Real-time Image Enhancement Using Edge-Optimized À-trous Wavelets
Overview

• Motivation
• À-trous Wavelets
• Edge-avoiding Wavelets
• Edge-optimized Wavelets
• Contrast Enhancement
• Including depth cues from stereo cameras
Edge-Avoiding À-Trous Wavelet Transform for fast Global Illumination Filtering

[Dammertz, Sewtz, Hanika, Lensch – HPG 2010]
Edge-Avoiding À-Trous Wavelet Transform for fast Global Illumination Filtering

[Dammertz, Sewtz, Hanika, Lensch – HPG 2010]
HPG 2010

paper1029

Edge-Avoiding A-Trous Wavelet Transform for fast Global Illumination Filtering

supplemental Material
Input to our System

Filtering is based on
• normal buffer
• position buffer
• original, noisy Monte Carlo image
  • may contain high frequency information for the illumination which
    is not represented in the other buffers
Pipeline

multiple iterations of filtering based on edge-stopping function

path traced image

position

normal

smoothed illumination

detail texture
Decimating Wavelet Transform

- Apply the same filter multiple times
  - reduce the image resolution at each iteration

- Benefits: multi-scale representation, very fast!
- Drawback: filtered information at few locations only
Decimating Wavelet Transform

- Apply the same filter multiple times
  - reduce the image resolution at each iteration

- Benefits: multi-scale representation, very fast!
- Drawback: filtered information at few locations only
  - Where is the edge?
Decimating Wavelet Transform

• Apply the same filter multiple times
  • reduce the image resolution at each iteration

• Benefits: multi-scale representation, very fast!
• Drawback: filtered information at few locations only
  • Where is the edge?
Decimating Wavelet Transform

- Apply the same filter multiple times
  - reduce the image resolution at each iteration

- Benefits: multi-scale representation, very fast!
- Drawback: filtered information at few locations only
  - Where is the edge?
Undecimated Transform

- At each iteration convolve with a Gaussian
  - double the filter size

- Benefit: filtered information at every pixel
- Drawback: huge effort due to growing filter size (4 x samples)
À-Trous Wavelet Transform

- “With holes”
- At each iteration convolve with a Gaussian
  - double the filter size
  - introduce more and more holes

Benefits:
- constant effort per iteration (in contrast to undecimated wavelets)
- filtered information at each pixel (in contrast to decimated wavelets)
À-Trous Wavelet Transform

- “With holes”
- At each iteration convolve with a Gaussian
  - double the filter size
  - introduce more and more holes

Benefits:
- constant effort per iteration (in contrast to undecimated wavelets)
- filtered information at each pixel (in contrast to decimated wavelets)
À-Trous Wavelet Transform

- "With holes"
- At each iteration convolve with a Gaussian
  - double the filter size
  - introduce more and more holes

- Benefits:
  - constant effort per iteration (in contrast to undecimated wavelets)
  - filtered information at each pixel (in contrast to decimated wavelets)
À-Trou Wavelet Transform

1. At level \( i = 0 \), start with input signal \( c_0(p) \).

2. \( c_{i+1}(p) = c_i(p) * h_i \), where \(*\) is the discrete convolution.
   The distance between the entries in the filter \( h_i \) is \( 2^i \).

3. \( d_i(p) = c_i(p) - c_{i+1}(p) \),
   where \( d_i \) are the detail or wavelet coefficients of level \( i \).

4. Repeat 2 and 3 until \( i = N \) (number of levels to compute).

5. \( \{d_0, d_1, \ldots, d_{N-1}, c_N\} \) is the wavelet transform of \( c \).

The reconstruction is given by \( c = c_N + \sum_{i=0}^{N-1} d_i \).
À-Trous Wavelet Filtering

1. At level $i=0$, start with input signal $c_0(p)$.

2. $c_{i+1}(p) = c_i(p) \ast h_i$, where $\ast$ is the discrete convolution.

   The distance between the entries in the filter $h_i$ is $2^i$.

3. $d_i(p) = c_i(p) - c_{i+1}(p)$,

   where $d_i$ are the detail or wavelet coefficients of level $i$.

4. Repeat 2 and 3 until $i = N$ (number of levels to compute)

5. $\{d_0, d_1, \ldots, d_{N-1}, c_N\}$ is the wavelet transform of $c$.

The filtered result is given by $c = c_N + \sum_{i=0}^{N-1} \alpha_i d_i$.
Edge-Optimized À-trous Wavelets | Hendrik Lensch

[Fattal et al. 2007, Dammertz et al. 2010]

**Edge-Stopping Function**

Compute weighted convolution (compare to bilateral filter)

\[
    c_{i+1}(p) = \frac{\sum_{q \in Q} h_i(q) \cdot w(p, q) \cdot c_i(p)}{\sum_{q \in Q} h_i(q) \cdot w(p, q)}
\]

With weights

\[
    w(p, q) = w_X(p, q) \quad \text{with} \quad w_X(p, q) = e^{-\frac{||I_p - I_q||}{\sigma_X^2}}
\]

\(\sigma_X\) can be controlled for each level independently

--

noisy

orig. à-trous

edge-preserving

\[w_{rt} \cdot w_n\]
Input
Level 0
Level 1
Level 2
Comparison to Other Wavelet Bases

edge-avoiding à-trous

CDF(2,2)
Comparison to Other Wavelet Bases

edge-avoiding à-trous

difference

edge-avoiding CDF(2,2)

[Fattal 2009]
Comparison to Other Wavelet Bases

- Edge-avoiding à-trous
  5x5 – 5 iterations, 5.6 ms

- Bilateral filter
  90x90 – one iteration, 2 min
Contrast Enhancement

input

edge-avoiding decimated wavelets  edge-optimized à-trous
Optimizing the Edge Weights

too weak halo  

too strong gradient reversal

optimized

[Hanika, Dammertz, Lensch – Pacific Graphics 2011]
Edge Optimization

- Each edge can have different width / sharpness / contrast
- For each level $i$ and each pixel $j$ optimize $\sigma_X$ such that the detail $d_{i,j}$ gets as small as possible while keeping a smooth base layer $c_{i,j}$:

$$e_j = d_{i,j}^2 + \lambda \cdot \left\| \nabla c_{i,j} \right\|$$

- Simply try four different $\sigma_X$ for each pixel
Edge Optimization

standard à-trous

edge-optimized à-trous

level 3  level 4  level 3  level 4
Edge Optimization – Detail Layer

standard à-trous

edge-optimized à-trous

level 3

level 4

level 3

level 4

edge-optimization keeps more of the true signal in the base layer
Denoising by Shrinkage

Estimate standard deviation of the noise in the image:

\[ \sigma_n = \frac{\text{median}(|d_0|)}{0.6745} \]

Compute optimal shrinkage threshold minimizing risk of information loss

\[ T = \frac{\sigma_{n,i}^2}{\sqrt{\max\{0, \sigma_{y,i}^2 - \sigma_{n,i}^2\}}} \]

with \( \sigma_{y,i}^2 = \frac{1}{N} \sum_p d_i(p)^2 \) and \( \sigma_{n,i} = \sigma_n \cdot 2^{-i} \)

Apply shrinkage and contrast boost (if wanted):

\[ d'_i = \max\{0, |d_i| - T\} \cdot \text{sign}(d_i) \]

and \( c_{i-1} = c_i + \beta \cdot d'_i \)
Denoising

input: 10% noise PSNR 26.2

à-trou PSNR 34.8
edge-optimized à-trou PSNR 35.9
Denoising

\[ \text{à-trous PSNR 34.8} \quad \text{edge-optimized à-trous: PSNR 35.9} \]
Young and Old
Edge-Optimized À-Trous Wavelets – Contrast Enhancement
Edge-Optimized À-Trous Wavelets – Smoothing
Comparison – Contrast Enhancement 2.5x

[Kass&Solomon – SIGGRAPH 2010]
1 second per megapixel

Edge-optimized à-trous
0.01 seconds per megapixel
Comparison – Contrast Enhancement 3.5x

[Kass&Solomon – SIGGRAPH 2010]  
1 second per megapixel

Edge-optimized à-trous  
0.01 seconds per megapixel
Benefits of Edge-Optimized À-Trous Wavelets

- Fast, fast, fast
- Simple
- Arbitrary filter sizes
- Control over individual frequency bands
  - avoid ringing in contrast enhancement
  - locally adapts to the strength of each edge
  - optimized denoising
MONOCULAR DEPTH CUES FROM STEREO CAMERAS

[Roessing, Hanika, Lensch – Eurographics 2012]
Miniaturization using virtual tilt-shift
Original (left eye)
Guiding the User’s Focus
Monocular Depth Cues

… in the absence of stereo displays

Fix per scene:
- object size
- size relations
- occlusion

Can be modified:
- sharpness
- contrast
- color saturation
- (occlusion)
Edge-optimized Wavelet – Pipeline

Requires multi-scale decomposition
Cleaning up the Depth Map

- Interpolation of missing depth values (stereo shadows)
- Multi-scale cross-bilateral filtering for exact silhouettes
Interpolation of Missing Values
Cross-Bilateral Wavelet Filtering
Depth-dependent Local Contrast Enhancement
Unsharp Masking the Depth Buffer + Local Contrast Enhancement
Edge-Optimized Traces Wavelets | Hendrik Lensch

Depth-based Saturation
Depth-dependent Color Saturation
User Study

Can we manipulate an image in such a way that search tasks are done faster without destroying the image content?

- no arrows, boxes
- object at specific, (magically determined) depth
- measure reaction time
User Study – Experiment 1

- Images with and without ball in any quadrant at any distance
- Images enhanced in various ways
- No “learning” possible
Experiment 1 - Results

- Search time significantly decreased
- Combination of multiple depth cues necessary
Experiment 2

- What happens if the wrong part is in focus?
- How much is the remainder destroyed?
Experiment 2 - Results

- DOF rendering with the EAW framework is less destructive than with Gaussian filtering.
Depth of Field Rendering
Edge-Optimized Wavelets

- Ultra-fast contrast enhancement or smoothing (without artifacts)
- Supports filtering over large distances
- Can be used to improve depth maps
- Direct feedback to the cameraman
- Enhance depth cues due to stereo
- Q: Use as a compact basis for computational learning
Thank you!

Students
Holger Dammertz, Daniel Setws, Johannes Hanika, Christoph Roessing

Cooperation
Daimler AG

