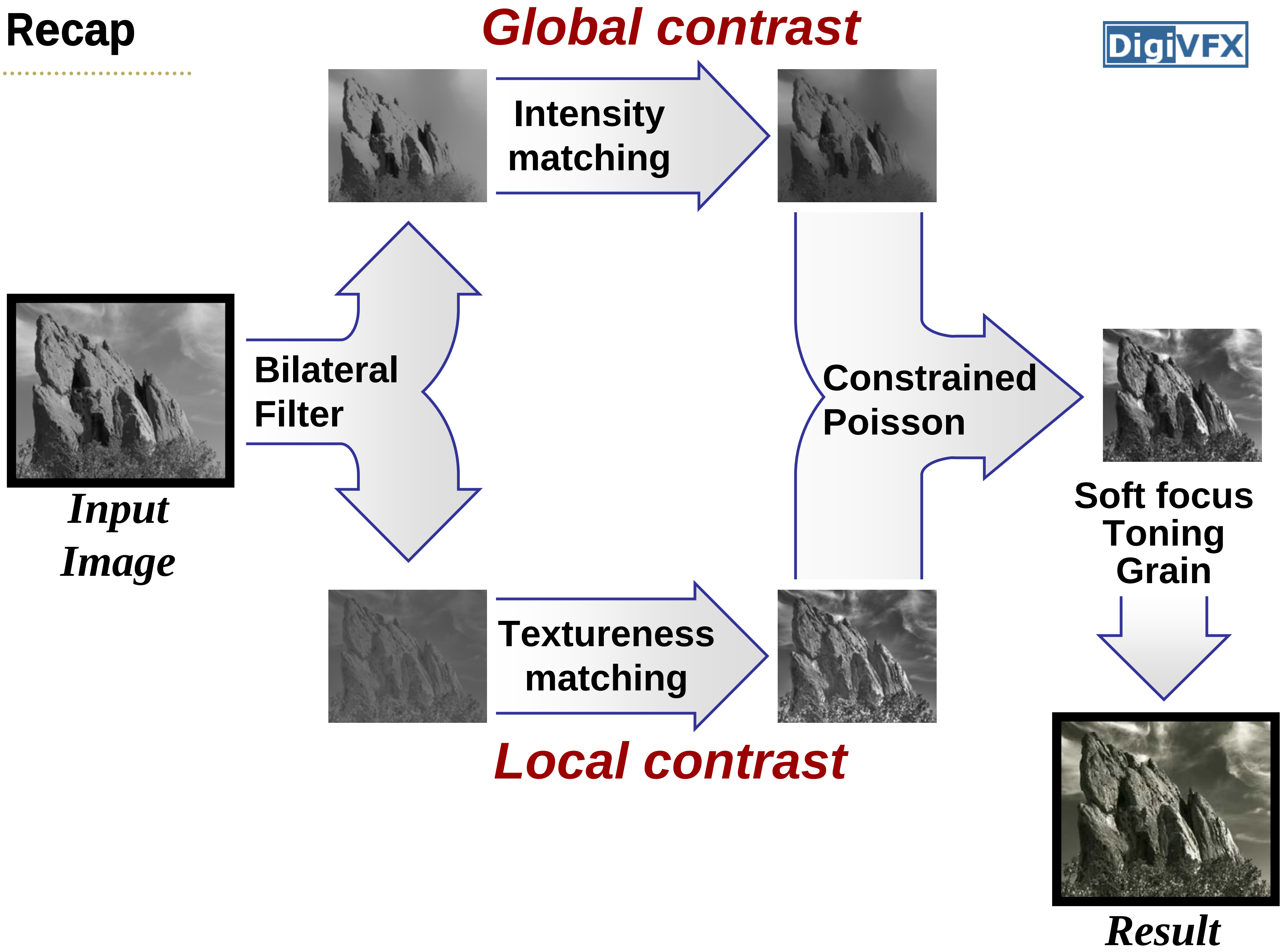
before
effects

after
effects



Global contrast
Intensity matching
Input Image
Bilateral Filter
Constrained Poisson
Soft focus
Toning
Grain
Textureness matching
Local contrast
Result

Recap


Global contrast
Intensity matching
Bilateral Filter
Input Image
Constrained Poisson
Soft focus
Toning
Grain
Textureness matching
Local contrast
Result

Results

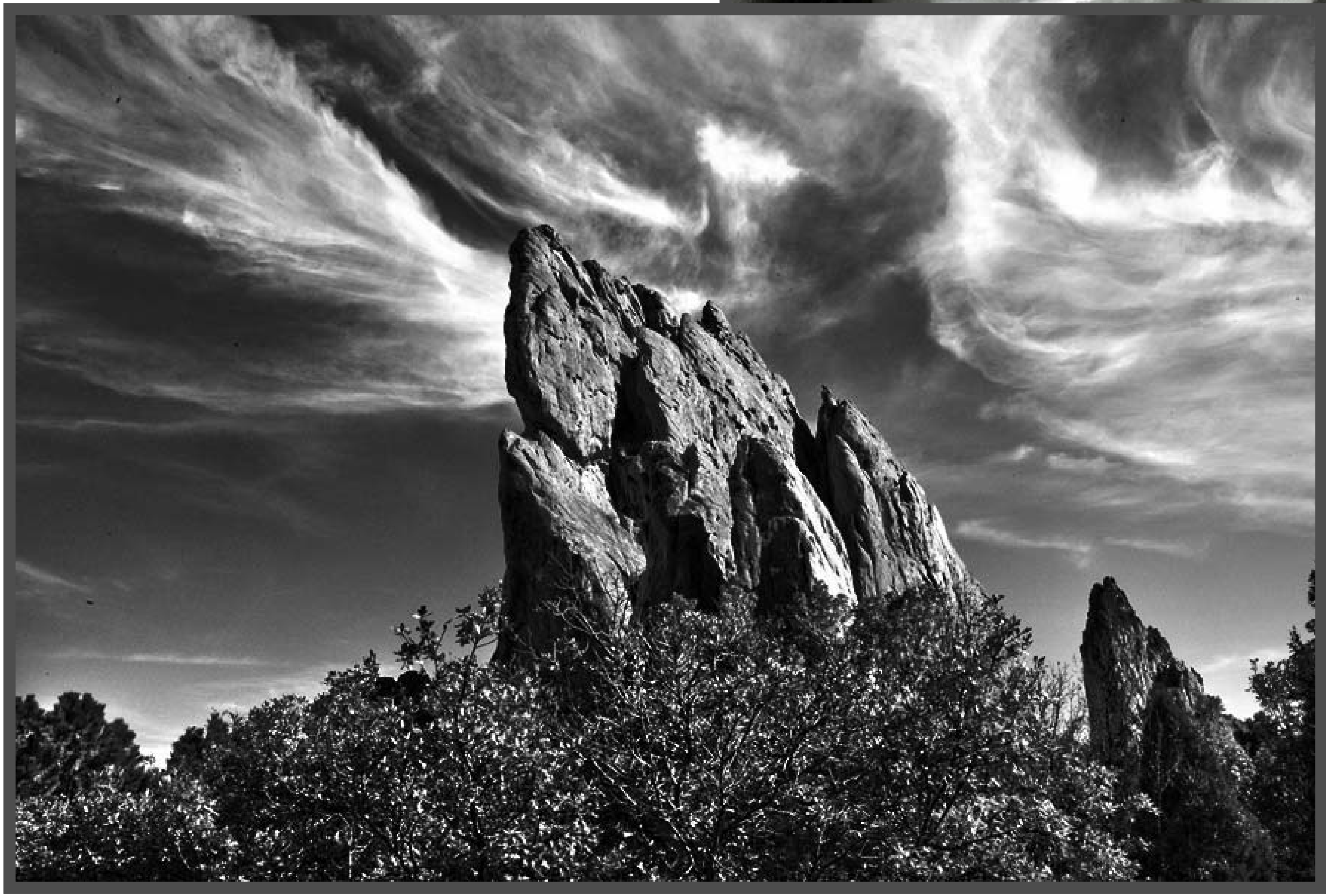
User provides input and model photographs.

➔ Our system **automatically** produces the result.

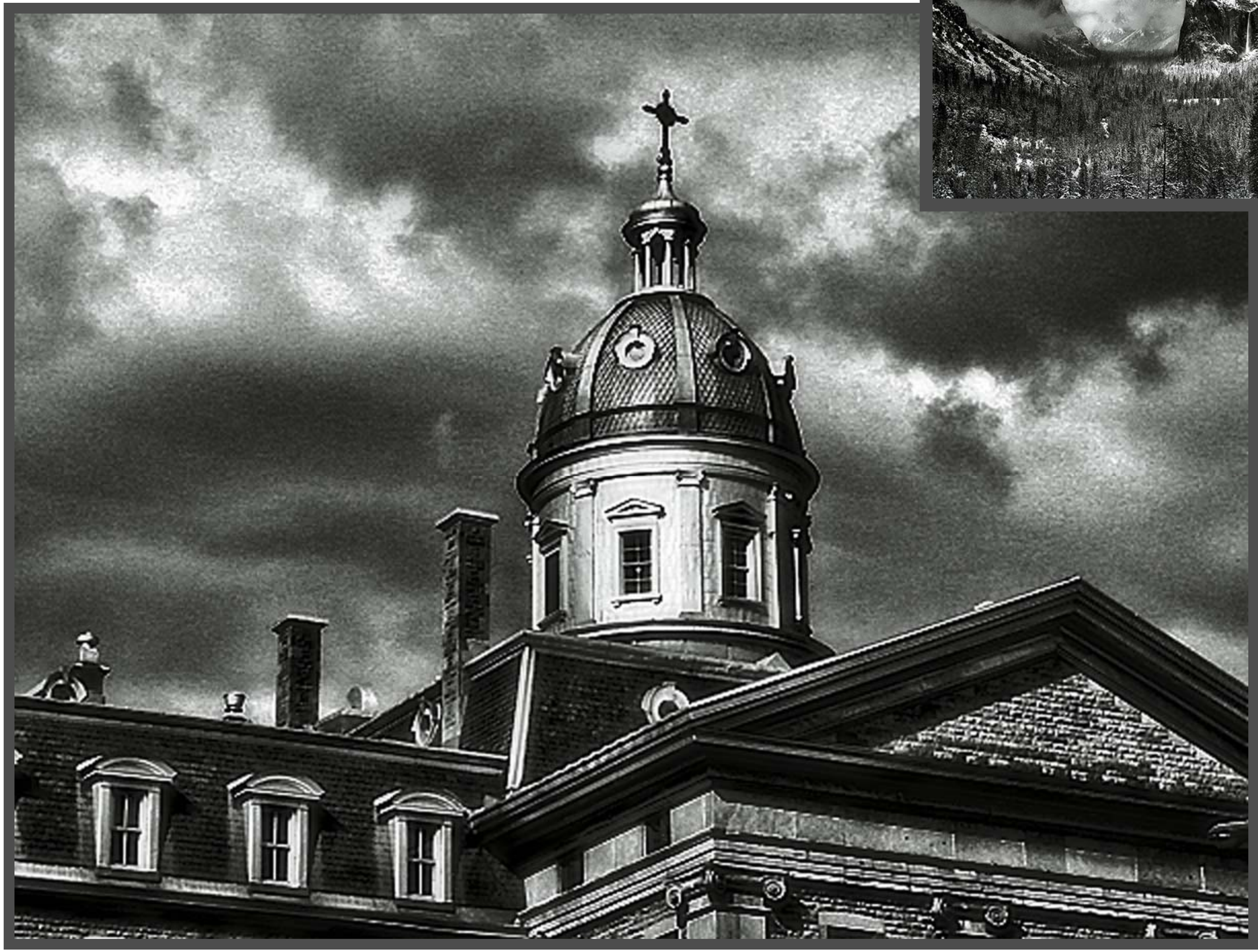
Running times:

- 6 seconds for 1 MPixel or less
- 23 seconds for 4 MPixels
- multi-grid Poisson solver and fast bilateral filter [Paris 06]

Result

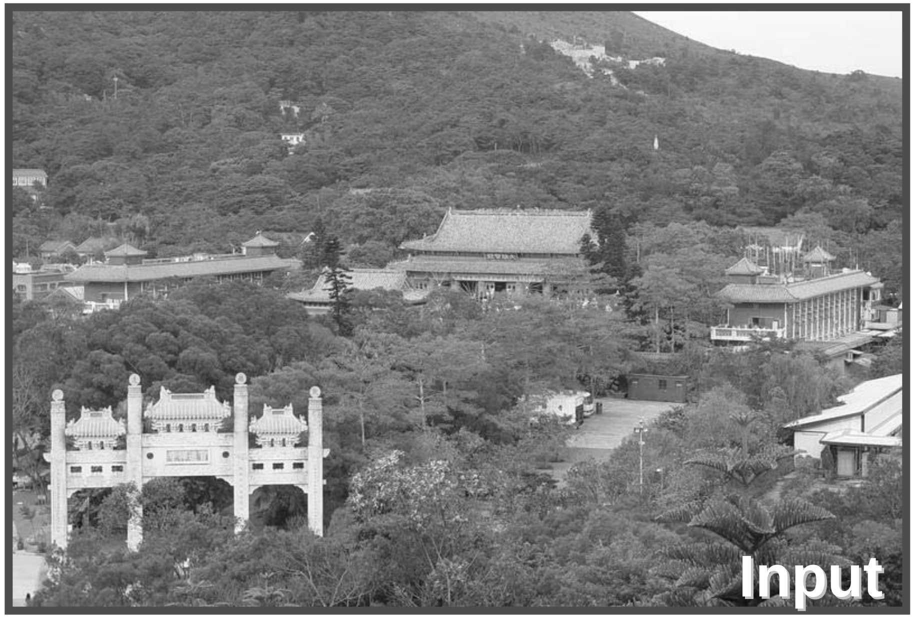


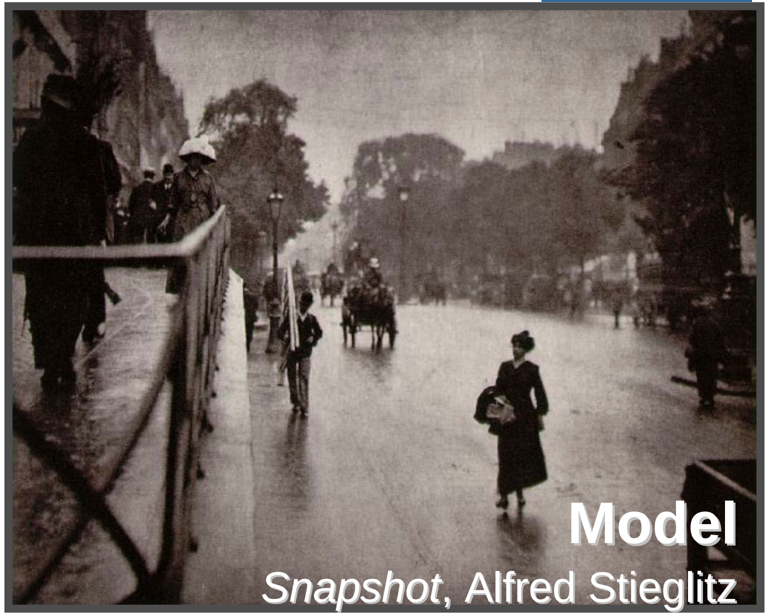
Input / Result

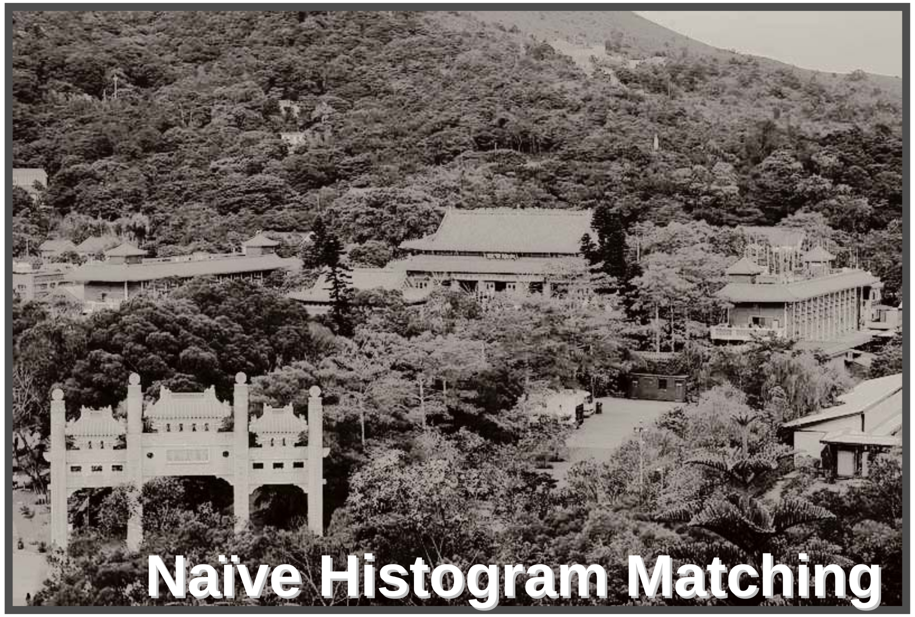


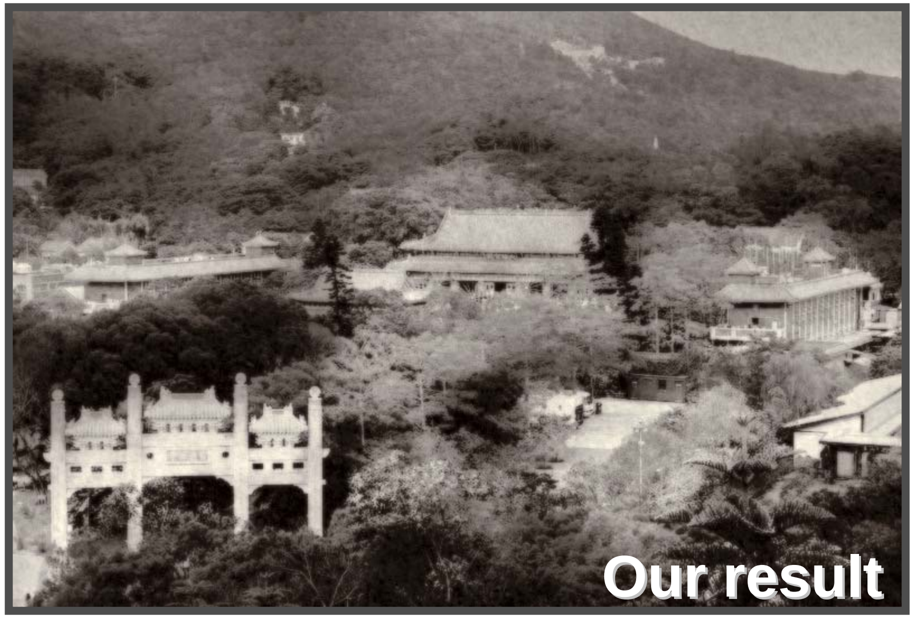
Result

Comparison with Naïve Histogram Matching



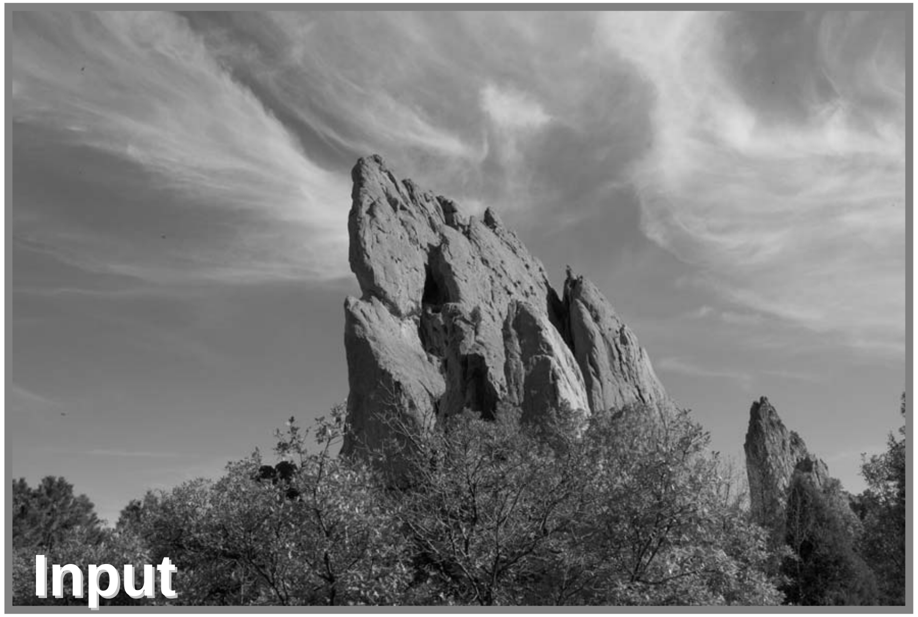


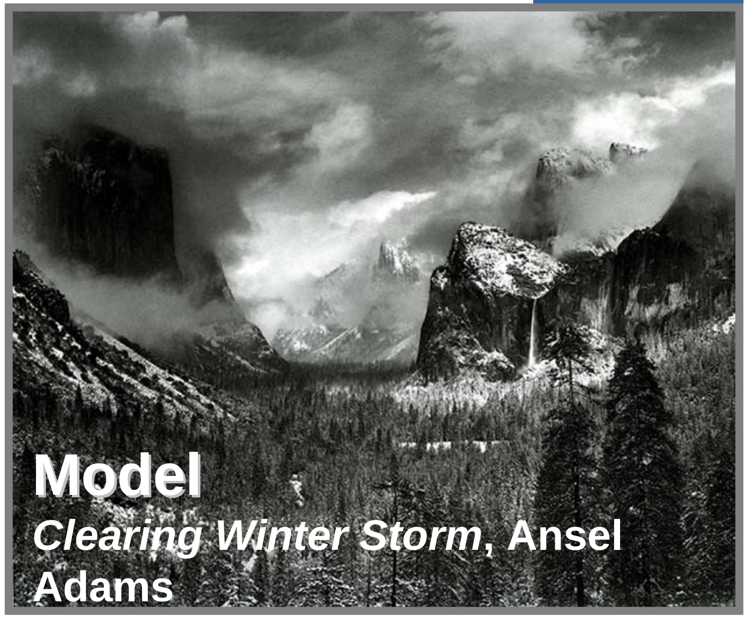




Local contrast, sharpness unfaithful

Comparison with Naïve Histogram Matching



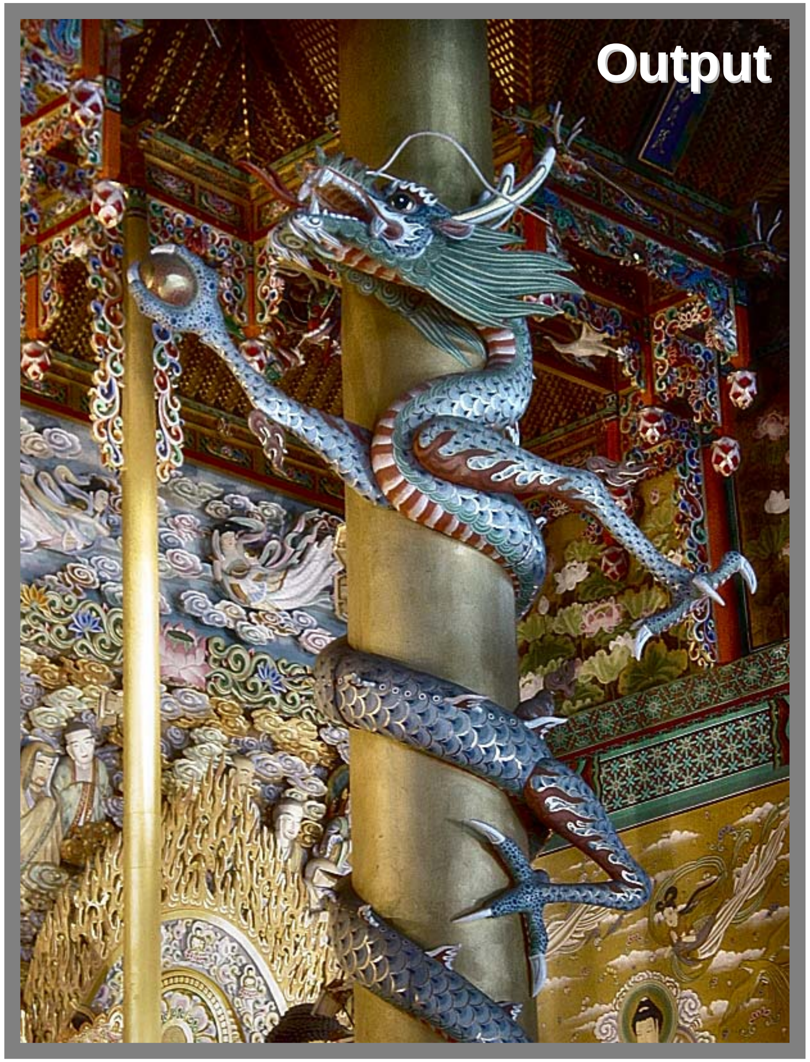


Local contrast too low

Color Images

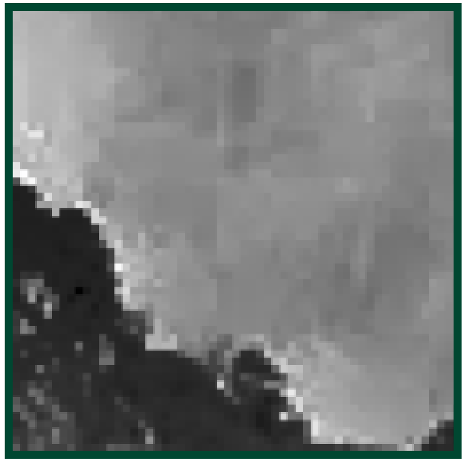
DigiVFX

- **Lab color space: modify only luminance**



Limitations

- Noise and JPEG artifacts
  - amplified defects



- Can lead to unexpected results if the image content is too different from the model
  - Portraits, in particular, can suffer

Conclusions

- Transfer “look” from a model photo
- Two-scale tone management
  - Global and local contrast
  - New edge-preserving textureness
  - Constrained Poisson reconstruction
  - Additional effects

References

- Patrick Perez, Michel Gangnet, Andrew Blake, Poisson Image Editing, SIGGRAPH 2003.
- Dani Lischinski, Zeev Farbman, Matt Uytendaelle and Richard Szeliski. Interactive Local Adjustment of Tonal Values. SIGGRAPH 2006.
- Carsten Rother, Andrew Blake, Vladimir Kolmogorov, GrabCut - Interactive Foreground Extraction Using Iterated Graph Cuts, SIGGRAPH 2004.
- Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David H. Salesin, Michael F. Cohen, Interactive Digital Photomontage, SIGGRAPH 2004.
- Sylvain Paris and Fredo Durand. A Fast Approximation of the Bilateral Filter using a Signal Processing Approach. ECCV 2006.
- Soonmin Bae, Sylvain Paris and Fredo Durand. Two-scale Tone Management for Photographic Look. SIGGRAPH 2006.