BIL 717
Image Processing
Mar. 25, 2015

Modern Image Smoothing

Erkut Erdem
Hacettepe University
Computer Vision Lab (HUCVL)

Acknowledgement: The slides on Rolling Guidance Filter are adapted from the slides prepared by Zhang et al. of The Chinese University of Hong Kong.

A little bit of history

- Gaussian Filtering / linear diffusion - the most widely used method

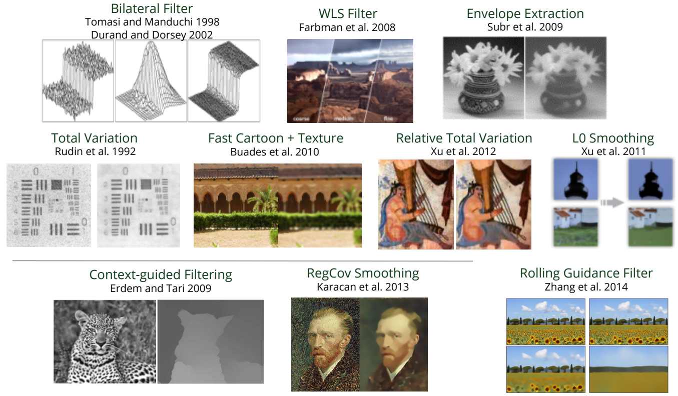
- mid 80's – unified formulations – a breakthrough!
  - methods that combine smoothing and edge detection (Geman & Geman'84, Mumford & Shah'89, Perona & Malik'90)

A little bit of (personal) history

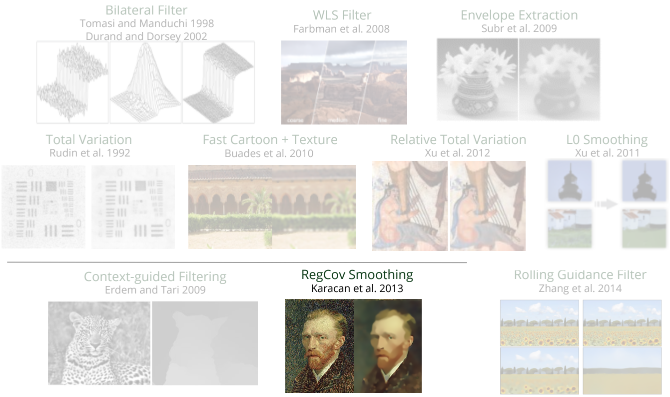
Standard unified formulations (nonlinear filters) fail to capture some details, e.g. due to texture!

- mid 80's – unified formulations – a breakthrough!
  - methods that combine smoothing and edge detection (Geman & Geman'84, Mumford & Shah'89, Perona & Malik'90)

Some seminal works



Some seminal works



Context-guided filtering

- Contextual knowledge extracted from local image regions guides the regularization process.
  - E. Erdem, A. Sancar-Yilmaz, and S. Tari, "Mumford-Shah Regularizer with Spatial Coherence", In SSVM 2007
  - E. Erdem and S. Tari, "Mumford-Shah Regularizer with Contextual Feedback", JMIV 2009

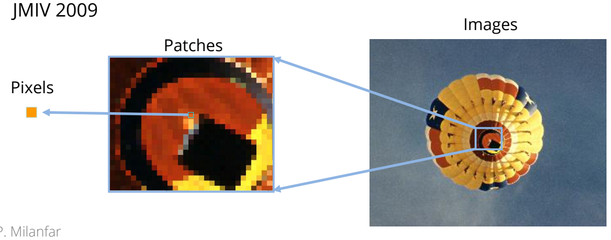


Image Credit: P. Milanfar

Structure-Texture Decomposition

- Decomposing an image into structure and texture components

Input Image

Structure-Texture Decomposition

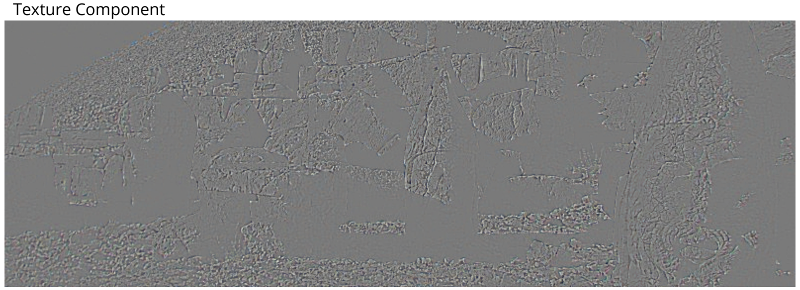
- Decomposing an image into structure and texture components

Structure Component

Structure-Texture Decomposition

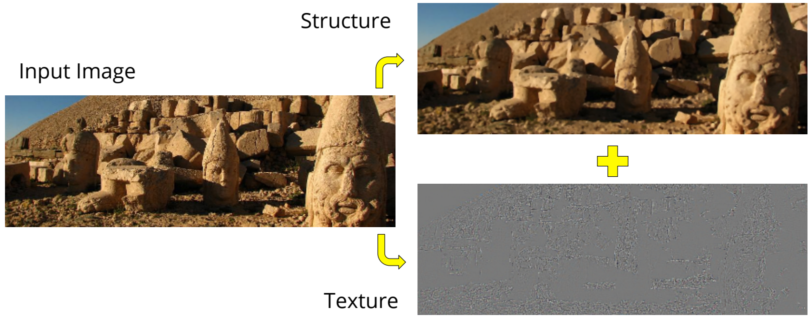
- Decomposing an image into structure and texture components

Texture Component



Structure-Texture Decomposition

- Decomposing an image into structure and texture components

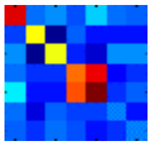


Region Covariances as Region Descriptors

Tuzel et al., ECCV 2006

$$F(x,y)=\phi(I,x,y)$$

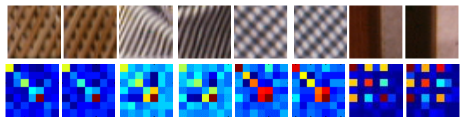
$$F(x,y) = \left[I(x,y) \quad \left|\frac{\partial I}{\partial x}\right| \quad \left|\frac{\partial I}{\partial y}\right| \quad \left|\frac{\partial^2 I}{\partial x^2}\right| \quad \left|\frac{\partial^2 I}{\partial y^2}\right| \quad x \quad y \right]^T$$



$$\mathbf{C}_R = \frac{1}{n-1}\sum_{i=0}^{n}(\mathbf{z}_k - \mu)(\mathbf{z}_k - \mu)^T$$

Motivation

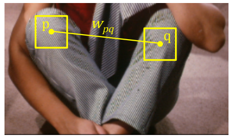
- Region covariances capture local structure and texture information.
- Similar regions have similar statistics.



RegCov Smoothing - Formulation

$$I = S + T$$

$$S(\mathbf{p}) = \frac{1}{Z_{\mathbf{p}}} \sum_{\mathbf{q} \in N(\mathbf{p}, r)} w_{\mathbf{pq}} I(\mathbf{q})$$



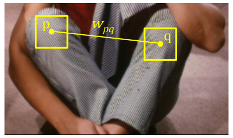
- Structure-texture decomposition via smoothing
- Smoothing as weighted averaging
- Different kernels (w_{pq}) result in different types of filters.
- Three novel patch-based kernels for structure texture decomposition.
- L. Karacan, A. Erdem, E. Erdem, “Structure Preserving Image Smoothing via Region Covariances”, ACM TOG 2013 (SIGGRAPH Asia 2013)

Model 1

- Depends on sigma-points representation of covariance matrices (Hong et al.,CVPR’09)

$\mathbf{C} = \mathbf{L}\mathbf{L}^T$ *Cholesky Decomposition* $\mathcal{S} = \{\mathbf{s}_i\}$ *Sigma Points*

$$\mathbf{s}_i = \begin{cases} \alpha\sqrt{d}\mathbf{L}_i & \text{if } 1 \leq i \leq d \\ -\alpha\sqrt{d}\mathbf{L}_i & \text{if } d+1 \leq i \leq 2d \end{cases}$$



Final representation

$$\Psi(\mathbf{C}) = (\mu, \mathbf{s}_1, \ldots, \mathbf{s}_d, \mathbf{s}_{d+1}, \ldots, \mathbf{s}_{2d})^T$$

Resulting kernel function

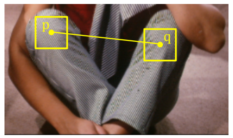
$$w_{\mathbf{pq}} \propto \exp\left(-\frac{\|\Psi(\mathbf{C_p}) - \Psi(\mathbf{C_q})\|^2}{2\sigma^2}\right)$$

Model 2

- An alternative way is to use statistical similarity measures.
- A Mahalanobis-like distance measure to compare to image patches.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(\mu_{\mathbf{p}} - \mu_{\mathbf{q}})\mathbf{C}^{-1}(\mu_{\mathbf{p}} - \mu_{\mathbf{q}})^T}$$

$$\mathbf{C} = \mathbf{C_p} + \mathbf{C_q}$$

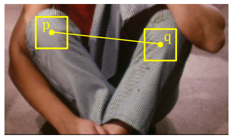


Resulting kernel $w_{\mathbf{pq}} \propto \exp\left(-\frac{d(\mathbf{p}, \mathbf{q})^2}{2\sigma^2}\right)$

Model 3

resulted from a discussion with Rahul Narain (Berkeley University)

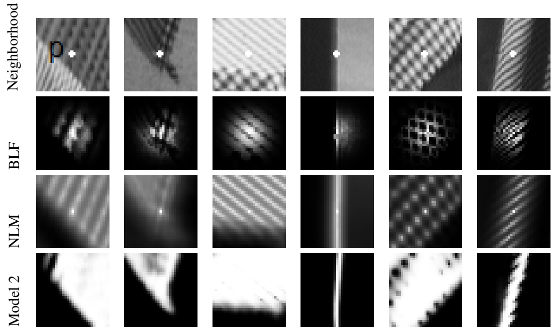
- We use Kullback-Leibler(KL)-Divergence measure from probability theory.
- A KL-Divergence form is used to calculate statistical distance between two multivariate normal distribution

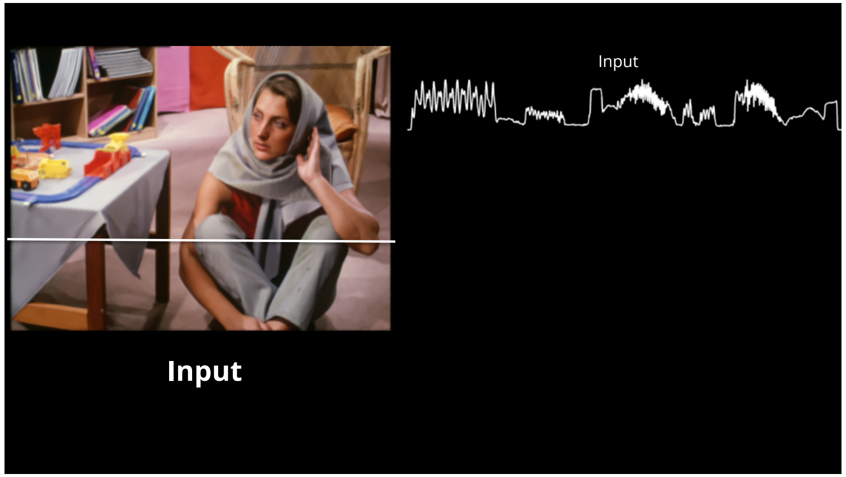


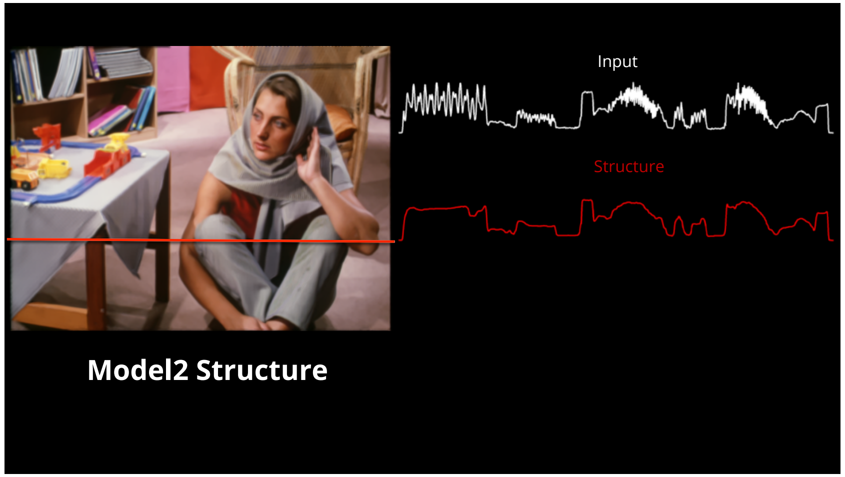
$$d_{KL}(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\left(tr(\mathbf{C_q}^{-1}\mathbf{C_p}) + (\mu_p - \mu_q)^T \mathbf{C_q}^{-1}(\mu_p - \mu_q) - k - ln\left(\frac{\det \mathbf{C_p}}{\det \mathbf{C_q}}\right)\right)$$

Resulting kernel $w_{pq} \propto \frac{d_{KL}(\mathbf{p}, \mathbf{q})}{2\sigma^2}$

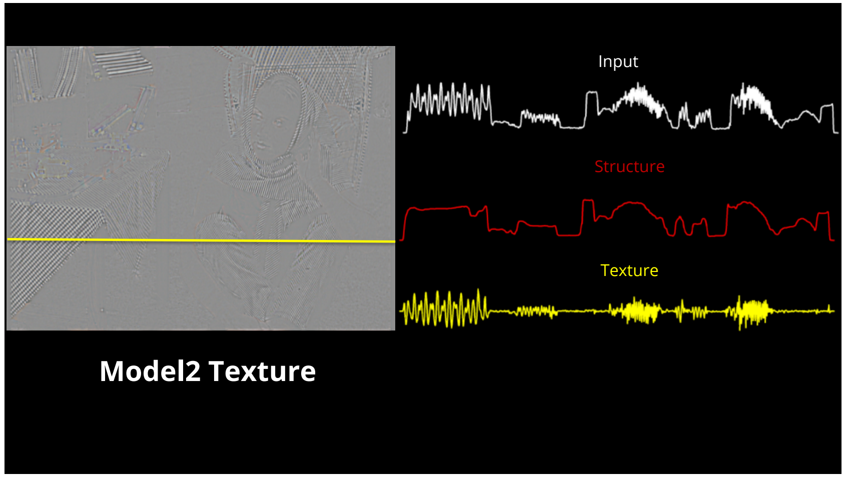
Smoothing Kernels



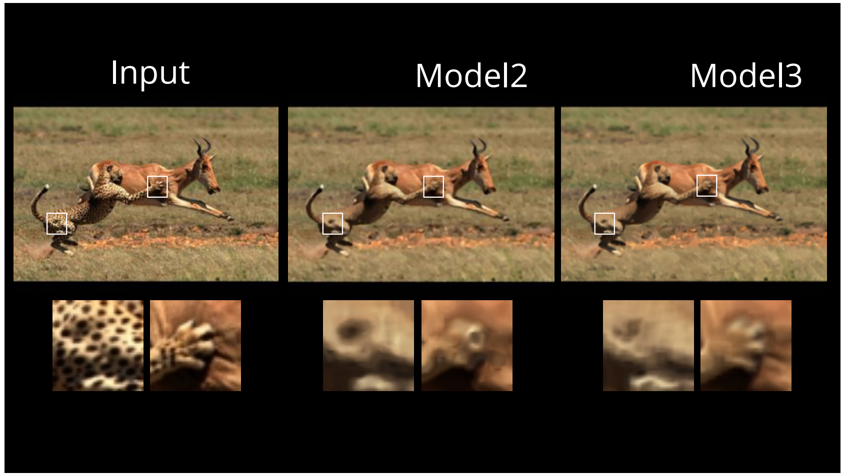




Input
Structure
Model2 Structure



Input
Structure
Texture
Model2 Texture



Input
Model2
Model3

Input

TV
Rudin et al.
1992

BLF
1998

WLS
Farbman et al.
2008

Envelope Extraction
Subr et al.
2009


Buades et al.
2010

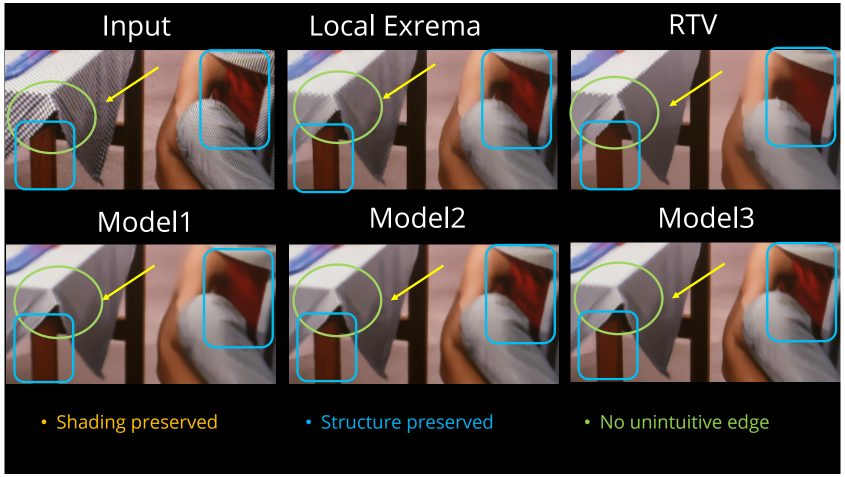

L0
Xu et al.
2011

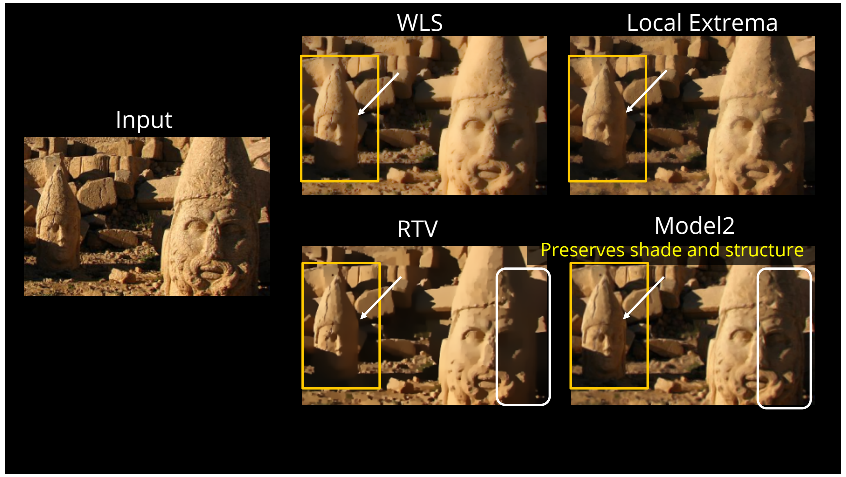

RTV
Xu et al.
2012

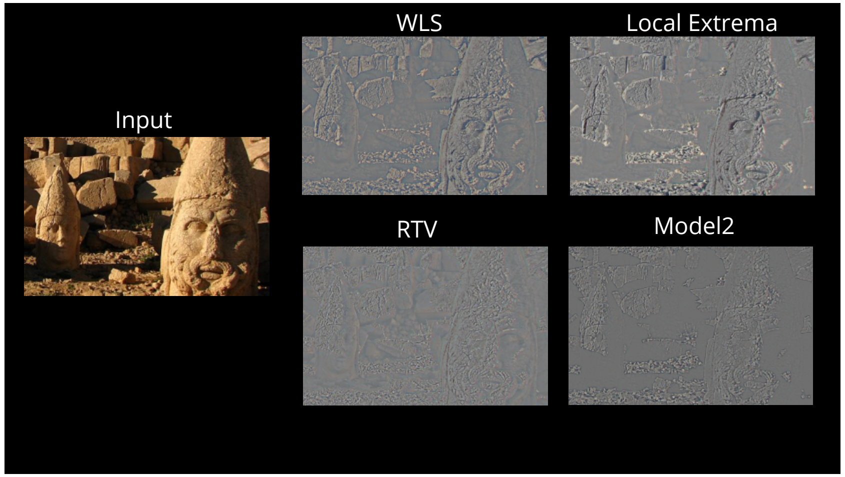

Model 1


Model 2


Model 3

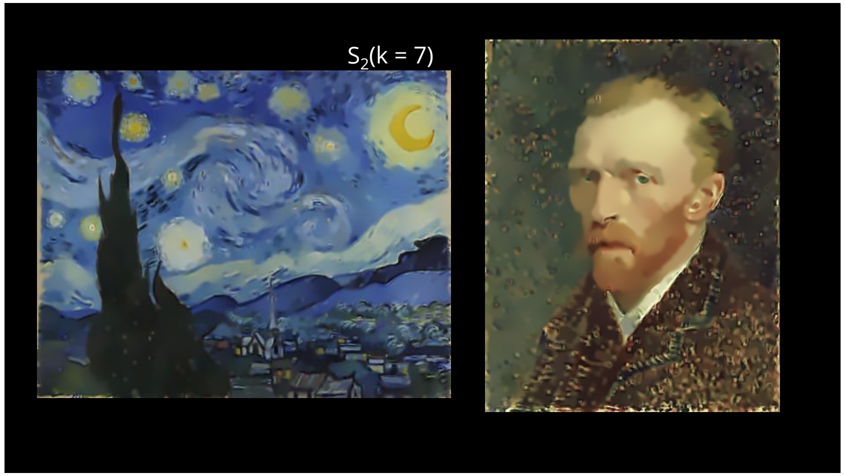

Input
Local Extrema
RTV
Model1
Model2
Model3
• Shading preserved
• Structure preserved
• No unintuitive edge

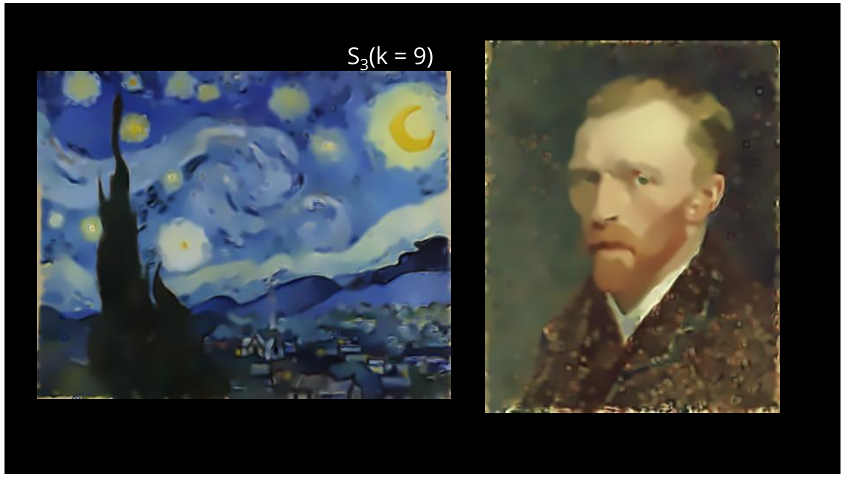

Input
WLS
Local Extrema
RTV
Model2
Preserves shade and structure

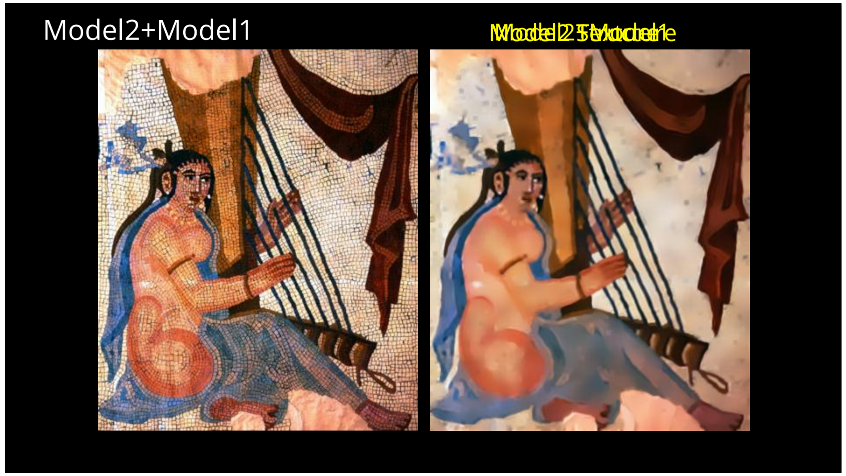

WLS
Local Extrema
Input
RTV
Model2

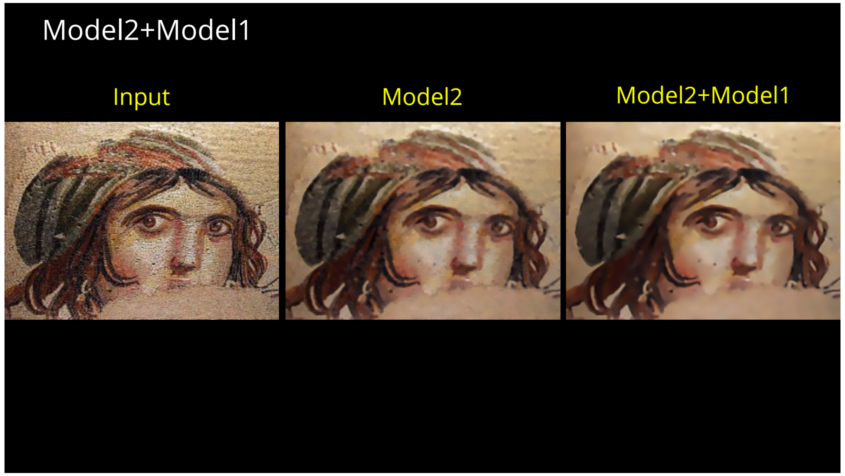

Multiscale decomposition

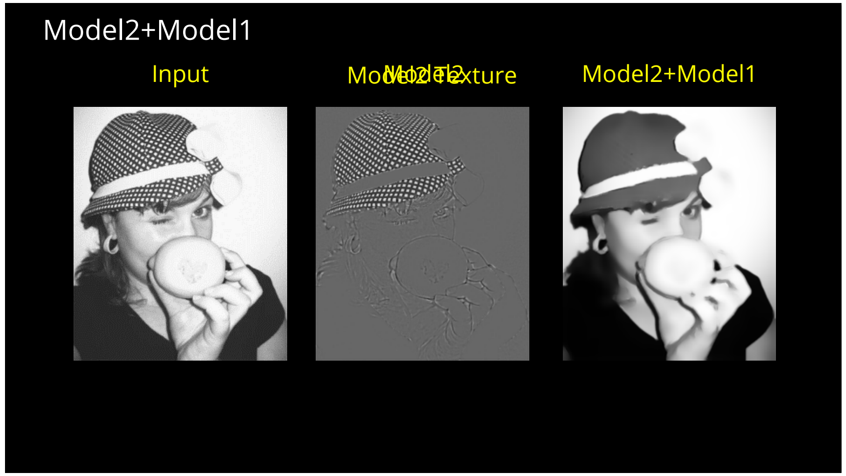

$S_1(k = 5)$


$S_2(k = 7)$


S_3(k = 9)


Model2+Model1


Model2+Model1
Input
Model2
Model2+Model1


Model2+Model1
Input
Model2+Model1

Edge Detection

Canny edges of original image | Canny edges of smoothed image

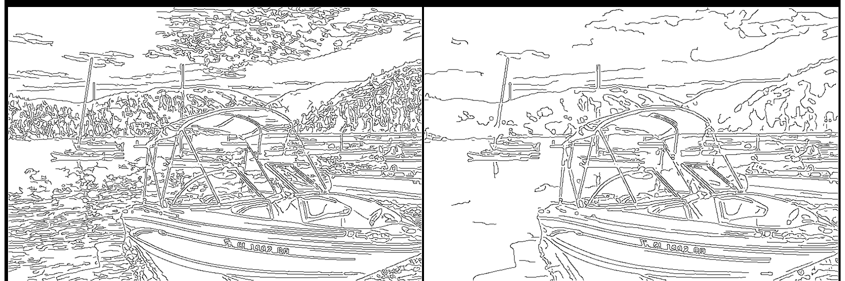


Image Abstraction

Detail Boosting

Image Composition

Inverse Halftoning

Smoothed

Shock filter
on the smooth image

Model2

Model2+Shock Filter

Kopf and Lischinski 2012

Image Retargeting

Input

Smoothed

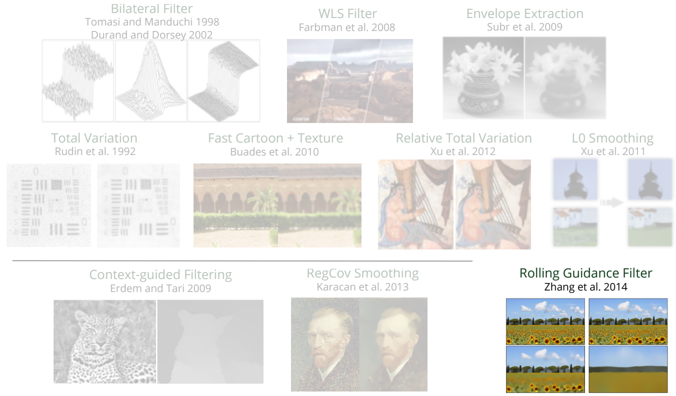
Extracted Seams
Avidan and Shamir 2007

Retargeting Results
Avidan and Shamir 2007

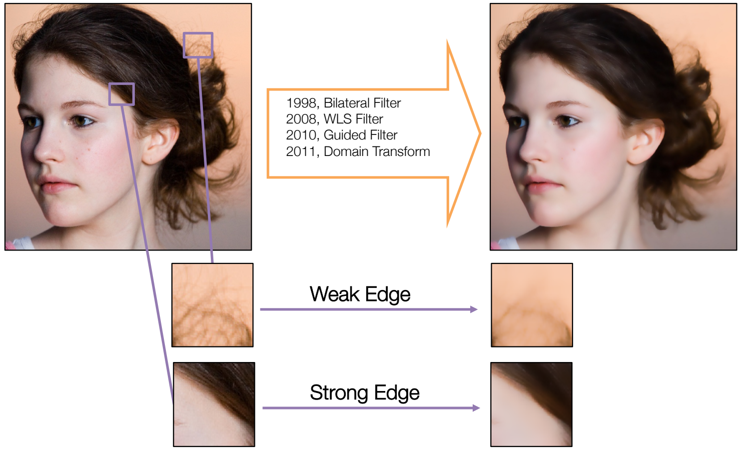
Where we are going

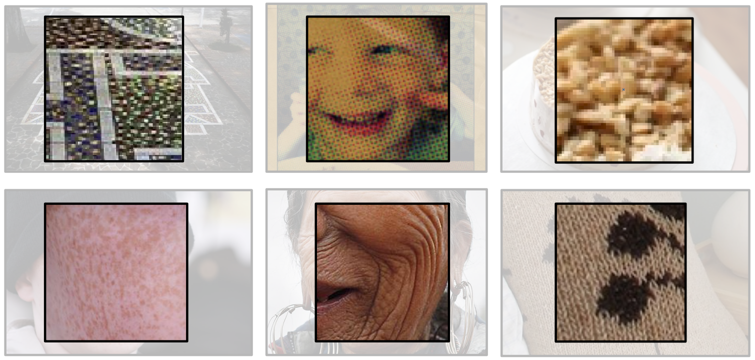
- Linear filtering
- Nonlinear filtering (unified formulations)
- Pixels to Patches (context is more important than content)
- New patch representations may reveal new smoothing behaviors
- Better the smoothing, better the applications!
- Clearly, we have a long way to go to solve the problem of image smoothing!

Some seminal works



An Important Steam: Edge Preserving



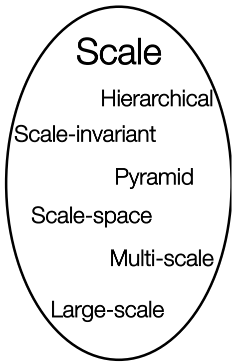


Many tiny contents are strong

What better characters them?

Scale!

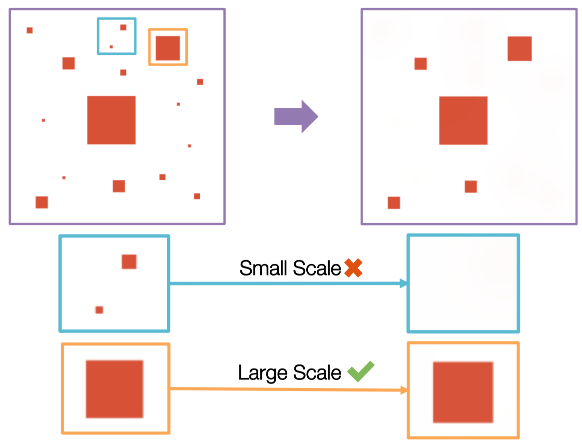
Scale in Computer Vision



- Segmentation
- Object Detection
- Saliency Detection
- Feature Extraction
- Video Analysis
- Edge Detection
- Optical Flow & Stereo
- Scene Understanding
- Action Recognition
- …

Scale + Image filter = ?

Scale-Aware Filtering



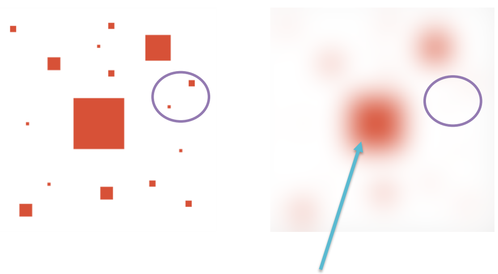
Interesting Fact

Rolling Guidance Filter (RGF) has only 1 line of code

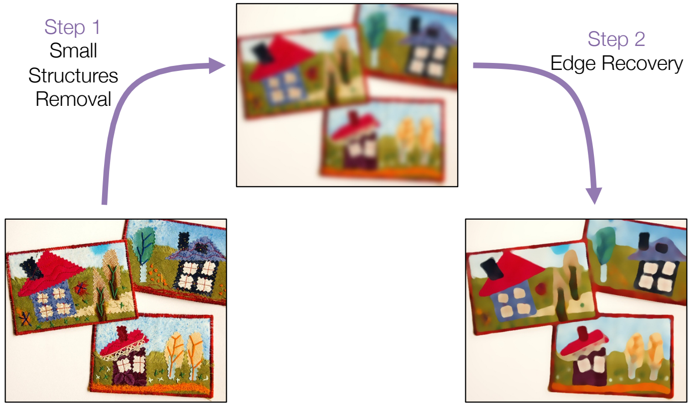
```
Mat rollingGuidanceFilter(Mat im, float scale, int iter){
    Mat res = im.mul(0);
    while(iter--) res = bilateralFilter(im,res,scale,SIGMA_R);
    return res;
}
```

Main Idea

- Scale Space Theory [Lindeberg, 1994]:
  - An object of size t, will be largely smoothed away with Gaussian filter of variance $t2$.

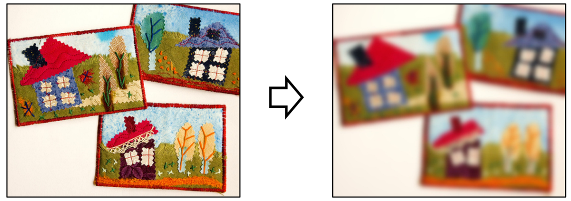


RGF: A scale-aware Filter



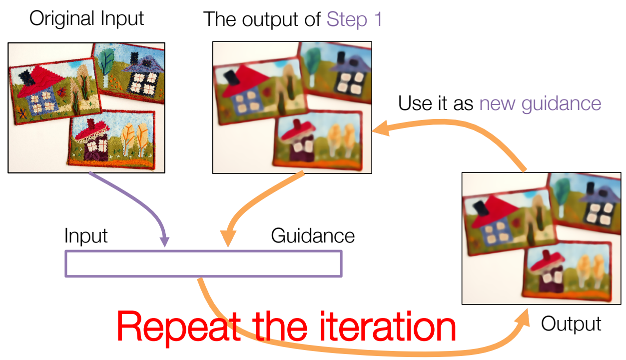
Step 1: Small Structures Removal

Gaussian Filter

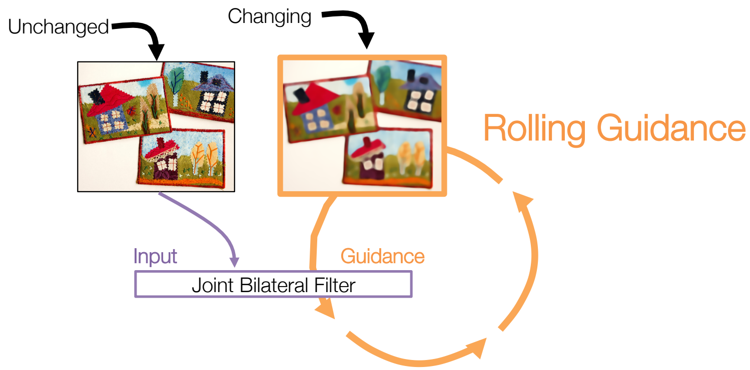


Step 2: Edge Recovery

- A rolling guidance



Rolling Guidance

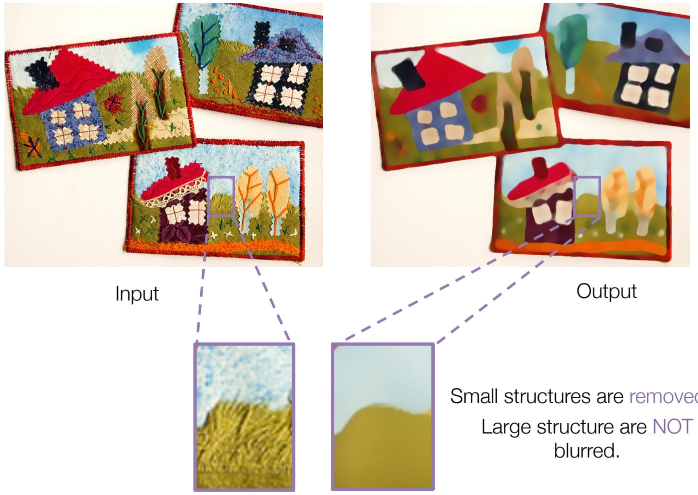


Guidance for the 1st iteration

Guidance for the 2nd iteration

Guidance for the 3rd iteration

Guidance for the 5th iteration



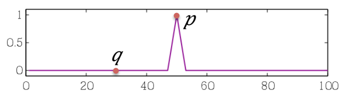
Small structures are removed.

Large structure are NOT blurred.

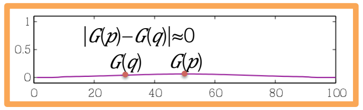
Why does rolling guidance work?

Small Structure

Input I



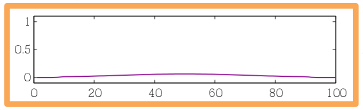
Guidance G (output of step 1)



$$J^{t+1}(p) = \frac{1}{K_p}\sum_{q\in N(p)} \exp\left(-\frac{\|p-q\|^2}{2\sigma_s^2}\right)\cdot I(q)$$

It becomes a Gaussian filter

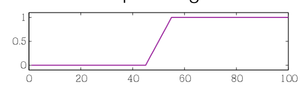
Joint Bilateral Filter



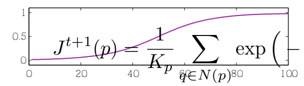
Same

Large Structure

Input Image



Result of Step 1

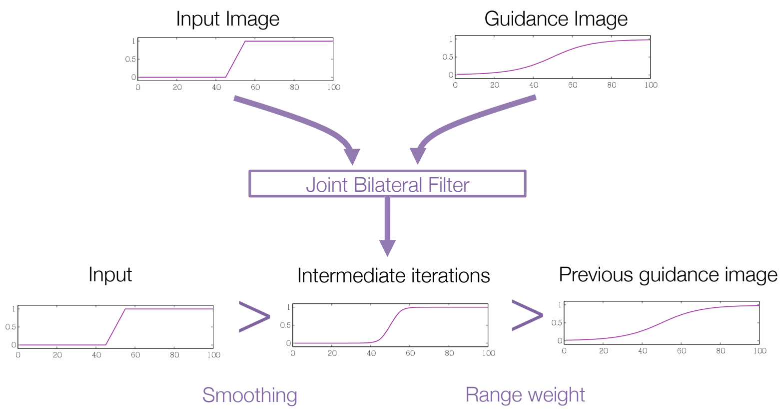


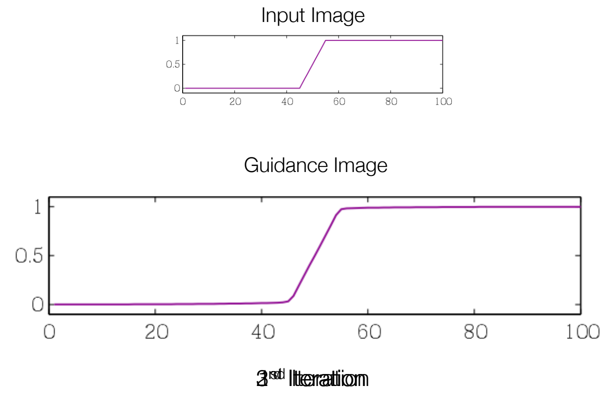
$$J^{t+1}(p) = \frac{1}{K_p}\sum_{q\in N(p)} \exp\left(-\frac{\|p-q\|^2}{2\sigma_s^2} - \frac{\|J^t(p)-J^t(q)\|^2}{2\sigma_r^2}\right) I(q)$$

≠1

Due to this range weight

It generates sharper results than Gaussian!



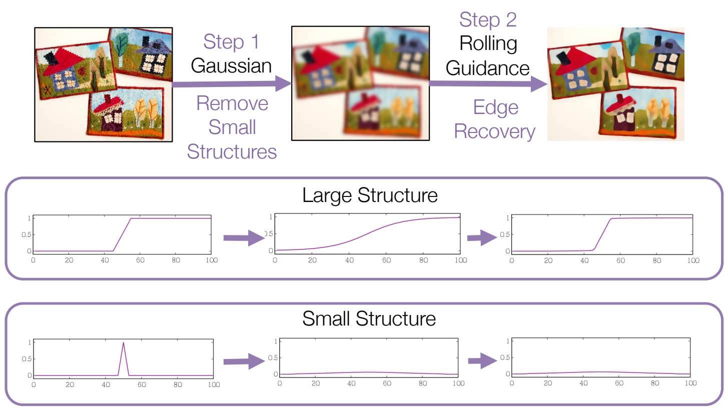


Large Structure

Take-home message

Rolling guidance recovers an edge as long as it still exists in the blurred image after Gaussian smoothing.

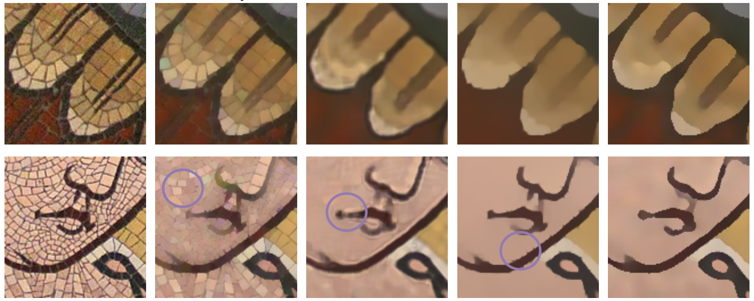
Rolling Guidance Filter



Comparison

Result Comparison

- Result comparison with related work



Input | [Subr et al.] | [Karacan et al.] | [Xu et al.] | RGF

Performance Comparison

- Performance comparison with related works

Input | RGF 2013]

For 4 Megapixel Image

2

seconds

Performance Comparison

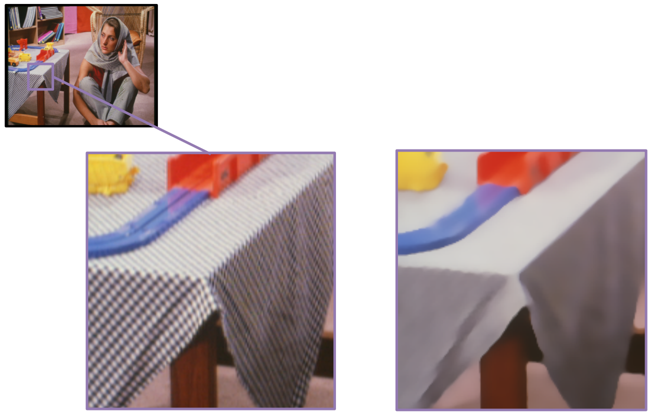
- Performance comparison

Algorithms	Time (seconds/Megapixel)
Local Extrema [Subr et al., 2009]	95
RTV [Xu et al., 2012]	14
Region Covariance [Karacan et al., 2013]	240
RGF	0.05 (Real-time)

Results & Application

Texture Removal

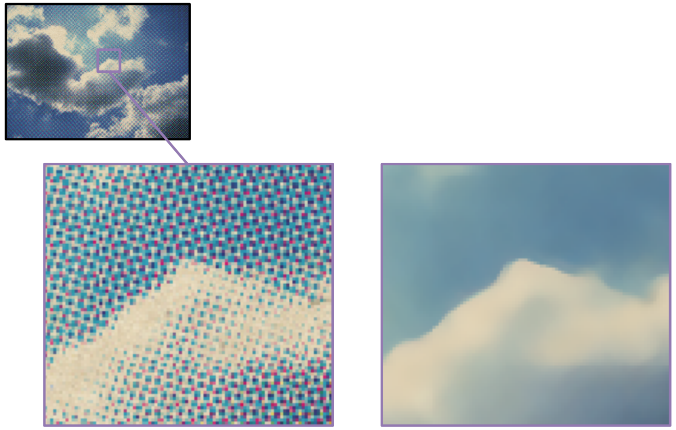
Texture Removal



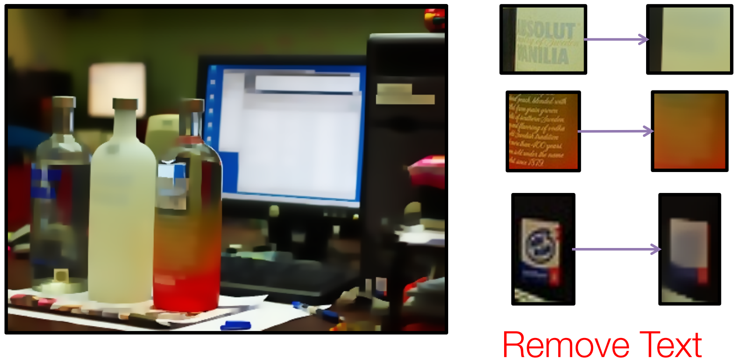
Texture Removal

Halftone Image

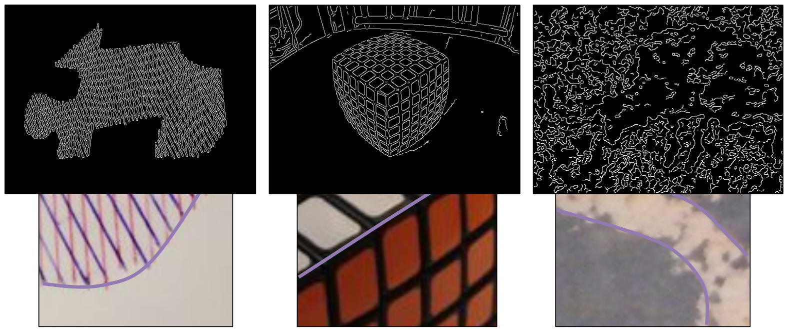
Halftone Image



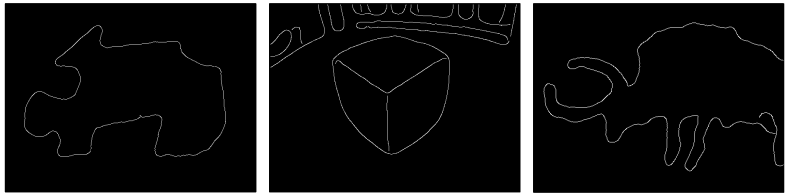
Small Text Removal



Virtual Edge Detection



Virtual Edge Detection



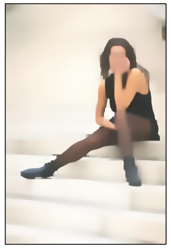
Natural Images

- Usable for
  - Segmentation
  - Saliency
  - Scene understanding
  - Background subtraction
  - Layer separation
  - Outlier removal
  - …

Input

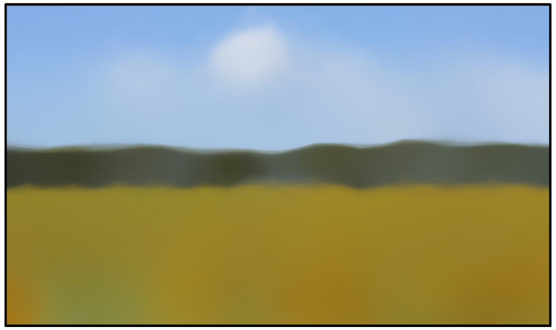
Boundary Detection

Filtered Input



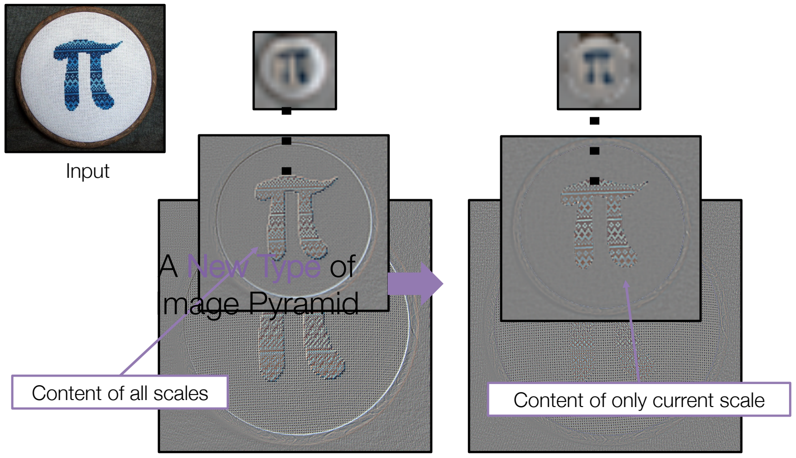
Boundary Detection

Multi-Scale Filtering



$\sigma s = 30$

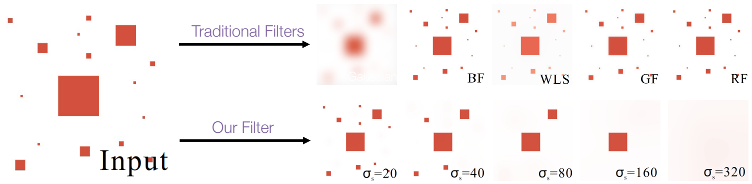
σs determine the scale.



Laplacian Pyramid
(generated with Gaussian)

Texture Pyramid
(generated with Rolling Guidance Filter)

Summary



Rolling Guidance Filter

Scale-aware

Multi-scale

Simple

Fast

Limitations

- Sharp corners could be rounded
  - It is because sharp corner presents high frequency change.
  - In other words, sharp corners are small-scale structures.