ACCOUNTING FOR UNCERTAINTY IN MEDICAL DATA: A CUDA IMPLEMENTATION OF NORMALIZED CONVOLUTION

Stefan Lindholm and Joel Kronander
Scientific Visualization Group
Linköping University
Edge Detection
Bilateral Filtering
Transfer Functions
Gradient Estimation

FILTERING MEDICAL IMAGES
UNCERTAINTY
**Known Uncertainty**

- 180 mAs CT
- 12 mAs CT

Notation: $s(k)$, $r(k)$
Filtering a known signal is straightforward
..but..
How do we treat uncertainty?

Local filter kernels, adaptive to the local signal uncertainty!
THEORY
**Convolution**

Interpret convolution as a projection

\[ h = s * f \]

Polynomial expansion

\[ h(k) = \sum_{l} s(k + l) f(-l) \]

\[ h_m(k) = \langle s(k + l) f_m(-l) \rangle \]

\[ h_m(k) = \sum_{l} s(k + l) G_l f_m(-l) \]
**Normalized Convolution**

\[ h = s \ast f \]

**Polynomial expansion**

\[ h(k) = \sum_{l} s(k + l) \overline{f(-l)} \]

\[ h_m(k) = \langle s(k + l) | \overline{f_m(-l)} \rangle \]

\[ h_m(k) = \sum_{l} s(k + l) G_{kl} f_m(-l) \]

**Applicability:** \( a(l) \)

**Signal certainty:** \( r(k) \)

\[ h_m(k) = \sum_{l} s(k + l) r(k + l) a(l) f_m(l) \]
Normalized Convolution

Polynomial Expansion for Orientation and Motion Estimation. [Far02] Farnebäck

\[ c = \begin{pmatrix} \langle a \ast b_1 \ast b_1 | r \rangle & \ldots & \langle a \ast b_1 \ast b_M | r \rangle \\ \vdots & \ddots & \vdots \\ \langle a \ast b_M \ast b_1 | r \rangle & \ldots & \langle a \ast b_M \ast b_M | r \rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle a \ast b_1 | r \ast s \rangle \\ \vdots \\ \langle a \ast b_M | r \ast s \rangle \end{pmatrix} \]

One matrix inversion per pixel in the image!

Adaptive metric! 😊
Spatially varying! 😞
Pre-computation! 😊
G is only M x M! 😊
CUDA
CUDA

- Standard optimizations
  - 512 cores
  - Coalesced memory reads
  - Latency hiding
  - Loop unrolling

- Optimizations for Normalized Convolution
  - Templates for recursive matrix inversion
  - Pre-processor defines

![Graph showing performance comparison between global and local memory.]
CUDA

- Standard optimizations
  - 512 cores
  - Coalesced memory reads
  - Latency hiding
  - Loop unrolling
- Optimizations for Normalized Convolution
  - Templates for recursive matrix inversion
  - Pre-processor defines

\[ A^{-1} = \frac{1}{\det(A)} \times \text{Adj}(A) \]

```cpp
template<int M> __device__ float Determinant()
{ ... Determinant<M-1>() ... }
template<> __device__ float Determinant<3>();
template<> __device__ float Determinant<2>();
template<> __device__ float Determinant<1>();
```

(512^2 image, 3 bases)
CUDA

CONCLUSIONS

G is only $M \times M$ 😊

...but even $M \times M$ is costly 😞

...and we need precision 😞

We can achieve real-time performance! 😊
RESULTS
RESULTS
RESULTS
Thank You!
\[ h_m(k) = \sum_l s(k+l)f_m(-l) \]
\[ h_m(k) = \sum_l s(k+l)G_l f_m(-l) \]
\[ h_m(k) = \sum_l s(k+l)r(k+l)a(l)f_m(l) \]
\[ h(k) = \langle s(k + l)|f^*(-l) \rangle \]
\[ c = G^{-1} \]
\[ c = (B^* G_0 B)^{-1} \]

\[
[111] \ast [111] = \frac{(1+1+1)}{3} = 1
\]

\[
[011]_{[111]} \ast [111] = \frac{(0+1+1)}{3} = 0.67
\]

\[
(1+1)/2 = 1
\]
**Convolution**

**Polynomial expansion**

\[ h(k) = \sum_{l} s(k + l) f(-l) \]

\[ h_m(k) = \{ s(k + l) \overline{f_m(-l)} \} \]

\[ h_m(k) = \sum_{l} s(k + l) G_{kl} \overline{f_m(-l)} \]

\[ h_m(k) = \sum_{l} s(k + l) r(k + l) a(l) f_m(l) \]
Normalized Convolution

Polynomial expansion

\[ f_1(l) \]
\[ \vdots \]
\[ f_M(l) \]

Applicability: \( a(l) \)
Signal certainty: \( r(k) \)

\[ h_m(k) = \sum_l s(k + l) r(k + l) a(l) f_m(l) \]

\[ c = G^{-1} h_m \]

Weighted least squares

Uncertainty!

Different metric!

Spatially varying!
Normalized Convolution

Polynomial Expansion for Orientation and Motion Estimation. [Far02] Farnebäck

\[
c = \begin{pmatrix}
\langle a \ast b_1 \ast \bar{b}_1 | r \rangle & \cdots & \langle a \ast b_1 \ast \bar{b}_M | r \rangle \\
\vdots & \ddots & \vdots \\
\langle a \ast b_M \ast \bar{b}_1 | r \rangle & \cdots & \langle a \ast b_M \ast \bar{b}_M | r \rangle
\end{pmatrix}^{-1}
\begin{pmatrix}
\langle a \ast b_1 | r \ast s \rangle \\
\vdots \\
\langle a \ast b_M | r \ast s \rangle
\end{pmatrix}
\]

\[
c = G^{-1} h_m
\]

Different metric! 🔴
Spatially varying! 🔴
G is only $M \times M$! 😊
Pre-computation! 😊