Medical Image Processing for Interventional Applications

Image Enhancement

Online Course – Unit 15

Andreas Maier, Jakob Wasza, Frank Schebesch
Pattern Recognition Lab (CS 5)

virtuelle hochschule bayern
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Topics

Image Enhancement

Motivation

Normalized Convolution

Bilateral Filtering

Summary

Take Home Messages

Further Readings

Range Imaging (RI)

Metric surface measurement:

- Marker-less, non-intrusive
- Real-time capable

Specification	PMD CamCube 3.0	Microsoft Kinect
Principle	ToF	SL
Resolution [px]	200 x 200	640 x 480
Frame rate [Hz]	40	30
Measurement range [m]	0.3 – 7.0	1.0 – 3.0
Field of view [°]	40 x 40	57 x 43
Noise level σ [mm] (at a working distance of 1 m)	$\pm$ 5.98	$\pm$ 0.92

Table 1: Comparison of the ToF camera model PMD CamCube 3.0 (top left) to the first generation of Kinect sensors for Xbox 360 (top right)

Figure 1: RI data

RI in Abdominal Surgery: Open Surgery

- Fuse pre-operative 3-D planning data and intra-interventional surface measurements:
  - Augmented reality
  - Navigation

- Challenges:
  - Accuracy in a medical environment
  - Real-time requirements in interventional imaging
  - Usability for surgeons and medical staff

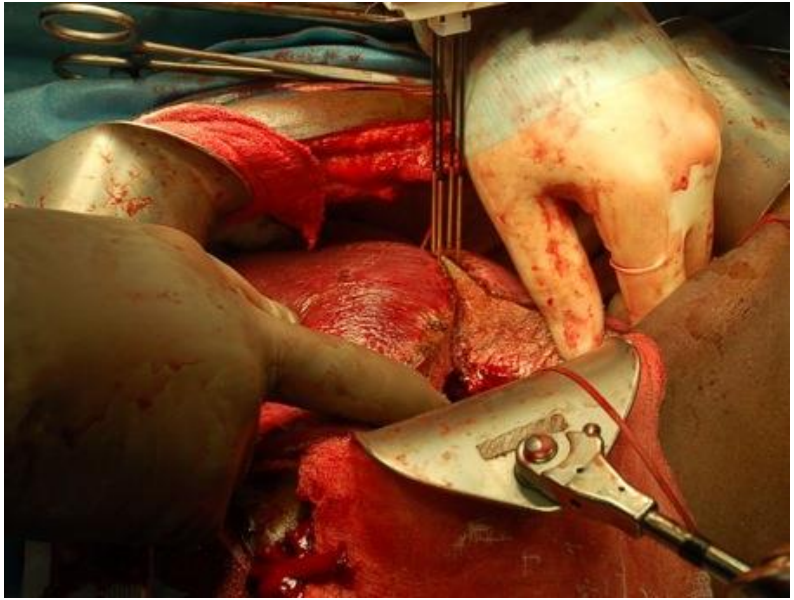


Figure 2: Image courtesy of NJLiverCare.Org

RI in Abdominal Surgery: Endoscopy

- Fuse conventional 2-D RGB endoscopy data with 3-D depth information
  - Measurement of regions of interest
  - Segmentation and tracking of tools
  - Navigation, collision avoidance
- Challenges:
  - Accuracy in a medical environment
  - Real-time requirements in interventional imaging
  - Usability for surgeons and medical staff

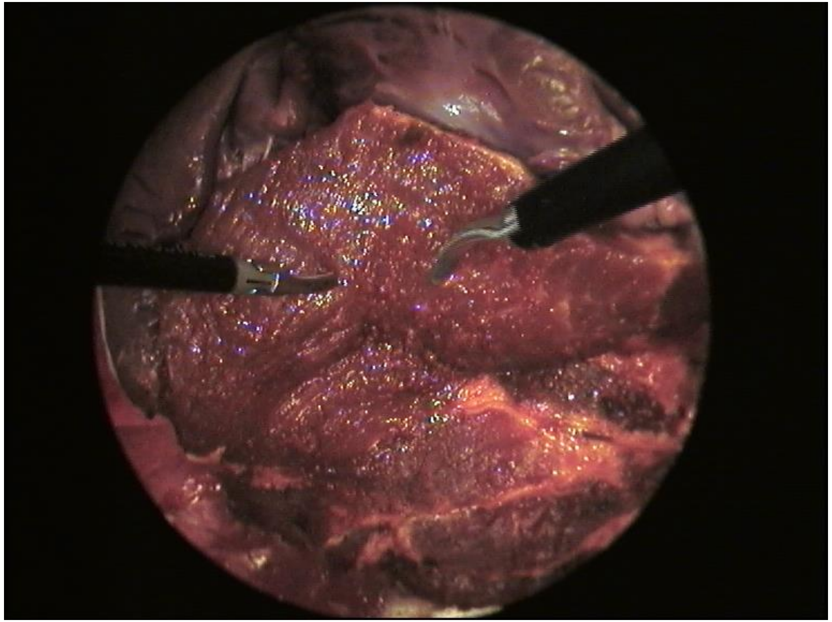


Figure 3: Image courtesy of KIT, Karlsruhe

RI in Abdominal Surgery

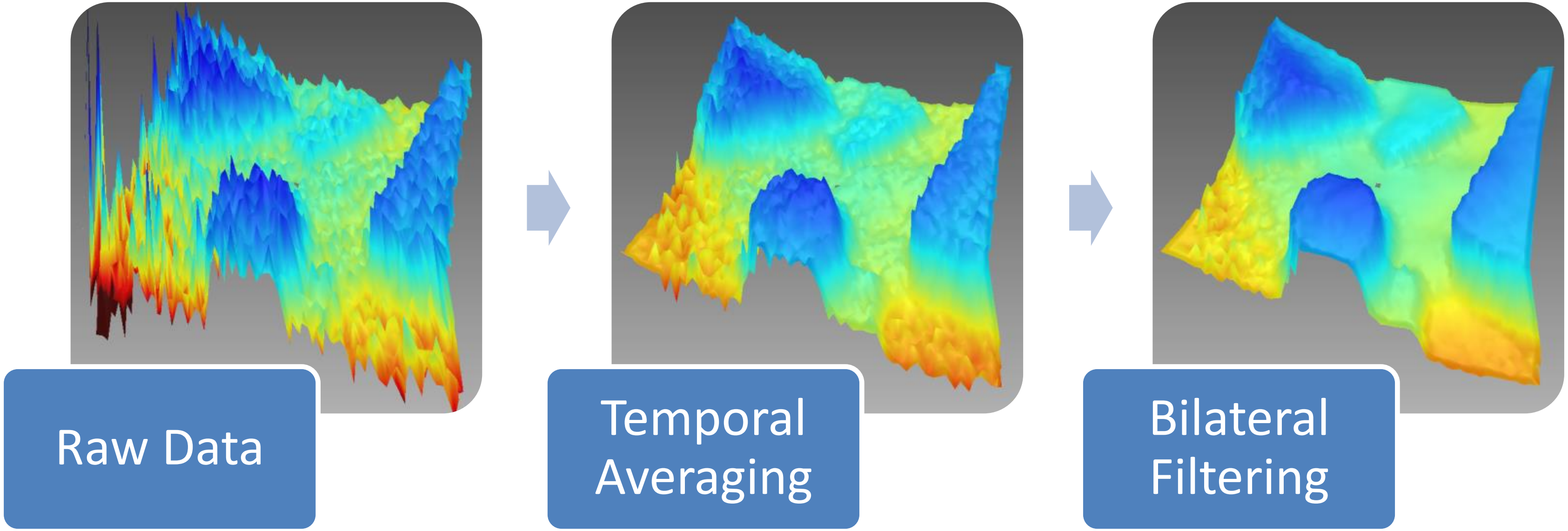
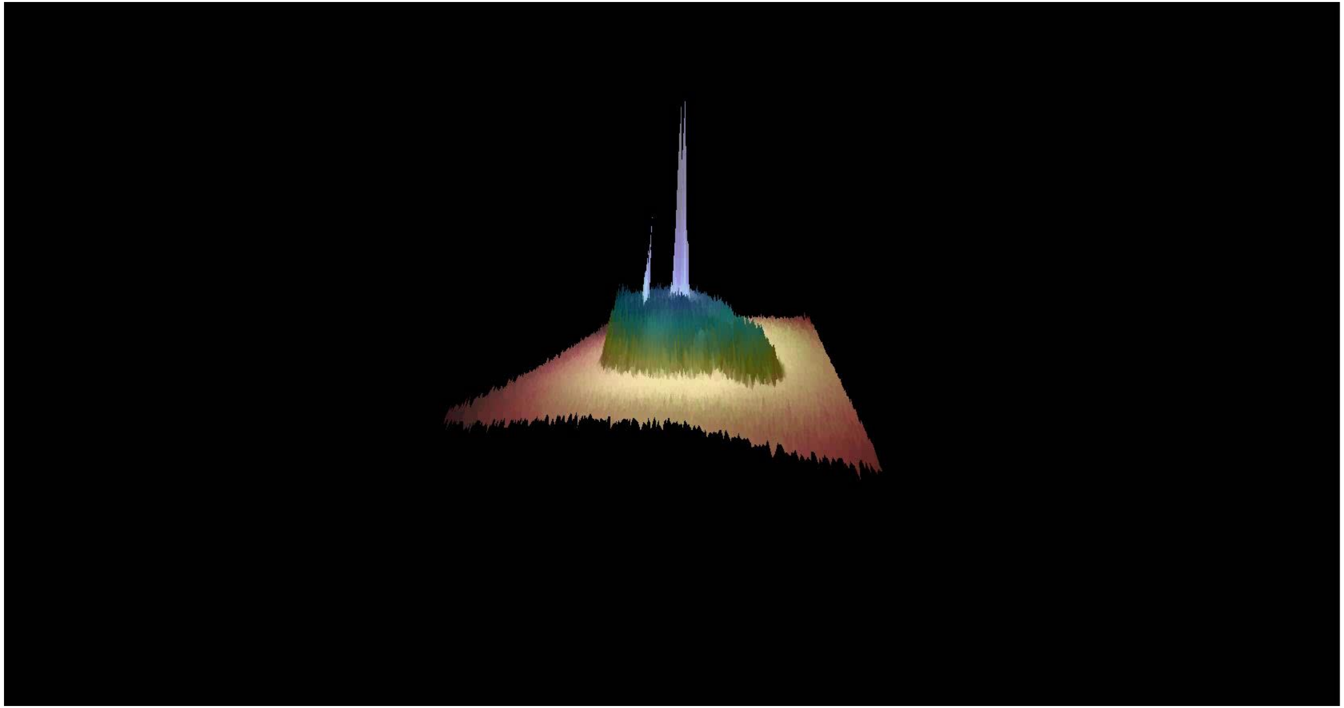


Figure 4: Preprocessing pipeline for interventional range imaging

RI in Abdominal Surgery: Example Video



Video 1: Hover over the static image and click play to watch

Nomenclature

- We consider discrete 2-D images using the following notation:
  - number of pixels N,
  - discrete pixel index $\boldsymbol{x} = (x, y)$,
  - local neighborhood $\omega_{\boldsymbol{x}},\ |\omega_{\boldsymbol{x}}| = (2r+1)^2,\ r = 1,2,\ldots$.

- Filter input → corrupted/noisy image $g(\boldsymbol{x})$
- Filter output → restored/denoised image $f(\boldsymbol{x})$

Refresher: Convolution

- Discrete convolution:

$$f(\boldsymbol{x}) = \{g * \mathcal{K}\}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}')\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')}{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')}$$

- Gaussian kernel:

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{\sigma^2}\right)$$

- Convolution theorem for linear, shift-invariant kernels:

$$g * \mathcal{K} = \mathcal{F}^{-1}\{\mathcal{F}\{g * \mathcal{K}\}\} = \mathcal{F}^{-1}\{\mathcal{F}\{g\} \cdot \mathcal{F}\{\mathcal{K}\}\}$$

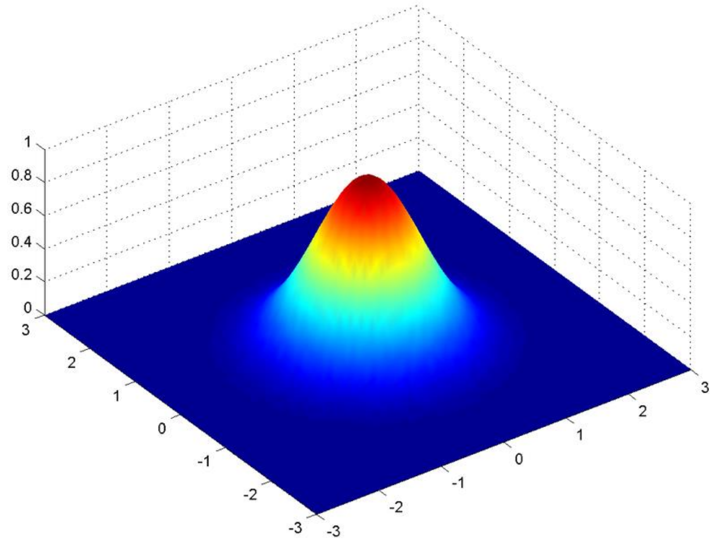


Figure 5: Example of a Gaussian kernel function

Normalized Convolution

- Discrete convolution:

$$f(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}')\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')}{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')}$$

- Normalized convolution (Knutsson and Westin, 1993):

$$f_{\mathrm{NC}}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}')\mathcal{A}(\boldsymbol{x}, \boldsymbol{x}')\mathcal{C}(\boldsymbol{x}')}{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \mathcal{A}(\boldsymbol{x}, \boldsymbol{x}')\mathcal{C}(\boldsymbol{x}')}$$

- Examples for certainty and applicability:

$$\mathcal{C}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } g(\boldsymbol{x}) \text{ is valid} \\ 0, & \text{else} \end{cases}$$

$$\mathcal{A}(\boldsymbol{x}, \boldsymbol{x}') = \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')$$

Bilateral Filtering

Problem: Conventional filters smooth across edges.

Idea: Incorporate edge-stopping functionality based on pixel similarity.

- Bilateral filter (Tomasi and Manduchi, 1998):

$$f_{\mathrm{BF}}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') c(\boldsymbol{x}, \boldsymbol{x}') s\big(g(\boldsymbol{x}), g(\boldsymbol{x}')\big)}{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} c(\boldsymbol{x}, \boldsymbol{x}')\, s\big(g(\boldsymbol{x}), g(\boldsymbol{x}')\big)}$$

- Spatial closeness c and range similarity s:

$$c(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{\sigma_c^2}\right)$$

$$s\big(g(\boldsymbol{x}), g(\boldsymbol{x}')\big) = \exp\left(-\frac{|g(\boldsymbol{x}) - g(\boldsymbol{x}')|^2}{\sigma_s^2}\right)$$

Bilateral Filtering

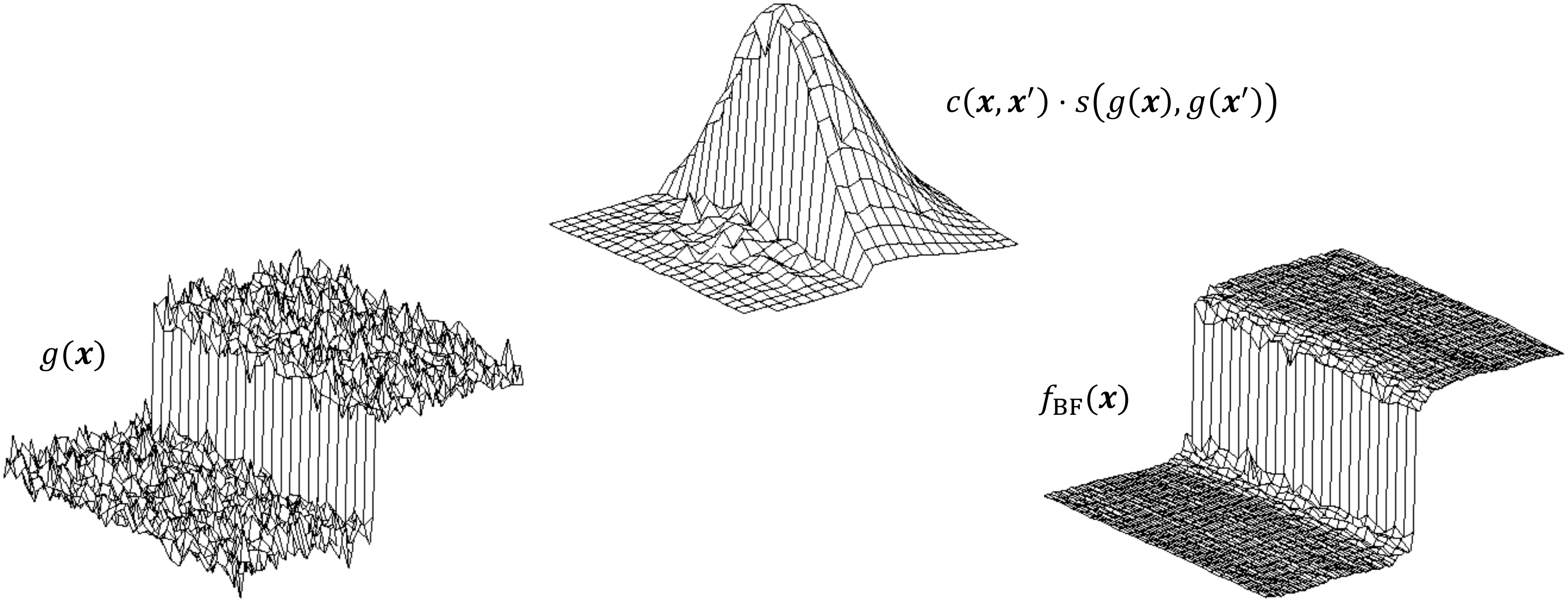


Figure 6: Working principle of the bilateral filter (images from Tomasi and Manduchi, 1998)

Bilateral Filtering

- Properties:
  - Edge-preserving denoising
  - Related to normalized convolution
  - Range similarity term **not** shift-invariant
- Complexity: $\mathcal{O}(Nr^2)$
- Closeness and similarity not restricted to the Gaussian case

Topics

Image Enhancement

Motivation

Normalized Convolution

Bilateral Filtering

Summary

Take Home Messages

Further Readings

Take Home Messages

- By range imaging several imaging applications for surgical treatments are motivated.
- In preparation for the guided filter in the next unit we have learned about normalized convolution and the important bilateral filter.
- The bilateral filter is one example for an edge-preserving filtering method for denoising.

Further Readings

- Hans Knutsson and Carl-Fredrik Westin. "Normalized and Differential Convolution: Methods for Interpolation and Filtering of Incomplete and Uncertain Data". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 1993, pp. 515–523. DOI: 10.1109/CVPR.1993.341081
- Carlo Tomasi and Roberto Manduchi. "Bilateral Filtering for Gray and Color Images". In: *Sixth International Conference on Computer Vision, 1998. Sponsored by the IEEE Computer Society, January 4-7, 1998, Bombay, India*. IEEE, Jan. 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815

Medical Image Processing for Interventional Applications

Guided Filter

Online Course – Unit 16

Andreas Maier, Jakob Wasza, Frank Schebesch
Pattern Recognition Lab (CS 5)

virtuelle hochschule bayern
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Topics

Guided Filter

Summary

Take Home Messages

Further Readings

Guided Filtering

- Proposed at the ECCV 2010 (later published as He, Sun and Tang, 2013)
- Applications:
  - Non-approximative edge-preserving denoising
  - HDR compression
  - Multi-modal image upsampling
  - ...
- Complexity: $\mathcal{O}(N)$
- **Basic idea:** guide the filtering process by a dedicated image $i(\boldsymbol{x})$.

Guided Filtering: Cost Function

Assumption: Output as a linear transform of the guidance $i(\boldsymbol{x})$, i.e.:

$$f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$$

Guided Filtering: Cost Function

Assumption: Output as a linear transform of the guidance $i(\boldsymbol{x})$, i.e.:

$$f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$$

Gradient relationship induced by this model:

$$\nabla f = a \nabla i$$

Guided Filtering: Cost Function

Assumption: Output as a linear transform of the guidance $i(\boldsymbol{x})$, i.e.:

$$f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$$

Gradient relationship induced by this model:

$$\nabla f = a \nabla i$$

Cost function:

$$\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left((a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}'))^2 + \epsilon a_{\boldsymbol{x}}^2 \right)$$

Guided Filtering: Cost Function

Assumption: Output as a linear transform of the guidance $i(\boldsymbol{x})$, i.e.:

$$f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$$

Gradient relationship induced by this model:

$$\nabla f = a \nabla i$$

Cost function:

$$\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \Big(\big(\underbrace{\boxed{a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}}}_{\text{Output}} - g(\boldsymbol{x}') \big)^2 + \epsilon a_{\boldsymbol{x}}^2 \Big)$$

Guided Filtering: Cost Function

Assumption: Output as a linear transform of the guidance $i(\boldsymbol{x})$, i.e.:

$$f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$$

Gradient relationship induced by this model:

$$\nabla f = a \nabla i$$

Cost function:

$$\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left(\left(\overset{\text{Output}}{\boxed{a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}}} - \overset{\text{Input}}{\boxed{g(\boldsymbol{x}')}} \right)^2 + \epsilon a_{\boldsymbol{x}}^2 \right)$$

Guided Filtering: Cost Function

Assumption: Output as a linear transform of the guidance $i(\boldsymbol{x})$, i.e.:

$$f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$$

Gradient relationship induced by this model:

$$\nabla f = a \nabla i$$

Cost function:

$$\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \Big(\big(\underbrace{\boxed{a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}}}_{\text{Output}} - \underbrace{\boxed{g(\boldsymbol{x}')}}_{\text{Input}} \big)^2 + \underbrace{\boxed{\epsilon}}_{\text{Regularization}} a_{\boldsymbol{x}}^2 \Big)$$

Guided Filtering: Partial Derivatives of the Cost Function

Cost function:

$$\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left(\left(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}') \right)^2 + \epsilon a_{\boldsymbol{x}}^2 \right)$$

Guided Filtering: Partial Derivatives of the Cost Function

Cost function:

$$\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left((a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}'))^2 + \epsilon a_{\boldsymbol{x}}^2 \right)$$

Partial derivatives:

$$\frac{\partial}{\partial a_{\boldsymbol{x}}} \mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = 2 \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left((a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}')) i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}} \right) \stackrel{!}{=} 0$$

Guided Filtering: Partial Derivatives of the Cost Function

Cost function:

$$\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left(\left(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}') \right)^2 + \epsilon a_{\boldsymbol{x}}^2 \right)$$

Partial derivatives:

$$\frac{\partial}{\partial a_{\boldsymbol{x}}} \mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = 2 \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left(\left(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}') \right) i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}} \right) \stackrel{!}{=} 0$$

$$\frac{\partial}{\partial b_{\boldsymbol{x}}} \mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = 2 \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}') \right)$$

$$= 2 a_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + 2 b_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} 1 - 2 \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') \stackrel{!}{=} 0$$

Guided Filtering: Partial Derivatives of the Cost Function

Cost function:

$$\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left(\left(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}') \right)^2 + \epsilon a_{\boldsymbol{x}}^2 \right)$$

Partial derivatives:

$$\frac{\partial}{\partial a_{\boldsymbol{x}}} \mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = 2 \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left(\left(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}') \right) i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}} \right) \stackrel{!}{=} 0$$

$$\frac{\partial}{\partial b_{\boldsymbol{x}}} \mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = 2 \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}') \right)$$

$$= 2 a_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + 2 b_{\boldsymbol{x}} \boxed{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} 1} - 2 \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') \stackrel{!}{=} 0$$

$$\boxed{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} 1 = |\omega_{\boldsymbol{x}}|}$$

Relation between Parameter $b_{\boldsymbol{x}}$ and Mean Filtering

Deriving $b_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial b_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = a_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}|\omega_{\boldsymbol{x}}| - \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') \stackrel{!}{=} 0$$

Relation between Parameter $b_{\boldsymbol{x}}$ and Mean Filtering

Deriving $b_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial b_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = a_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}|\omega_{\boldsymbol{x}}| - \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') \overset{!}{=} 0$$

$$b_{\boldsymbol{x}} = \frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') - a_{\boldsymbol{x}} \left(\frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') \right)$$

Relation between Parameter $b_{\boldsymbol{x}}$ and Mean Filtering

Deriving $b_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial b_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = a_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}|\omega_{\boldsymbol{x}}| - \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') \stackrel{!}{=} 0$$

$$b_{\boldsymbol{x}} = \boxed{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}}} g(\boldsymbol{x}') - a_{\boldsymbol{x}}\left(\boxed{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}}} i(\boldsymbol{x}')\right)$$

mean/box filter

Relation between Parameter $b_{\boldsymbol{x}}$ and Mean Filtering

Deriving $b_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial b_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = a_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} |\omega_{\boldsymbol{x}}| - \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') \overset{!}{=} 0$$

$$b_{\boldsymbol{x}} = \boxed{\frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}}} g(\boldsymbol{x}') - a_{\boldsymbol{x}} \left(\boxed{\frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}}} i(\boldsymbol{x}') \right)$$

mean/box filter

Probabilistic formulation: Mean filtering yields the expectation value $\mathrm{E}_{\omega_{\boldsymbol{x}}}$ if we interprete $g(\boldsymbol{x}), i(\boldsymbol{x})$ as uniformly distributed random variables.

Relation between Parameter $b_{\boldsymbol{x}}$ and Mean Filtering

Deriving $b_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial b_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = a_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}|\omega_{\boldsymbol{x}}| - \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') \stackrel{!}{=} 0$$

$$b_{\boldsymbol{x}} = \boxed{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}}} g(\boldsymbol{x}') - a_{\boldsymbol{x}}\left(\boxed{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}}} i(\boldsymbol{x}')\right)$$

mean/box filter

Probabilistic formulation: Mean filtering yields the expectation value $\mathrm{E}_{\omega_{\boldsymbol{x}}}$ if we interprete $g(\boldsymbol{x}), i(\boldsymbol{x})$ as uniformly distributed random variables.

$$\Rightarrow \quad b_{\boldsymbol{x}} = E_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] - a_{\boldsymbol{x}} E_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})]$$

Relation between Parameter $a_{\boldsymbol{x}}$ and Mean Filtering

Deriving $a_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial a_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \Big((a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}')) i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}} \Big) \overset{!}{=} 0$$

Relation between Parameter $a_{\boldsymbol{x}}$ and Mean Filtering

Deriving $a_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial a_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left((a_{\boldsymbol{x}} i(\boldsymbol{x}') + \boxed{b_{\boldsymbol{x}}} - g(\boldsymbol{x}'))i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}} \right) \overset{!}{=} 0$$

$$b_{\boldsymbol{x}} = E_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] - a_{\boldsymbol{x}} E_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})]$$

Relation between Parameter $a_{\boldsymbol{x}}$ and Mean Filtering

Deriving $a_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial a_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left((a_{\boldsymbol{x}} i(\boldsymbol{x}') + \boxed{b_{\boldsymbol{x}}} - g(\boldsymbol{x}')) i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}} \right) \stackrel{!}{=} 0$$

$$b_{\boldsymbol{x}} = E_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] - a_{\boldsymbol{x}} E_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})]$$

Some calculations later …

$$a_{\boldsymbol{x}} \left(\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})] \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + \epsilon \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} 1 \right) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}')$$

Relation between Parameter $a_{\boldsymbol{x}}$ and Mean Filtering

Deriving $a_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial a_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} \left((a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}'))i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}}\right) \overset{!}{=} 0$$

Some calculations later …

$$\cdot \frac{1}{|\omega_{\boldsymbol{x}}|}$$

$$\boxed{a_{\boldsymbol{x}}\left(\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}')i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})] \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + \epsilon \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} 1\right)} = \boxed{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}')i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} i(\boldsymbol{x}')}$$

Relation between Parameter a_x and Mean Filtering

Deriving a_x:

$$\frac{1}{2}\frac{\partial}{\partial a_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}}\Big(\big(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}')\big)i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}}\Big) \overset{!}{=} 0$$

Some calculations later …

$$\cdot\frac{1}{|\omega_{\boldsymbol{x}}|}$$

$$\boxed{a_{\boldsymbol{x}}\left(\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} i(\boldsymbol{x}')i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})]\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + \epsilon\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} 1\right)} = \boxed{\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} g(\boldsymbol{x}')i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})]\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} i(\boldsymbol{x}')}$$

$$\Rightarrow\quad a_{\boldsymbol{x}}\left(\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} i(\boldsymbol{x}')i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})]\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} i(\boldsymbol{x}') + \epsilon\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} 1\right)$$

$$= \frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} g(\boldsymbol{x}')i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})]\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} i(\boldsymbol{x}')$$

Relation between Parameter $a_{\boldsymbol{x}}$ and Mean Filtering

Deriving $a_{\boldsymbol{x}}$:

$$\frac{1}{2}\frac{\partial}{\partial a_{\boldsymbol{x}}}\mathcal{J}(a_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}}\Big(\big(a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}} - g(\boldsymbol{x}')\big)i(\boldsymbol{x}') + \epsilon a_{\boldsymbol{x}}\Big) \overset{!}{=} 0$$

Some calculations later …

mean/box filter

$$\Rightarrow \quad a_{\boldsymbol{x}}\left(\boxed{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}}} i(\boldsymbol{x}')i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})]\boxed{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}}} i(\boldsymbol{x}') + \epsilon \cancel{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}} 1}\right)$$

$$= \boxed{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}}} g(\boldsymbol{x}')i(\boldsymbol{x}') - E_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})]\boxed{\frac{1}{|\omega_{\boldsymbol{x}}|}\sum_{\boldsymbol{x}'\in\omega_{\boldsymbol{x}}}} i(\boldsymbol{x}')$$

Relation between Parameter $a_{\boldsymbol{x}}$ and Mean Filtering

Therefore we found:

$$a_{\boldsymbol{x}}\left(\mathrm{E}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})i(\boldsymbol{x})] - \mathrm{E}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})]\mathrm{E}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})] + \epsilon\right) = \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})i(\boldsymbol{x})] - \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})]\mathrm{E}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})].$$

Relation between Parameter $a_{\boldsymbol{x}}$ and Mean Filtering

Therefore we found:

$$a_{\boldsymbol{x}}\left(\mathrm{E}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})i(\boldsymbol{x})] - \mathrm{E}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})]\mathrm{E}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})] + \epsilon\right) = \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})i(\boldsymbol{x})] - \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})]\mathrm{E}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})].$$

Using the computational formulas for variance and covariance:

$$\mathrm{Var}(X) = \mathrm{E}[X^2] - \mathrm{E}[X]^2, \qquad \text{and} \qquad \mathrm{Cov}[X, Y] = \mathrm{E}[XY] - \mathrm{E}[X]\mathrm{E}[Y],$$

we finally obtain:

$$a_{\boldsymbol{x}} = \frac{\mathrm{Cov}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x}), i(\boldsymbol{x})]}{\mathrm{Var}_{\omega_{\boldsymbol{x}}}[i(\boldsymbol{x})] + \epsilon}.$$

Special Cases

- Guided filtering, linear model: $f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$
- Lets consider the case that $i(\boldsymbol{x}) = g(\boldsymbol{x})$:

$$a_{\boldsymbol{x}} = \frac{\mathrm{Var}_{\omega_{\boldsymbol{x}}}\big(g(\boldsymbol{x})\big)}{\mathrm{Var}_{\omega_{\boldsymbol{x}}}\big(g(\boldsymbol{x})\big) + \epsilon}, \qquad b_{\boldsymbol{x}} = \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] - a_{\boldsymbol{x}} \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})].$$

Special Cases

- Guided filtering, linear model: $f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$
- Lets consider the case that $i(\boldsymbol{x}) = g(\boldsymbol{x})$:

$$a_{\boldsymbol{x}} = \frac{\mathrm{Var}_{\omega_{\boldsymbol{x}}}\big(g(\boldsymbol{x})\big)}{\mathrm{Var}_{\omega_{\boldsymbol{x}}}\big(g(\boldsymbol{x})\big) + \epsilon}, \qquad b_{\boldsymbol{x}} = \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] - a_{\boldsymbol{x}} \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})].$$

Special Cases

- Guided filtering, linear model: $f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$
- Lets consider the case that $i(\boldsymbol{x}) = g(\boldsymbol{x})$:

$$a_{\boldsymbol{x}} = \frac{\mathrm{Var}_{\omega_{\boldsymbol{x}}}\left(g(\boldsymbol{x})\right)}{\mathrm{Var}_{\omega_{\boldsymbol{x}}}\left(g(\boldsymbol{x})\right) + \epsilon}, \qquad b_{\boldsymbol{x}} = \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] - a_{\boldsymbol{x}} \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})].$$

- Trivial case $\epsilon = 0$:
  - **Flat patch**: if $g(\boldsymbol{x})$ is constant across $\omega_{\boldsymbol{x}}$: $a_{\boldsymbol{x}} = 0, b_{\boldsymbol{x}} = \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})]$
  - **High variance**: if $g(\boldsymbol{x})$ changes across $\omega_{\boldsymbol{x}}$: $a_{\boldsymbol{x}} = 1, b_{\boldsymbol{x}} = 0$

Special Cases

- Guided filtering, linear model: $f(\boldsymbol{x}') = a_{\boldsymbol{x}} i(\boldsymbol{x}') + b_{\boldsymbol{x}}, \forall \boldsymbol{x}' \in \omega_{\boldsymbol{x}}$
- Lets consider the case that $i(\boldsymbol{x}) = g(\boldsymbol{x})$:

$$a_{\boldsymbol{x}} = \frac{\mathrm{Var}_{\omega_{\boldsymbol{x}}}\big(g(\boldsymbol{x})\big)}{\mathrm{Var}_{\omega_{\boldsymbol{x}}}\big(g(\boldsymbol{x})\big) + \epsilon}, \qquad b_{\boldsymbol{x}} = \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] - a_{\boldsymbol{x}} \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})].$$

- Trivial case $\epsilon = 0$:
  - **Flat patch**: if $g(\boldsymbol{x})$ is constant across $\omega_{\boldsymbol{x}}$: $a_{\boldsymbol{x}} = 0, b_{\boldsymbol{x}} = \mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})]$
  - **High variance**: if $g(\boldsymbol{x})$ changes across $\omega_{\boldsymbol{x}}$: $a_{\boldsymbol{x}} = 1, b_{\boldsymbol{x}} = 0$
- The criterion of **flat patch** or **high variance** is given by ϵ.

Guided Filtering

- So far, only **one** local window was considered.
  - → Apply the model to **all** local windows.
  - → Pixel $\boldsymbol{x}$ is involved in all local windows $\omega_{\boldsymbol{x}'}$ that contain $\boldsymbol{x}$.
  - → Average all possible local coefficients $a_{\boldsymbol{x}'}, b_{\boldsymbol{x}'}$.

Guided Filtering

- So far, only **one** local window was considered.
  - → Apply the model to **all** local windows.
  - → Pixel $\boldsymbol{x}$ is involved in all local windows $\omega_{\boldsymbol{x}'}$ that contain $\boldsymbol{x}$.
  - → Average all possible local coefficients $a_{\boldsymbol{x}'}, b_{\boldsymbol{x}'}$.
- Final filter output for all local windows:

$$f_{\mathrm{GF}}(\boldsymbol{x}) = \frac{1}{|\omega_{\boldsymbol{x}'}|} \sum_{\boldsymbol{x}' :\, \boldsymbol{x} \in \omega_{\boldsymbol{x}'}} \left(a_{\boldsymbol{x}'} i(\boldsymbol{x}) + b_{\boldsymbol{x}'}\right) = \mathrm{E}_{\omega_{\boldsymbol{x}'}}[a_{\boldsymbol{x}}] i(\boldsymbol{x}) + \mathrm{E}_{\omega_{\boldsymbol{x}'}}[b_{\boldsymbol{x}}]$$

Guided Filtering

- So far, only **one** local window was considered.
  - → Apply the model to **all** local windows.
  - → Pixel $\boldsymbol{x}$ is involved in all local windows $\omega_{\boldsymbol{x}'}$ that contain $\boldsymbol{x}$.
  - → Average all possible local coefficients $a_{\boldsymbol{x}'}, b_{\boldsymbol{x}'}$.
- Final filter output for all local windows:

$$f_{\mathrm{GF}}(\boldsymbol{x}) = \frac{1}{\left|\omega_{\boldsymbol{x}'}\right|} \sum_{\boldsymbol{x}' :\, \boldsymbol{x} \in \omega_{\boldsymbol{x}'}} \left(a_{\boldsymbol{x}'} i(\boldsymbol{x}) + b_{\boldsymbol{x}'}\right) = \mathrm{E}_{\omega_{\boldsymbol{x}'}}[a_{\boldsymbol{x}}] i(\boldsymbol{x}) + \mathrm{E}_{\omega_{\boldsymbol{x}'}}[b_{\boldsymbol{x}}]$$

➔ **Guided filtering can be expressed with mean filters only!**

Guided Filtering

- Mean or box filtering is the backbone of guided filtering:

$$\mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] = \frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') = \{g * \mathcal{M}_r\}(\boldsymbol{x})$$

Guided Filtering

- Mean or box filtering is the backbone of guided filtering:

$$\mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] = \frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') = \{g * \mathcal{M}_r\}(\boldsymbol{x})$$

- Integral images (Viola and Jones, 2001):

$$\vartheta(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} g(x', y')$$

Guided Filtering

- Mean or box filtering is the backbone of guided filtering:

$$\mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] = \frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') = \{g * \mathcal{M}_r\}(\boldsymbol{x})$$

- Integral images (Viola and Jones, 2001):

$$\vartheta(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} g(x', y')$$

- Mean filtering using integral images:

$$\{g * \mathcal{M}_r\}(x, y) = \big(\vartheta(x + r, y + r) - \vartheta(x + r, y - r - 1)\big) - \big(\vartheta(x - r - 1, y + r) - \vartheta(x - r - 1, y - r - 1)\big)$$

Guided Filtering

- Mean or box filtering is the backbone of guided filtering:

$$\mathrm{E}_{\omega_{\boldsymbol{x}}}[g(\boldsymbol{x})] = \frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}') = \{g * \mathcal{M}_r\}(\boldsymbol{x})$$

- Integral images (Viola and Jones, 2001):

$$\vartheta(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} g(x', y')$$

- Mean filtering using integral images:

$$\mathcal{O}(1)$$

$$\{g * \mathcal{M}_r\}(x, y) = \boxed{\big(\vartheta(x+r, y+r) - \vartheta(x+r, y-r-1)\big) - \big(\vartheta(x-r-1, y+r) - \vartheta(x-r-1, y-r-1)\big)}$$

Example

1	32	20	3	31	16
8	15	16	21	1	8
30	9	26	13	18	16
26	22	18	8	30	19
29	24	1	21	19	3
17	12	11	24	29	2

Figure 1: Image data

1	33	53	56	87	103
9	56	92	116	148	172
39	95	157	194	244	284
65	143	223	268	348	407
94	196	277	343	442	504
111	225	317	407	535	599

Figure 2: Integral image representation

Example

$\mathcal{M}_1$

1	32	20	3	31	16
8	15	16	21	1	8
30	9	26	13	18	16
26	22	18	8	30	19
29	24	1	21	19	3
17	12	11	24	29	2

Figure 1: Image data

1	33	53	56	87	103
9	56	92	116	148	172
39	95	157	194	244	284
65	143	223	268	348	407
94	196	277	343	442	504
111	225	317	407	535	599

Figure 2: Integral image representation

Example

$\mathcal{M}_1$

1	32	20	3	31	16
8	15	16	21	1	8
30	9	26	13	18	16
26	22	18	8	30	19
29	24	1	21	19	3
17	12	11	24	29	2

Figure 1: Image data

1	33	53	56	87	103
9	56	92	116	148	172
39	95	157	194	244	284
65	143	223	268	348	407
94	196	277	343	442	504
111	225	317	407	535	599

Figure 2: Integral image representation

$$26 + 13 + 18 + 18 + 8 + 30 + 1 + 21 + 19 = 154$$

Example

$\vartheta(x-r-1, y-r-1)$

$\mathcal{M}_1$

1	32	20	3	31	16
8	15	16	21	1	8
30	9	26	13	18	16
26	22	18	8	30	19
29	24	1	21	19	3
17	12	11	24	29	2

Figure 1: Image data

1	33	53	56	87	103
9	56	92	116	148	172
39	95	157	194	244	284
65	143	223	268	348	407
94	196	277	343	442	504
111	225	317	407	535	599

Figure 2: Integral image representation

$\vartheta(x+r, y+r)$

$$26 + 13 + 18 + 18 + 8 + 30 + 1 + 21 + 19 = 154$$

Example

$\mathcal{M}_1$

1	32	20	3	31	16
8	15	16	21	1	8
30	9	26	13	18	16
26	22	18	8	30	19
29	24	1	21	19	3
17	12	11	24	29	2

Figure 1: Image data

$\vartheta(x-r-1, y-r-1)$

$\vartheta(x+r, y-r-1)$

$\vartheta(x-r-1, y+r)$

$\vartheta(x+r, y+r)$

1	33	53	56	87	103
9	56	92	116	148	172
39	95	157	194	244	284
65	143	223	268	348	407
94	196	277	343	442	504
111	225	317	407	535	599

Figure 2: Integral image representation

$$26+13+18+18+8+30+1+21+19=154$$

$$(442-148)-(196-56)=154$$

Topics

Guided Filter

Summary

Take Home Messages

Further Readings

Take Home Messages

- As the name expresses, a guided filter makes use of a guidance image to model, e.g., a smooth but edge-preserving filter.
- From the derivation of a guided filter by using a model that is linear in a neighborhood $\omega_{\boldsymbol{x}}$ of $\boldsymbol{x}$, we found a relationship of the filter with mean filtering of both guidance and target image.
- Mean or box filtering can be computed efficiently using integral images.

Further Readings

- Kaiming He, Jian Sun, and Xiaoou Tang. "Guided Image Filtering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.6 (June 2013), pp. 1397–1409. DOI: 10.1109/TPAMI.2012.213
- Paul Viola and Michael Jones. "Rapid Object Detection Using a Boosted Cascade of Simple Features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. IEEE, Dec. 2001, pp. I-511–I-518. DOI: 10.1109/CVPR.2001.990517

Medical Image Processing for Interventional Applications

Real-time Preprocessing using GPUs

Online Course – Unit 17

Andreas Maier, Jakob Wasza, Frank Schebesch
Pattern Recognition Lab (CS 5)

virtuelle hochschule bayern
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Topics

Real-time Preprocessing using GPUs

Implementation Strategies

Numerical Issues

Summary

Take Home Messages

Further Readings

Motivation: Why Use GPUs?

Challenge: Real-time capable preprocessing pipeline

→ Time constraints in interventional imaging

→ Driving business decisions: “to buy or not to buy”

Solution: GPU based implementations

- High degree of parallelism in the GPU architecture
- Concurrent execution of image processing tasks
- Hardware accelerated interpolation

Figure 1: NVIDIA GTX 680 (left), Quadro K5000 (middle), Tesla K20X (right) (image source: Nvidia)

Hardware Architecture for CPUs and GPUs

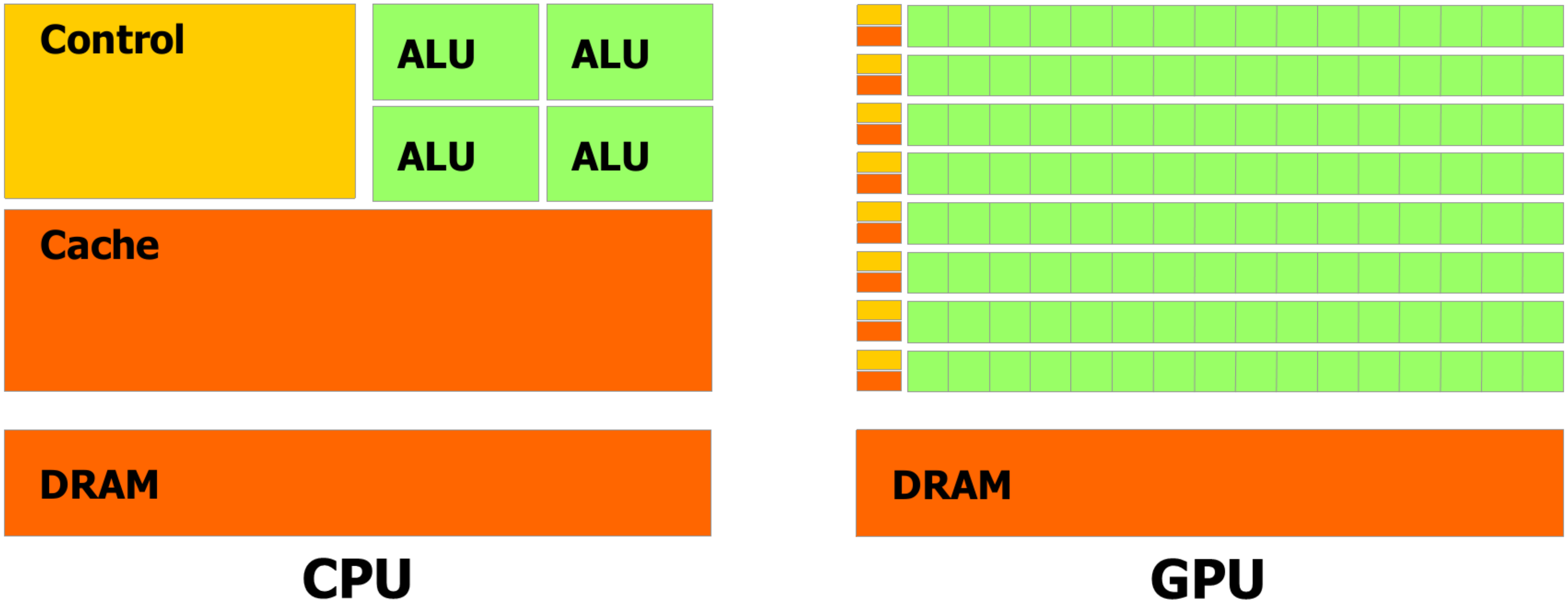


Figure 2: These schematics of CPU (left) and GPU (right) architecture show that the GPU devotes more transistors to data processing.

Application to Preprocessing

- Recall the bilateral filter:

$$f_{\mathrm{BF}}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}) c(\boldsymbol{x}, \boldsymbol{x}') s\big(g(\boldsymbol{x}), g(\boldsymbol{x}')\big)}{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} c(\boldsymbol{x}, \boldsymbol{x}') s\big(g(\boldsymbol{x}), g(\boldsymbol{x}')\big)}$$

- Has to be computed for all pixels $\boldsymbol{x}$
  - → Assign a (light-weighted) GPU thread to each pixel
  - → Massive parallelization
- Memory access patterns in $\omega_{\boldsymbol{x}}$ and boundary conditions
  - → Texture cache designed for 2-D spatial locality
  - → Texture clamping
- Re-use components (spatial closeness)

Numerical Issues

- Single-precision floating-point numbers:
  - IEEE 754-2008 standard
  - Seven decimal digits
- Output is different from the mathematically exact result:
  - rounding errors or accuracy,
  - overflow.
- Consequence for guided filtering:
  - In particular: $\mathrm{Cov}(X, Y) \neq \mathrm{E}(XY) - \mathrm{E}(X)\mathrm{E}(Y)$
  - Integral images → exploit the separability of mean filtering: $X * \mathcal{M}_r = \boldsymbol{m}_r^T * (\boldsymbol{m}_r * X), \boldsymbol{m}_r = [1, \dots, 1]$

Real-time Preprocessing Using GPUs

Figure 3: Erroneous covariances using std integral images and mean filtering

Figure 4: Erroneous covariances using separable mean filtering

Real-time Preprocessing Using GPUs

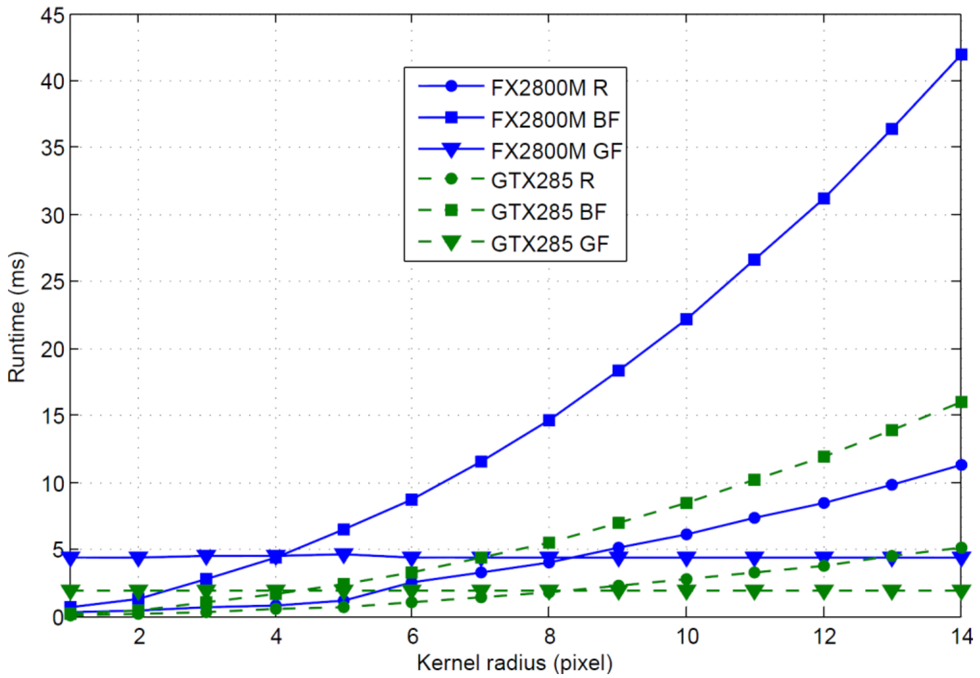


Figure 5: Filter runtimes for VGA resolution (640x480)

Topics

Real-time Preprocessing using GPUs

- Implementation Strategies
- Numerical Issues

Summary

- Take Home Messages
- Further Readings

Take Home Messages

- Preprocessing is a fundamental step which includes the restoration of invalid values and edge-preserving denoising.
- In order to accelerate computation, we discussed algorithmic approaches and the use of GPU hardware.
- The GPU can be a powerful tool when parallelizable preprocessing steps are performed on an image.

Further Readings

Collection of the literature used in the last units on bilateral and guided filtering:

- Hans Knutsson and Carl-Fredrik Westin. "Normalized and Differential Convolution: Methods for Interpolation and Filtering of Incomplete and Uncertain Data". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 1993, pp. 515–523. DOI: 10.1109/CVPR.1993.341081
- Carlo Tomasi and Roberto Manduchi. "Bilateral Filtering for Gray and Color Images". In: *Sixth International Conference on Computer Vision, 1998. Sponsored by the IEEE Computer Society, January 4-7, 1998, Bombay, India*. IEEE, Jan. 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815
- Kaiming He, Jian Sun, and Xiaoou Tang. "Guided Image Filtering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.6 (June 2013), pp. 1397–1409. DOI: 10.1109/TPAMI.2012.213
- Paul Viola and Michael Jones. "Rapid Object Detection Using a Boosted Cascade of Simple Features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. IEEE, Dec. 2001, pp. I-511–I-518. DOI: 10.1109/CVPR.2001.990517

Medical Image Processing for Interventional Applications

Energy-Resolving Imaging

Online Course – Unit 18

Andreas Maier, Frank Schebesch
Pattern Recognition Lab (CS 5)

virtuelle hochschule bayern

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Topics

X-ray Projections From Energy-Resolving Detectors

Monochromatic Material Decomposition

Joint Bilateral Filter

Summary

- Take Home Messages
- Further Readings

X-ray Spectrum

- X-ray radiation is typically polychromatic.

- For different materials, the amount of absorbed photons is depending on the photons‘ energy.

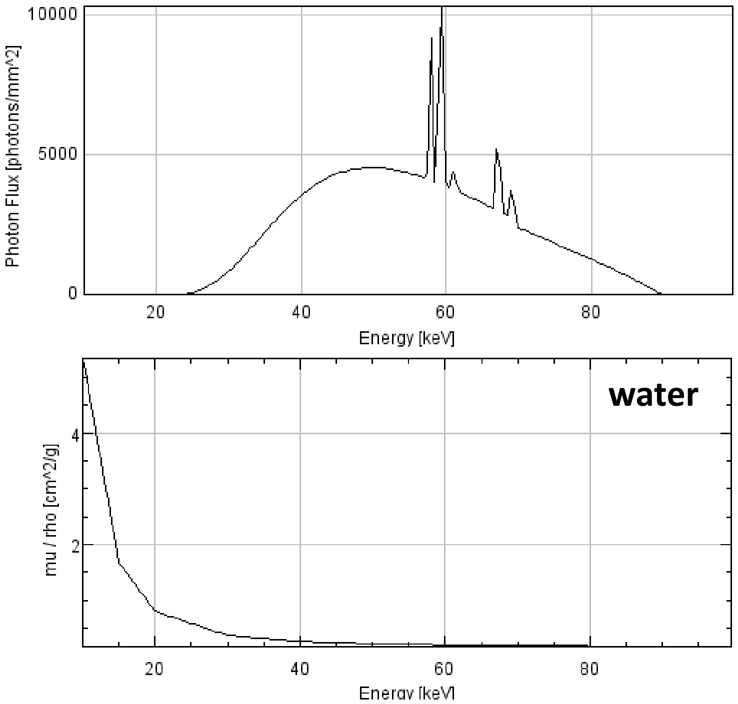


Figure 1: Example for a spectrum of bremsstrahlung (top), and the absorption spectrum of water (bottom)

X-ray Spectrum

- X-ray radiation is typically polychromatic.

- For different materials, the amount of absorbed photons is depending on the photons‘ energy.

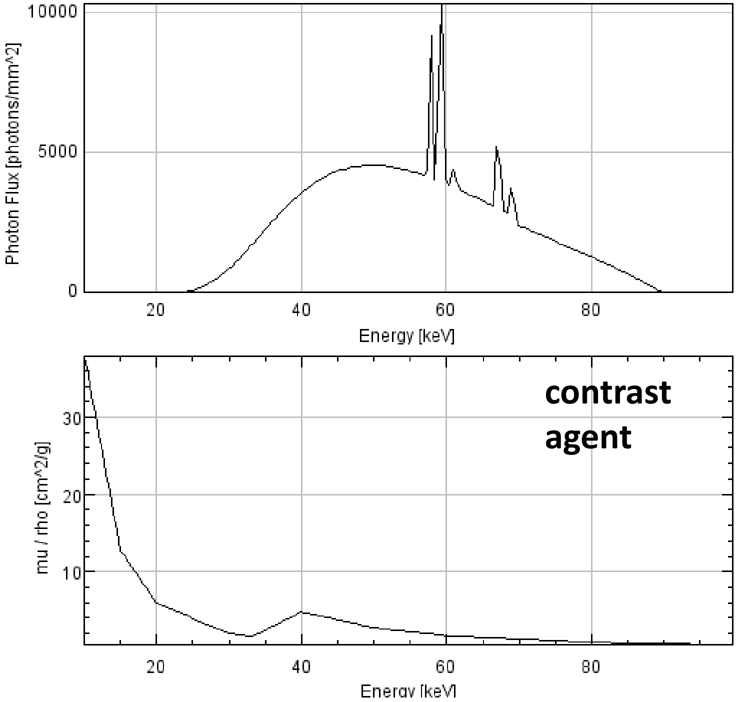


Figure 2: Example for a spectrum of bremsstrahlung (top), and the absorption spectrum of a contrast agent (bottom)

X-ray Projection

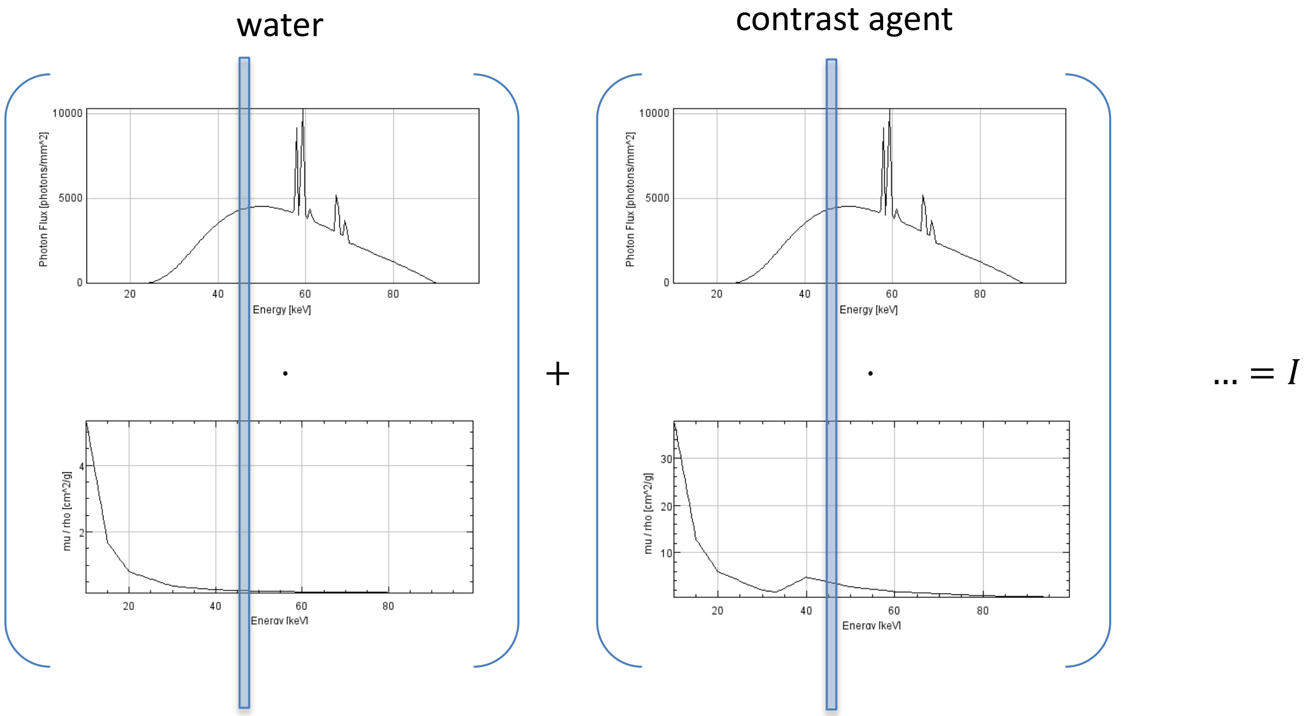


Figure 3: An X-ray projection is the combined result of remaining photons after material dependent absorption.

Energy-Resolving Detectors

- Recent detectors can measure energy levels of X-ray radiation.
- Photons can be distinguished between energies.
- This yields so-called **bins**.

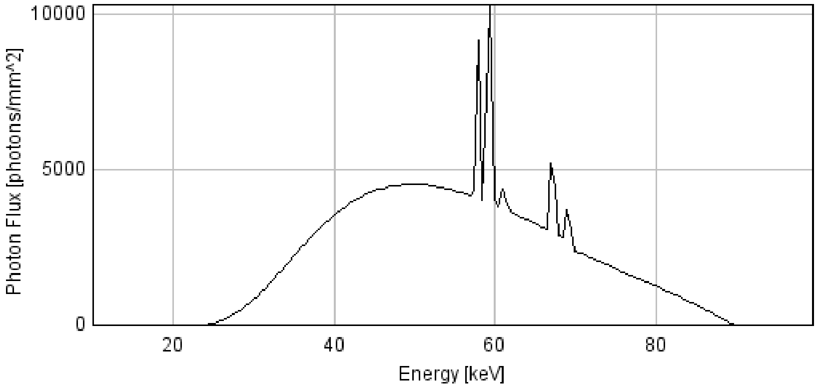


Figure 4: Spectrum subdivided into three energy bins

Energy-Resolving Detectors

- Recent detectors can measure energy levels of X-ray radiation.
- Photons can be distinguished between energies.
- This yields so-called **bins**.

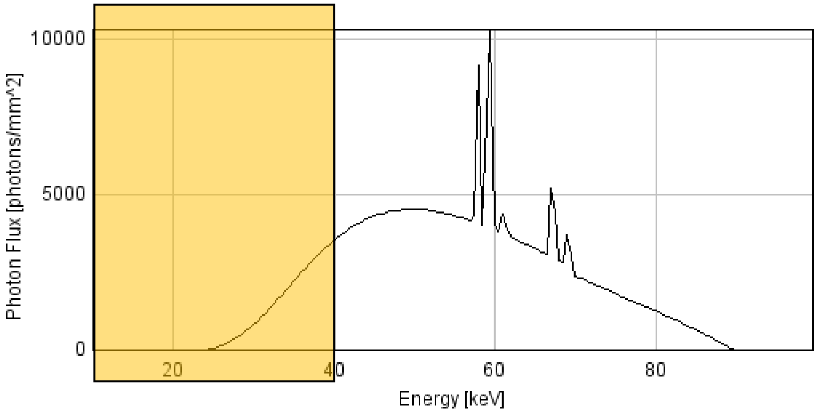


Figure 4: Spectrum subdivided into three energy bins

Energy-Resolving Detectors

- Recent detectors can measure energy levels of X-ray radiation.
- Photons can be distinguished between energies.
- This yields so-called **bins**.

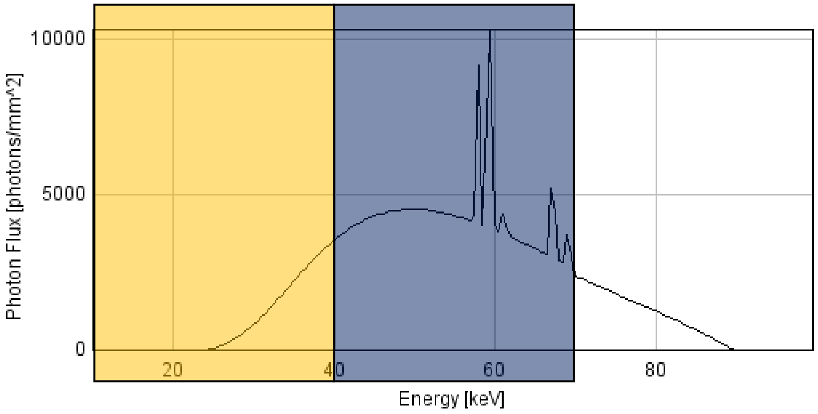


Figure 4: Spectrum subdivided into three energy bins

Energy-Resolving Detectors

- Recent detectors can measure energy levels of X-ray radiation.
- Photons can be distinguished between energies.
- This yields so-called **bins**.

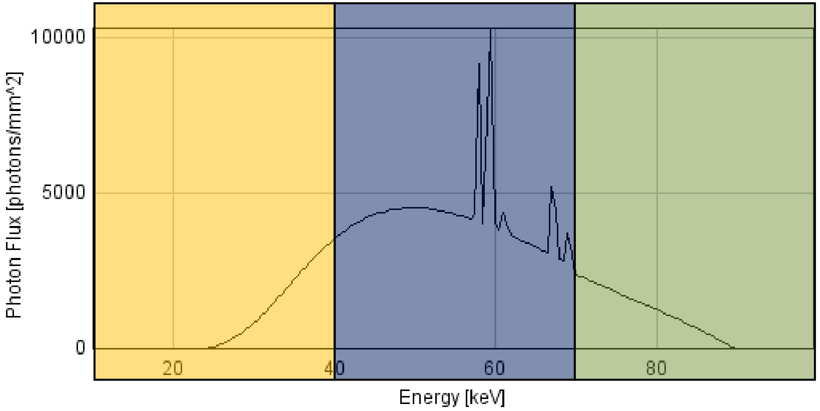


Figure 4: Spectrum subdivided into three energy bins

Simulation

Material images as input:

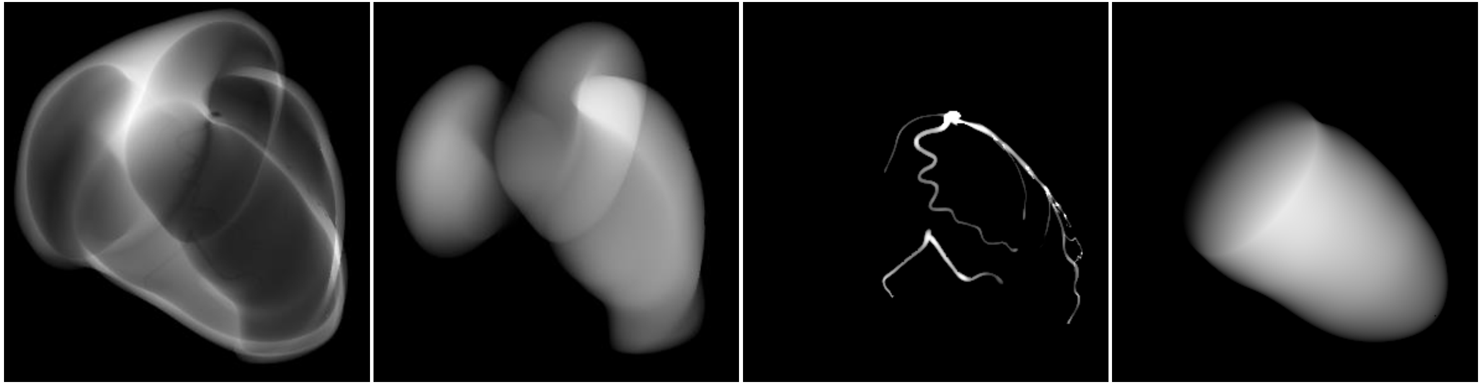


Figure 5: **Myocardium** (left), **blood** (middle left), **Ultravist 370** (middle right), blood of **left ventricle** (right)

Simulation

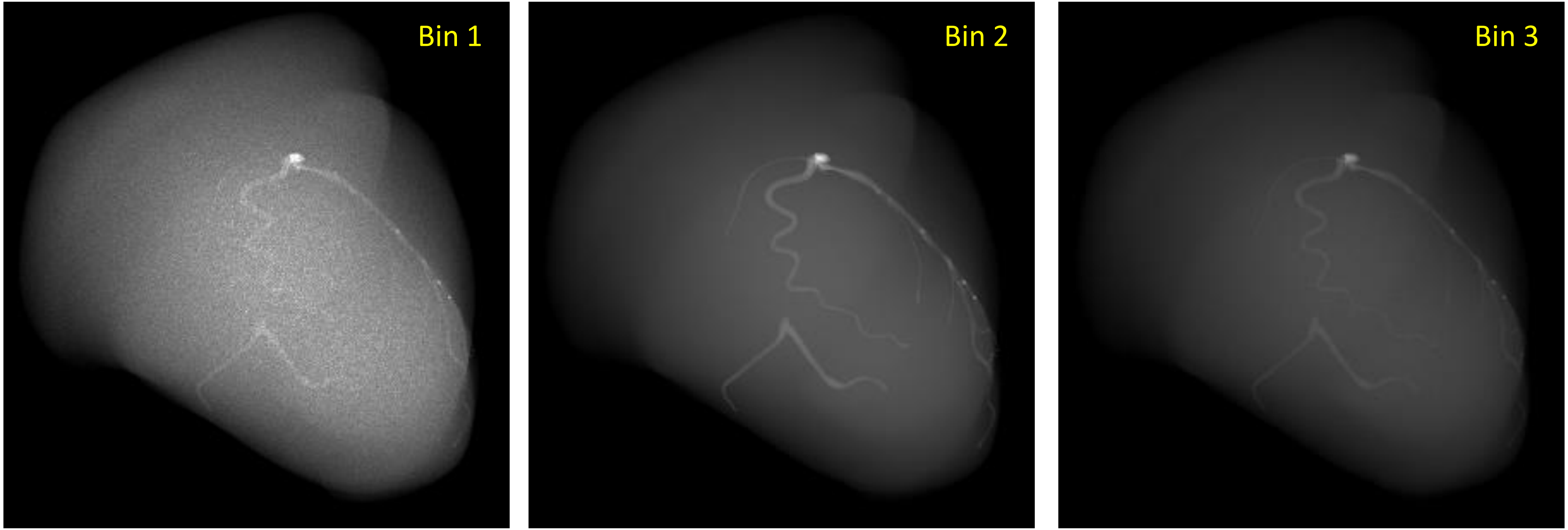


Figure 6: Energy-resolved images, bins 1-3 from left to right, (generated using CONRAD)

Topics

X-ray Projections From Energy-Resolving Detectors

Monochromatic Material Decomposition

Joint Bilateral Filter

Summary

- Take Home Messages
- Further Readings

Monochromatic Material Decomposition (Firsching et al., 2008)

Assumption: Bins I_b effectively contain a single energy measurement b.

- The material dependent absorption at energy b is $\mu(b, j)$ for a material listed by index j.

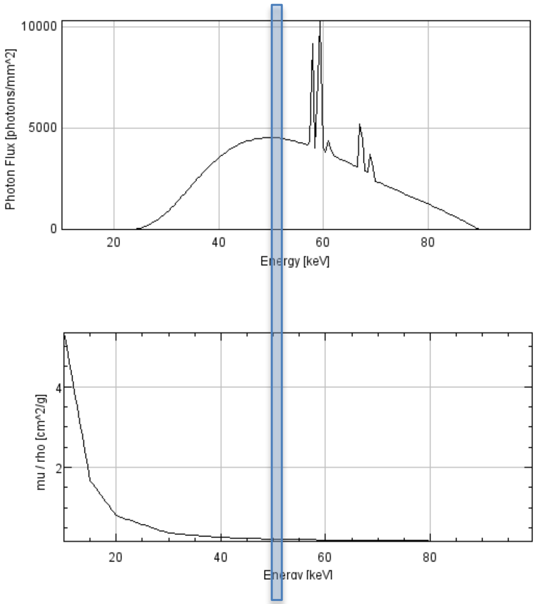


Figure 7: A single energy for a certain bin (top) and its according absorption coefficient (bottom)

Monochromatic Material Decomposition (Firsching et al., 2008)

Assumption: Bins I_b effectively contain a single energy measurement b.

- The material dependent absorption at energy b is $\mu(b,j)$ for a material listed by index j.
- A measured value is equal to the integral over all materials j:

$$I_{0b} e^{-\sum_j \mu(b,j) l_j} = I_b,$$

$$\sum_j \mu(b,j) l_j = -\ln \frac{I_b}{I_{0b}},$$

$$\sum_j \mu(b,j) l_j = q_b.$$

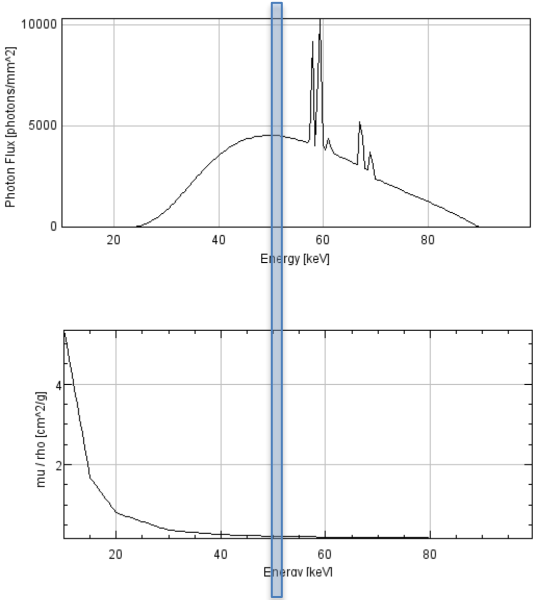


Figure 7: A single energy for a certain bin (top) and its according absorption coefficient (bottom)

Monochromatic Material Decomposition

- There is only a certain number N of materials:

$$\sum_{j=1}^{N} \mu(b,j) l_j = q_b \,.$$

- Formulation in matrix notation for N materials:

$$\begin{pmatrix} \mu(b,1) \\ \vdots \\ \mu(b,N) \end{pmatrix}^T \begin{pmatrix} l_1 \\ \vdots \\ l_N \end{pmatrix} = q_b,$$

$$\boldsymbol{\mu}_b^T \boldsymbol{l} = q_b.$$

Monochromatic Material Decomposition

- Formulation in matrix notation for N materials:

$$\boldsymbol{\mu}_b^T \boldsymbol{l} = q_b$$

$$\begin{pmatrix} \boldsymbol{\mu}_1^T \\ \vdots \\ \boldsymbol{\mu}_M^T \end{pmatrix} \boldsymbol{l} = \begin{pmatrix} q_1 \\ \vdots \\ q_M \end{pmatrix}$$

$$\boldsymbol{M}\boldsymbol{l} = \boldsymbol{q}$$

- Solution using pseudoinverse:

$$\boldsymbol{l} = \boldsymbol{M}^+ \boldsymbol{q}$$

Monochromatic Material Decomposition

- To determine the decomposition of N materials, at least N bins are required.
  - → At least as many bins as number of materials are required.
- A decomposition is only possible if the rank of $\boldsymbol{M}$ is greater or equal to N.
  - → Basic materials must not be linearly dependent.

Topics

X-ray Projections From Energy-Resolving Detectors

Monochromatic Material Decomposition

Joint Bilateral Filter

Summary

- Take Home Messages
- Further Readings

Denoising with Joint Bilateral Filter (Lu et al., 2015)

- Sum bins to create guidance image:

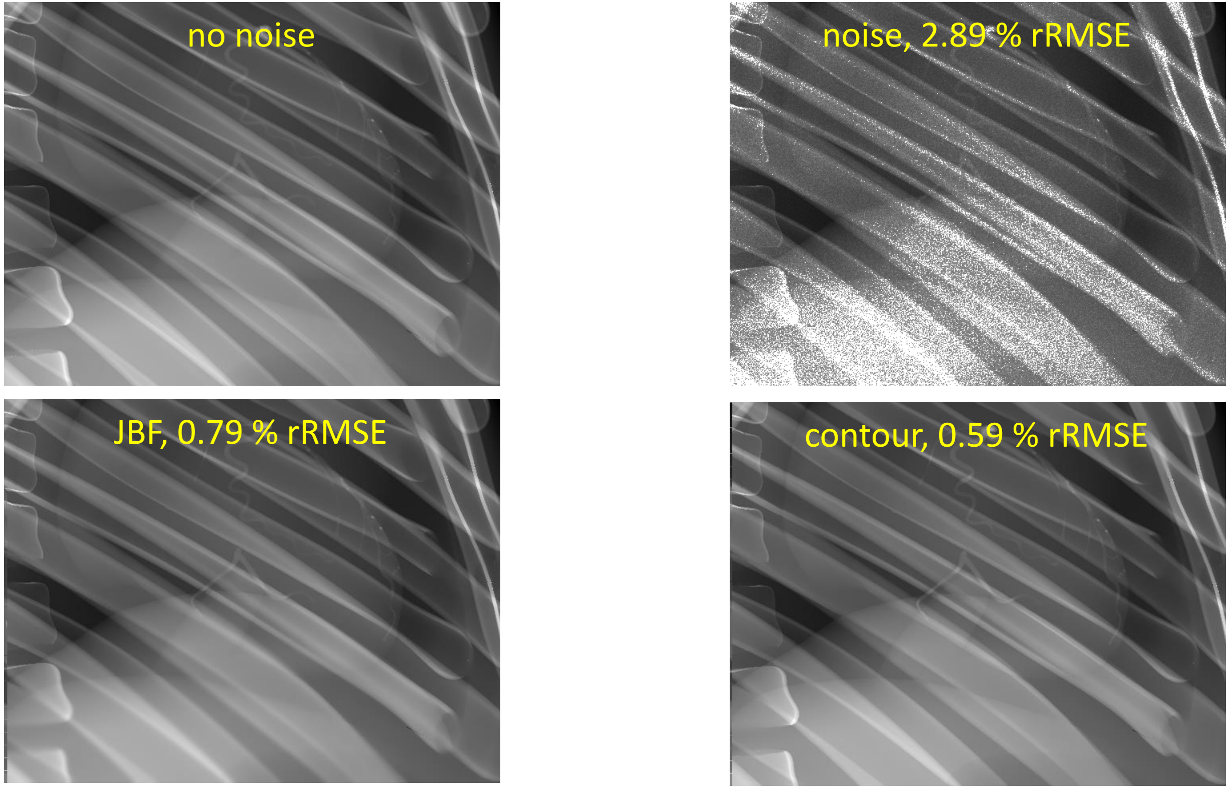
$$I(x,y) = \sum_b I_b(x,y)$$

- Use bilateral filter:
  - Spatial kernel → Gaussian $G_s(\boldsymbol{x},\boldsymbol{x}')$ with σ_s chosen like in normal bilateral filter
  - Range kernel → Gaussian $G_I(\boldsymbol{x},\boldsymbol{x}')$ with σ_I determined using the guidance image I
- Contour-aware filtering (angiography):

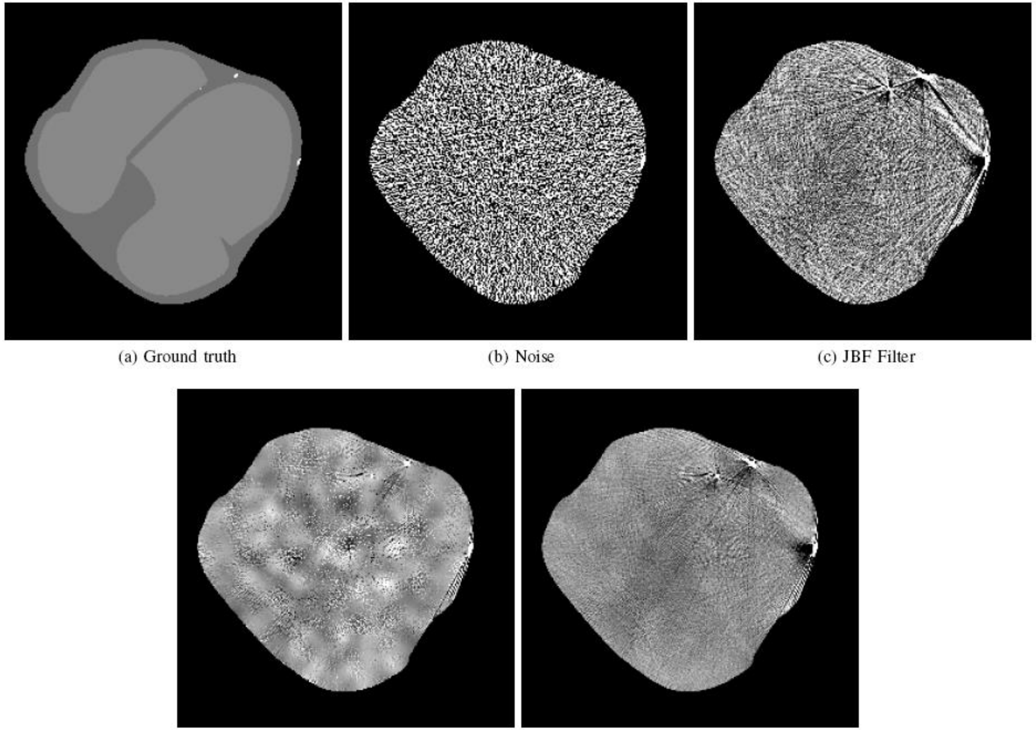
$$\sigma_I(x,y) = \bar{I}(x,y)z, \qquad z = \left(1 - \frac{I_2}{I_1}\right)$$

  - $\bar{I}$ is the guidance image filtered with an average filter.
  - z is determined such that a certain contrast difference $D = I_1 - I_2$ is preserved.

Results



Reconstructions (Manhart et al., 2014)



(a) Ground truth (b) Noise (c) JBF Filter

(d) Noise + ID-JBF (e) JBF + ID-JBF

Topics

X-ray Projections From Energy-Resolving Detectors

Monochromatic Material Decomposition

Joint Bilateral Filter

Summary

- Take Home Messages
- Further Readings

Take Home Messages

- With energy-resolving detectors projections are acquired for different parts of the radiation energy spectrum.
- This information is binned and can be used to decompose the volume into different materials.
- Denoising is an application of a joint bilateral filter which makes use of energy-binned projection data.

Further Readings

- Markus Firsching et al. "Material Resolving X-ray Imaging Using Spectrum Reconstruction with Medipix2". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 591.1 (June 2008), pp. 19–23. DOI: 10.1016/j.nima.2008.03.017
- Yanye Lu et al. "Projection-Based Denoising Method for Photon-Counting Energy-Resolving Detectors". In: *Bildverarbeitung für die Medizin 2015: Algorithmen - Systeme - Anwendungen. Proceedings des Workshops vom 15. bis 17. März 2015 in Lübeck*. Ed. by Heinz Handels et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 137–142. DOI: 10.1007/978-3-662-46224-9_25
- Michael Manhart et al. "Guided Noise Reduction for Spectral CT with Energy-Selective Photon Counting Detectors". In: *Proceedings of the Third CT Meeting*. Ed. by Frédéric Noo. Salt Lake City, UT, USA, June 2014, pp. 91–94

Medical Image Processing for Interventional Applications

Learning of Material Decomposition

Online Course – Unit 19
Andreas Maier, Frank Schebesch
Pattern Recognition Lab (CS 5)

virtuelle hochschule bayern
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Topics

Polychromatic Material Decomposition

Material Decomposition with Pattern Recognition

Summary

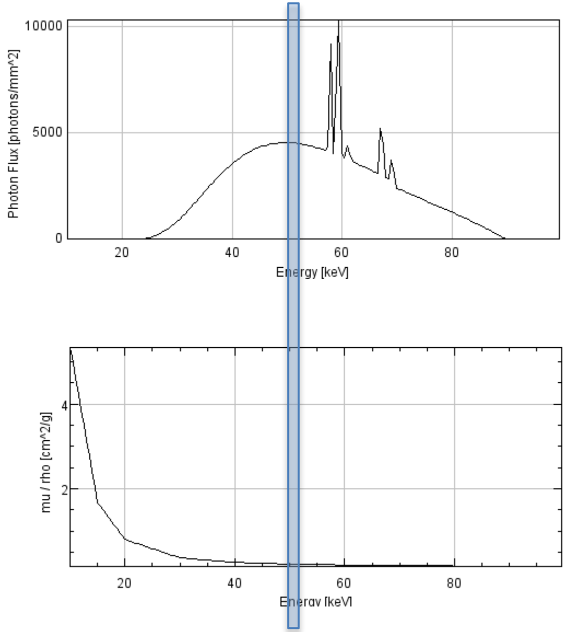
Take Home Messages

Further Readings

Polychromatic Material Decomposition (Maaß et al., 2011)

So far we have not considered the polychromatic nature of X-ray radiation:

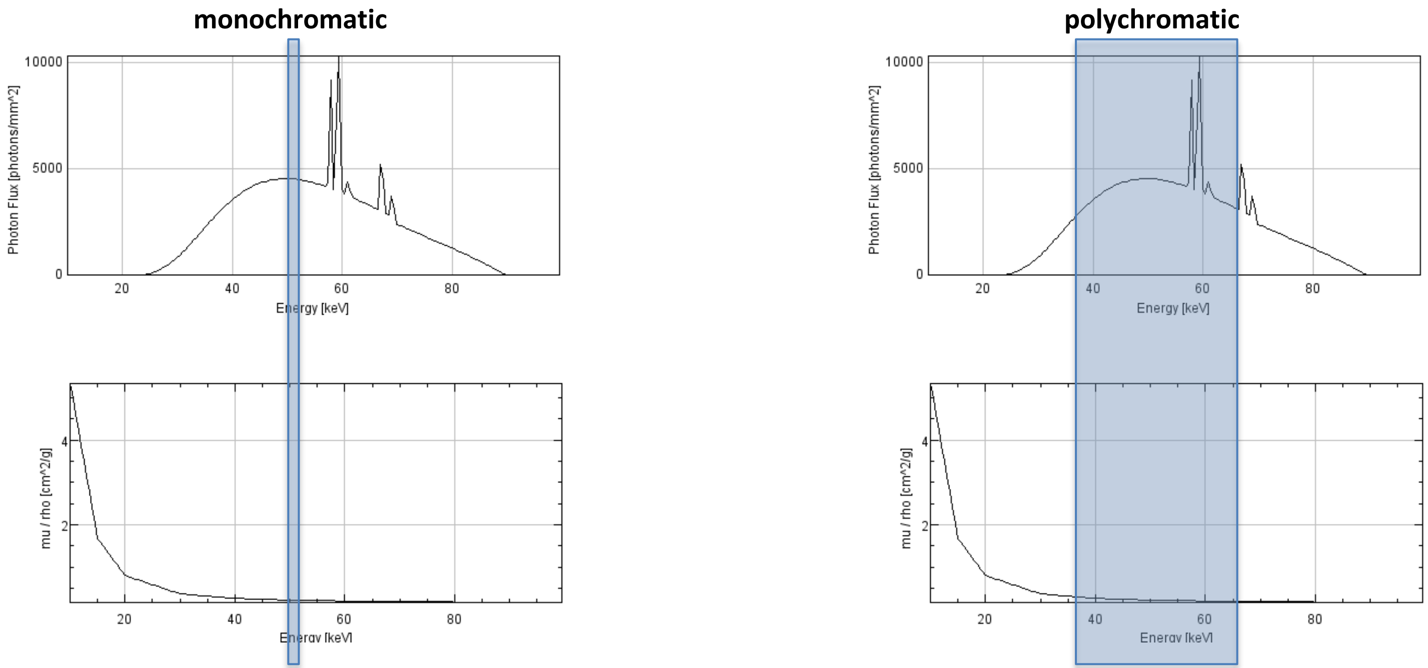
monochromatic



$$I_{0b} e^{-\int \mu(b,j) l_j \mathrm{d}j} = I_b$$

Polychromatic Material Decomposition (Maaß et al., 2011)

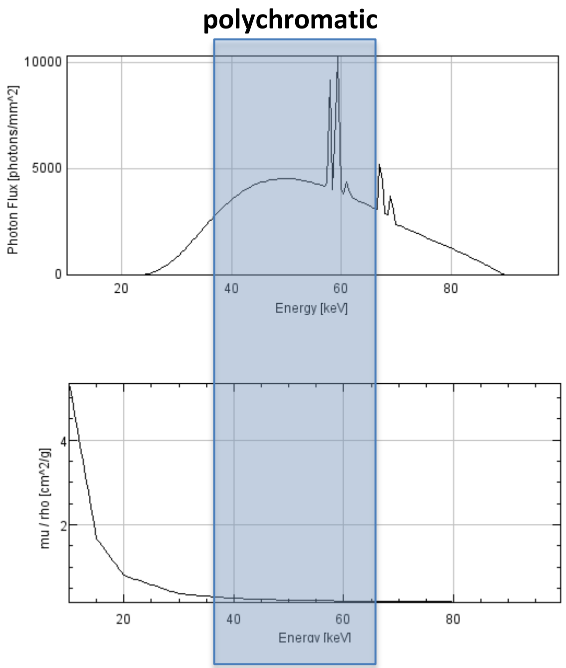
So far we have not considered the polychromatic nature of X-ray radiation:



Polychromatic Material Decomposition (Maaß et al., 2011)

So far we have not considered the polychromatic nature of X-ray radiation:

$$\int I_{0b'} e^{-\int \mu(b',j) l_j \mathrm{d}j} \mathrm{d}b' = I_b$$



Polychromatic Material Decomposition (Maaß et al., 2011)

The polychromatic absorption outcome in bin b is given by:

$$\int I_{0b'} e^{-\int \mu(b',j) l_j \mathrm{d}j} \mathrm{d}b' = I_b.$$

The inversion of this problem is quite difficult.

→ Inversion can be approximated by a polynomial of degree K:

$$l(q_1, q_2, \dots, q_B) = \sum_{k_1, k_2, \dots, k_B = 0}^{K-1} c_{k_1, k_2, \dots, k_B} q_1^{k_1} q_2^{k_2} \dots q_B^{k_B}.$$

Topics

Polychromatic Material Decomposition

Material Decomposition with Pattern Recognition

Summary

Take Home Messages

Further Readings

Material Decomposition with Pattern Recognition (Lu et al., 2015)

- Inversion can be formulated as a general problem of function estimation:

$$l(\boldsymbol{q}) = f_{ML}(\boldsymbol{q}).$$

- Typical models from machine learning can be used:
  - random forests,
  - multilayer perceptrons (MLP),
  - support vector machines.

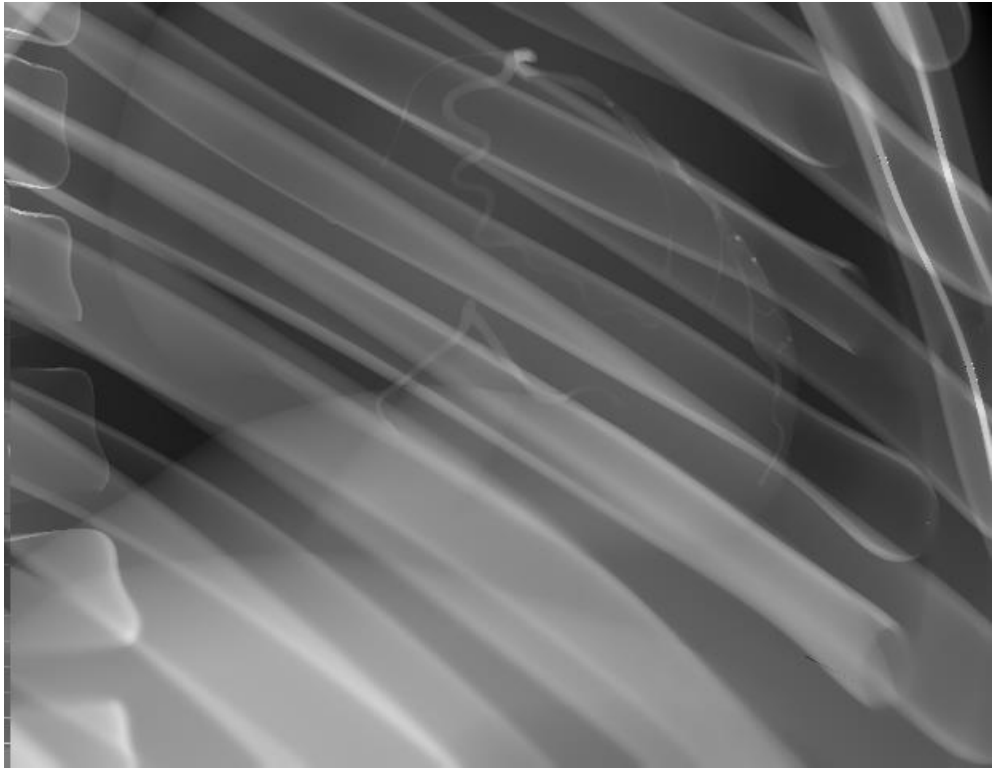
Material Decomposition with Pattern Recognition (Lu et al., 2015)

Utilization of additional information becomes possible:

- Structure characteristics can be used to describe shapes.
- Material images should build a basis and should not be correlated.
- Material images must fulfill certain consistency conditions.

→ A decomposition of more material types than bins becomes possible.

Input Image (Bin 1)



Visualization Contrast Agent

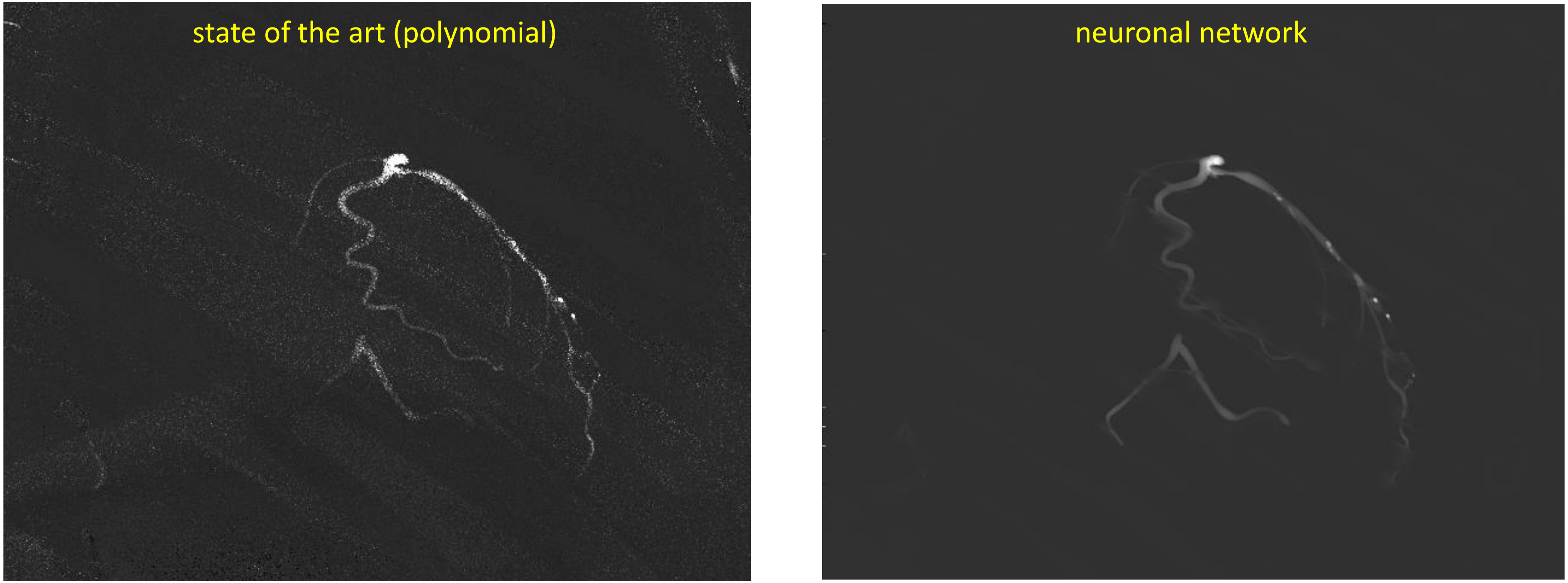
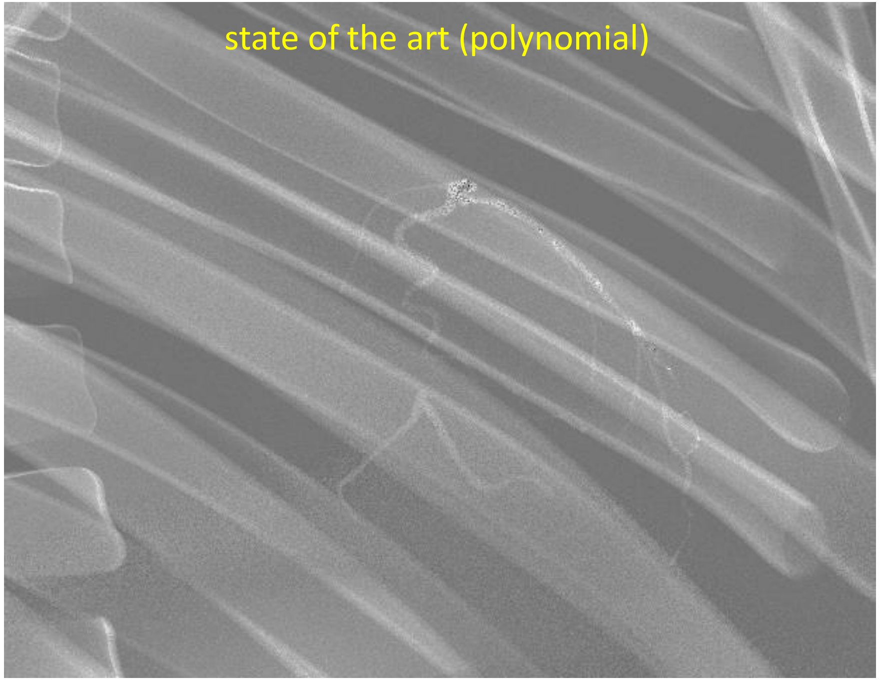


Figure 4: Polynomial fit (left), multilayer perceptron (MLP) (right)

Visualization Bone



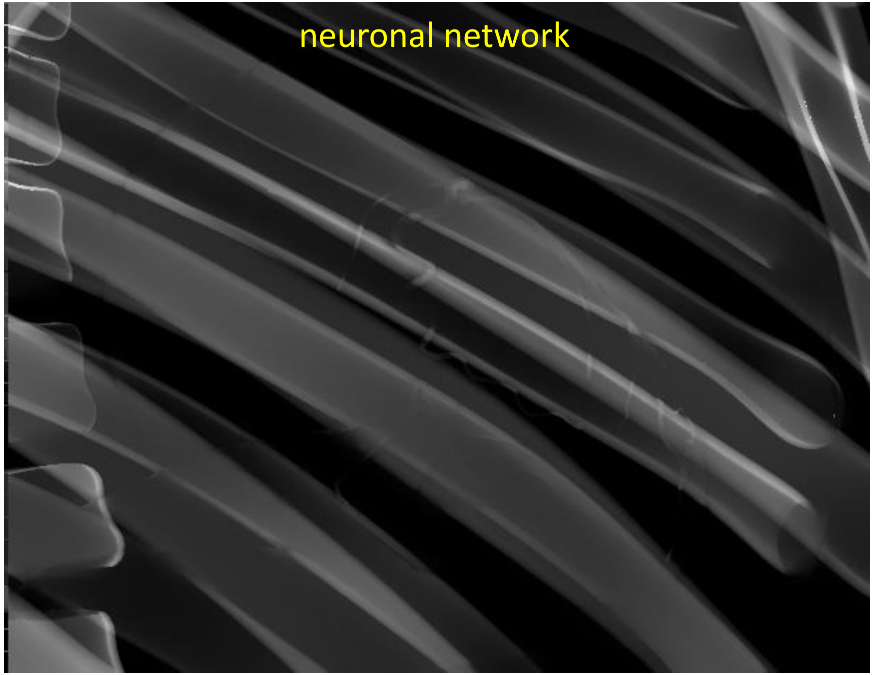


Figure 5: Polynomial fit (left), multilayer perceptron (MLP) (right)

Original Acquisitions

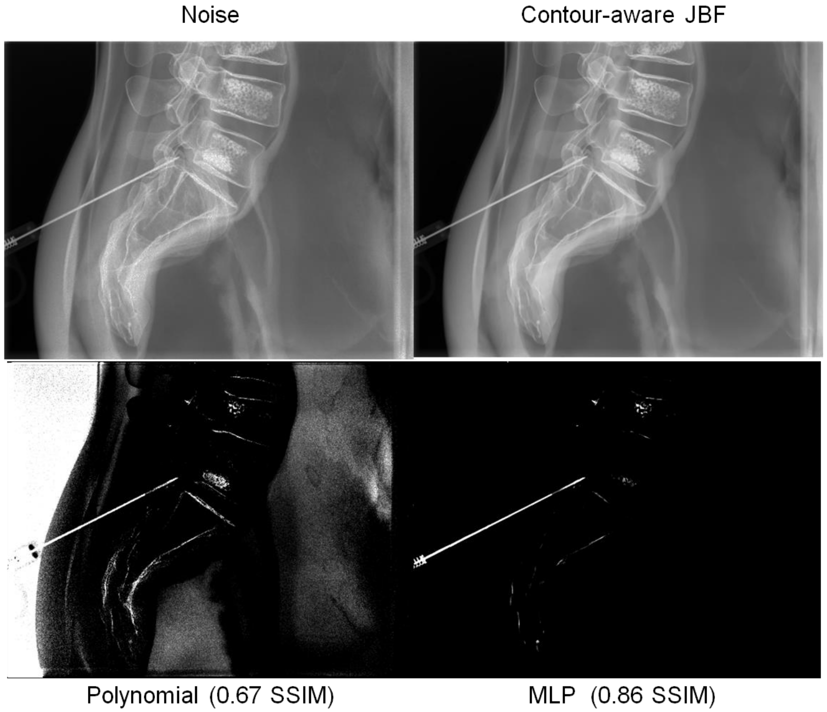


Figure 6: These images were acquired in cooperation with Stanford University.

Topics

Polychromatic Material Decomposition

Material Decomposition with Pattern Recognition

Summary

Take Home Messages

Further Readings

Take Home Messages

- The polychromatic model to describe the physical phenomenon of X-ray attenuation for a specific radiation emission system is even closer to physics than the monochromatic model.
- Respectively, material decomposition makes use of either of these models. The monochromatic case can be described by a linear system of equations, while we used methods from machine learning for the polychromatic case.

Further Readings

- Markus Firsching et al. “Material Resolving X-ray Imaging Using Spectrum Reconstruction with Medipix2”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 591.1 (June 2008), pp. 19–23. DOI: 10.1016/j.nima.2008.03.017
- Nicole Maaß et al. “Empirical Multiple Energy Calibration (EMEC) for Material-Selective CT”. . In: *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE*. IEEE, Oct. 2011, pp. 4222–4229. DOI: 10.1109/NSSMIC.2011.6153810
- Yanye Lu et al. “Projection-based Material Decomposition by Machine Learning using Image-based Features for Computed Tomography”. In: *The 13th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. Newport, Rhode Island, USA, 2015, pp. 448–451