Modern Image Smoothing

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

- 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)
**Some seminal works**

- **Gaussian Filter**
- **Total Variation**
- **Bilateral Filter**
- **Envelope Extraction**
- **Context-guided Filtering**
- **Rolling Guidance Filter**
- **RegCov Smoothing**

**Context-guided filtering**

- Contextual knowledge extracted from local image regions guides the regularization process.

**Structure-Texture Decomposition**

- Decomposing an image into structure and texture components

**Context-guided Filtering**


**Structure-Texture Decomposition**

- Decomposing an image into structure and texture components
Structure-Texture Decomposition

- Decomposing an image into structure and texture components

Texture Component

Structure-Texture Decomposition

- Decomposing an image into structure and texture components

Input Image

Structure

Texture

Region Covariances as Region Descriptors

Tzimi et al., ECCV 2006

\[ F(x, y) = \begin{bmatrix} I_{(x,y)} & \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} & \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} & x & y \end{bmatrix}^T \]

\[ C_R = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T \]

Motivation

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

\[ I = S + T \]
\[ S(p) = \frac{1}{2p} \sum_{q \in N(p,r)} w_{pq} l(q) \]

- 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.

Model 1

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

\[ C = LL^T \]

**Cholesky Decomposition**

\[ S = \{s_i\} \]

**Sigma Points**

\[ s_i = \begin{cases} \alpha \sqrt{\Lambda_i} & \text{if } 1 \leq i \leq d \\ -\alpha \sqrt{\Lambda_i} & \text{if } d + 1 \leq i \leq 2d \end{cases} \]

**Final representation**

\[ \Psi(C) = (\mu, s_1, \ldots, s_d, s_{d+1}, \ldots, s_{2d})^T \]

**Resulting kernel function**

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

Model 2

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

\[ d(p, q) = \sqrt{\mu_p - \mu_q}C^{-1}(\mu_p - \mu_q)^T \]

\[ C = C_p + C_2 \]

**Resulting kernel**

\[ w_{pq} \propto \exp \left( -\frac{d(p, q)^2}{2\sigma^2} \right) \]

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

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

**Resulting kernel**

\[ w_{pq} \propto \frac{d_{KL}(p, q)}{2\sigma^2} \]
Smoothing Kernels

Figure 3.2.: Our filtering kernels consider local image geometry on calculation of filtering weights by capturing texture information.

\[ \text{Model 1} \]

Using the set \( S \) defined by Equation (21), a vectorial representation of a covariance matrix can be obtained by simply concatenating the elements of \( S \). Moreover, first-order statistics can be easily incorporated to this representation scheme by including the mean vector of the patch similarity weights. Including \((x, y)\) into the feature set is important since it allows us to encode the correlation of other features with the spatial coordinates. The feature set can be extended to include other features, like for example rotationally invariant forms of the derivatives, if desired.

In the experiments, we handle color images by computing the patch similarity weights \( w_{pq} \) using the intensity information and taking the weighted average over the corresponding RGB vectors rather than the intensity values in Equation 23. We empirically found that including RGB components to the feature set does not change the results much but increases the running times.

\[ \text{Model 2} \]

\[ \text{Model 3} \]
Model2 Structure

Model2 Texture

Input

Model2

Model3

Input
TV
Rudin et al.
1992

BLF
1998

WLS
Farbman et al.
2008

Envelope Extraction
Subr et al.
2009
Model 2

Model 3

Input

Input

Local Exrema

Local Exrema

RTV

RTV

Model 1

Model 2

Model 3

• Shading preserved
• Structure preserved
• No unintuitive edge

Preserves shade and structure
WLS  Local Extrema
RTV  Model2

Multiscale decomposition

$S_1(k = 5)$  $S_2(k = 7)$
Edge Detection

Canny edges of original image

Canny edges of smoothed image
Image Composition

Inverse Halftoning
Image Retargeting
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

- Total Variation: Rudin et al. 1992
- Fast Cartoon + Texture: Buades et al. 2010
- Relative Total Variation: Xu et al. 2012
- L0 Smoothing: Xu et al. 2011
- Context-guided Filtering: Erdem and Tari 2009
- RegCog Smoothing: Xu et al. 2012
An Important Steam: Edge Preserving

1998, Bilateral Filter
2008, WLS Filter
2010, Guided Filter
2011, Domain Transform

Weak Edge

Strong Edge

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

Hierarchical
Scale-invariant
Pyramid
Scale-space
Multi-scale
Large-scale
Scale + Image filter = ?

Interesting Fact

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

Main Idea

• Scale Space Theory [Lindeberg, 1994]:
  – An object of size $\sigma$, will be largely smoothed away with Gaussian filter of variance $\sigma^2$. 

```matlab
function res = rollingGuidanceFilter(im, scale, iter)
    res = im; % Initialize
    for i = 1:iter
        res = bilateralFilter(im, res, scale, SIGMA_R);
    end
    return res;
end
```
RGF: A scale-aware Filter

Step 1: Small Structures Removal
- Gaussian Filter

Step 2: Edge Recovery
- Joint Bilateral Filter

Step 2: Edge Recovery
- A rolling guidance
  - Original Input
  - The output of Step 1
  - Use it as new guidance

Rolling Guidance
- Input
- Guidance
- Unchanged
- Changing
- Joint Bilateral Filter
- Rolling Guidance

Repeat the iteration
Small Structure

Input / Guidance $g'$ (output of step 1)

Joint Bilateral Filter

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

It becomes a Gaussian filter

Large Structure

Input Image

Result of Step 1

Due to this range weight, it generates sharper results than Gaussian!

Why does rolling guidance work?
Rolling guidance recovers an edge as long as it still exists in the blurred image after Gaussian smoothing.
Comparison

Performance Comparison

• Performance comparison with related works

For 4 Megapixel Image

Input  RGF [2013]

2 seconds

Result Comparison

• Result comparison with related work

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

Performance Comparison

• Performance comparison

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Time (seconds/Megapixel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Extrema [Subr et al., 2009]</td>
<td>95</td>
</tr>
<tr>
<td>RTV [Xu et al., 2012]</td>
<td>14</td>
</tr>
<tr>
<td>Region Covariance [Karacan et al., 2013]</td>
<td>240</td>
</tr>
<tr>
<td>RGF</td>
<td>0.05 (Real-time)</td>
</tr>
</tbody>
</table>
Results & Application

Texture Removal

Texture Removal

Texture Removal
Virtual Edge Detection

Natural Images

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

Boundary Detection

Multi-Scale Filtering

\[ a_s = 30 \]

\( a_s \) determine the scale.
Limitations

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