A New Multi-Lateral Filter for Real-Time Depth Enhancement

Frederic Garcia¹, ² Djamila Aouada² Bruno Mirbach¹ Thomas Solignac¹ Björn Ottersten²

¹ Advanced Engineering Department, IEE S.A.
{frederic.garcia, bruno.mirbach, thomas.solignac}@iee.lu

² Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
{djamila.aouada, bjorn.ottersten}@uni.lu

August 31, 2011

Frederic Garcia was supported by the AFR Grant Scheme (Aides à la Formation-Recherche), managed by the National Research Fund Luxembourg (FNR). AFR Grant: TR-PHD BFR08-120.
Goal

Increase: spatial resolution & depth accuracy

ToF camera
3D MLI Sensor™

2D camera
Flea® 2

Hybrid ToF
multi-camera Rig

ToF Depth Map
• Low Resolution (56x61) pixels
• Noisy

Colour Image
• High Resolution (640x480) pixels
• Almost no noise

Enhanced Depth Map
• High Resolution (640x480) pixels
• Significantly reduced noise
Outline

- Background: Low-level data fusion
  - Bilateral filter
  - Joint Bilateral Upsampling (JBU)
- Problem: Misalignment between depth and 2-D data
- Solution:
  - Previous work ➔ PWAS filter
  - New filter ➔ UML filter
- Real-Time
- Experimental results
- Conclusions
Background: Low-level data fusion

- Bilateral filter: [Tomasi98]

\[ J_1(p) = \frac{\sum_{q \in N(p)} f_S(p, q) \cdot f_R(R(p), R(q)) \cdot R(q)}{\sum_{q \in N(p)} f_S(p, q) \cdot f_R(R(p), R(q))} \]

Pros & Cons:
- ✔ Reduced noise
- ✔ Preserved depth edges
- ✗ Not enhanced depth edges

\( f_{S,R}(\cdot) \): Gaussian functions
Background: Low-level data fusion

- **Bilateral filter**: [Tomasi98]

\[
J_1(p) = \frac{\sum_{q \in N(p)} f_S(p, q) \cdot f_R(R(p), R(q)) \cdot R(q)}{\sum_{q \in N(p)} f_S(p, q) \cdot f_R(R(p), R(q))}
\]

\(f_{S,R}(\cdot):\) Gaussian functions
Background: Low-level data fusion

- **Joint Bilateral Upsampling (JBU):** [Kopf07]

\[
J_2(p) = \frac{\sum_{q \in N(p)} f_S(p, q) \cdot f_1(I(p), I(q)) \cdot R(q)}{\sum_{q \in N(p)} f_S(p, q) \cdot f_1(I(p), I(q))}
\]

**Pros & Cons:**

- Reduced noise
- Preserved depth edges
- Enhanced depth edges
- Blurred depth edges
- Copied texture
Problem: Misalignment depth and 2-D edges

- Depth measurement is inaccurate on edge pixels:
Previous work in low-level data fusion

- Pixel Weighted Average Strategy for Depth Sensor Data Fusion (PWAS): [Garcia10]

\[
J_3(p) = \frac{\sum_{q \in N(p)} f_S(p,q) \cdot f_1(I(p), I(q)) \cdot Q(q) \cdot R(q)}{\sum_{q \in N(p)} f_S(p,q) \cdot f_1(I(p), I(q)) \cdot Q(q)}
\]

\[Q = f_Q(-|\nabla R|)\]

\(f_{S,I,Q}(\cdot)\): Gaussian functions

![J_3, PWAS](image)
Previous work in low-level data fusion

- Pixel Weighted Average Strategy for Depth Sensor Data Fusion (PWAS): [Garcia10]

\[
J_3(p) = \frac{\sum_{q \in N(p)} f_S(p,q) \cdot f_1(I(p),I(q)) \cdot Q(q) \cdot R(q)}{\sum_{q \in N(p)} f_S(p,q) \cdot f_1(I(p),I(q)) \cdot Q(q)}, \quad Q = f_Q(-|\nabla R|)
\]

\(f_{S,I,Q}(\cdot):\) Gaussian functions

**Pros & Cons:**

- Reduced noise
- Preserved depth edges
- Enhanced depth edges
- Eliminated edge blurring
- Reduced texture copying

J₃, PWAS
UML filter for depth sensor data fusion

- Unified Multi-Lateral (UML) filter for depth sensor data fusion
  - Considers 2-D and depth data as guidance information:

\[
J_5(p) = (1 - \beta(p)) \cdot J_3(p) + \beta(p) \cdot J_4(p)
\]

[Chan08, Kim10] different JBU filter extensions considering 2-D and depth data
UML filter for depth sensor data fusion

- Unified Multi-Lateral (UML) filter for depth sensor data fusion
  - Considers 2-D and depth data as guidance information:

\[
J_5(p) = (1 - \beta(p)) \cdot J_3(p) + \beta(p) \cdot J_4(p), \quad \beta = Q
\]
UML filter for depth sensor data fusion

- Unified Multi-Lateral (UML) filter for depth sensor data fusion
  - Considers 2-D and depth data as guidance information:

\[
J_5(p) = (1 - \beta(p)) \cdot J_3(p) + \beta(p) \cdot J_4(p), \quad \beta = Q
\]

**Pros & Cons:**
- Reduced noise
- Preserved depth edges
- Enhanced depth edges
- Eliminated edge blurring
- Eliminated texture copying

[Chan08, Kim10] different JBU filter extensions considering 2-D and depth data
Why “Unified”?  

- UML filter allows different filter configurations to behave as:

\[ J_5(p) = (1 - \beta(p)) \cdot J_3(p) + \beta(p) \cdot J_4(p) \]

- Bilateral filter, \( \beta(\cdot) = 1 \) & \( \sigma_Q = \infty \)

- JBU filter, \( \beta(\cdot) = 0 \) & \( \sigma_Q = \infty \)

- PWAS filter, \( \beta(\cdot) = 0 \)
Real-Time UML filter (I)

- Range data quantization [Yang09]
  - Calculate the filter response for each quantized level, $I_k$ for $I(p)$:

  $$J_3(p, I_k) = \frac{\sum_{q \in N(p)} f_S(p, q) \cdot f_I(I_k, I(q)) \cdot Q(q) \cdot R(q)}{\sum_{q \in N(p)} f_S(p, q) \cdot f_I(I_k, I(q)) \cdot Q(q)}$$

  and $R_l$ for $R(p)$:

  $$J_4(p, R_l) = \frac{\sum_{q \in N(p)} f_S(p, q) \cdot f_R(R_l, R(q)) \cdot Q(q) \cdot R(q)}{\sum_{q \in N(p)} f_S(p, q) \cdot f_R(R_l, R(q)) \cdot Q(q)}$$

  $\Rightarrow J_3(p, I_k)$ and $J_4(p, R_l)$ can be expressed as convolutions

$f_{S,I,R}(\cdot)$: Gaussian functions

### A New Multi-lateral Filter for Real-Time Depth Enhancement
Real-Time UML filter (II)

- Range data quantization [Yang09]
  - Calculate the filter response for each quantized level \((I_k, R_l)\)
  - Filtered output \(J_5(p)\) results from:
    - 2-point interpolation between \(J_3(p, I_k)\) and \(J_3(p, I_{k+1})\)
    - 2-point interpolation between \(J_4(p, R_l)\) and \(J_4(p, R_{l+1})\)
- Data downsampling [Paris09]
  - Filtering can be performed on downsampled images
  - Filtered output \(J_5(p)\) results from a 4-point interpolation
- We combine both approaches
  - Calculate the filter response for each quantized level \(I_k\) and \(R_l\) on downsampled images
  - Perform a 8-point interpolation
Experimental results (I)

- Runtime analysis (Acer Aspire 4810T, Intel Core™2 Solo processor SU3500):

<table>
<thead>
<tr>
<th></th>
<th>1x</th>
<th>2x</th>
<th>4x</th>
<th>8x</th>
<th>16x</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBU</td>
<td>1.88s</td>
<td>0.49s</td>
<td>0.13s</td>
<td>0.06s</td>
<td>0.05s</td>
</tr>
<tr>
<td>PWAS</td>
<td>1.89s</td>
<td>0.50s</td>
<td>0.13s</td>
<td>0.06s</td>
<td>0.05s</td>
</tr>
<tr>
<td>UML</td>
<td>13.59s</td>
<td>3.17s</td>
<td>0.65s</td>
<td>0.18s</td>
<td>0.08s</td>
</tr>
</tbody>
</table>

- SSIM measure depending on the input data sampling (Middelbury Stereo Dataset¹):

<table>
<thead>
<tr>
<th></th>
<th>2x</th>
<th>4x</th>
<th>8x</th>
<th>16x</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBU</td>
<td>95.78</td>
<td>95.46</td>
<td>94.89</td>
<td>92.25</td>
</tr>
<tr>
<td>PWAS</td>
<td>99.71</td>
<td>99.51</td>
<td>98.80</td>
<td>95.11</td>
</tr>
<tr>
<td>UML</td>
<td>99.85</td>
<td>99.65</td>
<td>98.86</td>
<td>95.17</td>
</tr>
</tbody>
</table>

¹Middlebury Stereo Dataset, http://vision.middlebury.edu/stereo
Experimental results (II)

- Experiments considering our own recorded sequences:

(a) JBU  
(b) PWAS  
(c) UML  

(d) Enhanced depth map

(d) JBU  
(e) PWAS  
(f) UML  

(h) Enhanced depth map
Experimental results (III)

ToF mapped data

Fused data

2D mapped data
Conclusions

- The proposed multi-lateral filter enhances low resolution depth maps with a global noise level reduction.

- The credibility map allows an accurate adjustment of depth edges.

- The filter prevents texture copying and limits edge blurring to the credibility map boundaries.

- Depth accuracy has been increased within geometrically smoothed regions.

- We achieve real-time performance from the proposed filter implementation.
Thank you!

Frederic Garcia¹,² Djamila Aouada² Bruno Mirbach¹ Thomas Solognac¹ Björn Ottersten²

¹ Advanced Engineering Department, IEE S.A.
{frederic.garcia, bruno.mirbach, thomas.solognac}@iee.lu

² Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
{djamila.aouada, bjorn.ottersten}@uni.lu

August 31, 2011


Experimental results

- Experiments considering the Middelbury Stereo Dataset

Middelbury Stereo Dataset: http://vision.middlebury.edu/stereo
Experimental results

- Comparison between JBU, PWAS & UML using different $\sigma_S$ values: