Part 4 -
Display-Aware Visualization and Processing

MOTIVATION


goal:
perform computations
at output resolution
250 megapixels
<1 megapixel visible


resolution level 0


resolution level 3

SIGGRAPH ASIA 2018 TOKYO

DISPLAY-AWARE IMAGE OPERATIONS

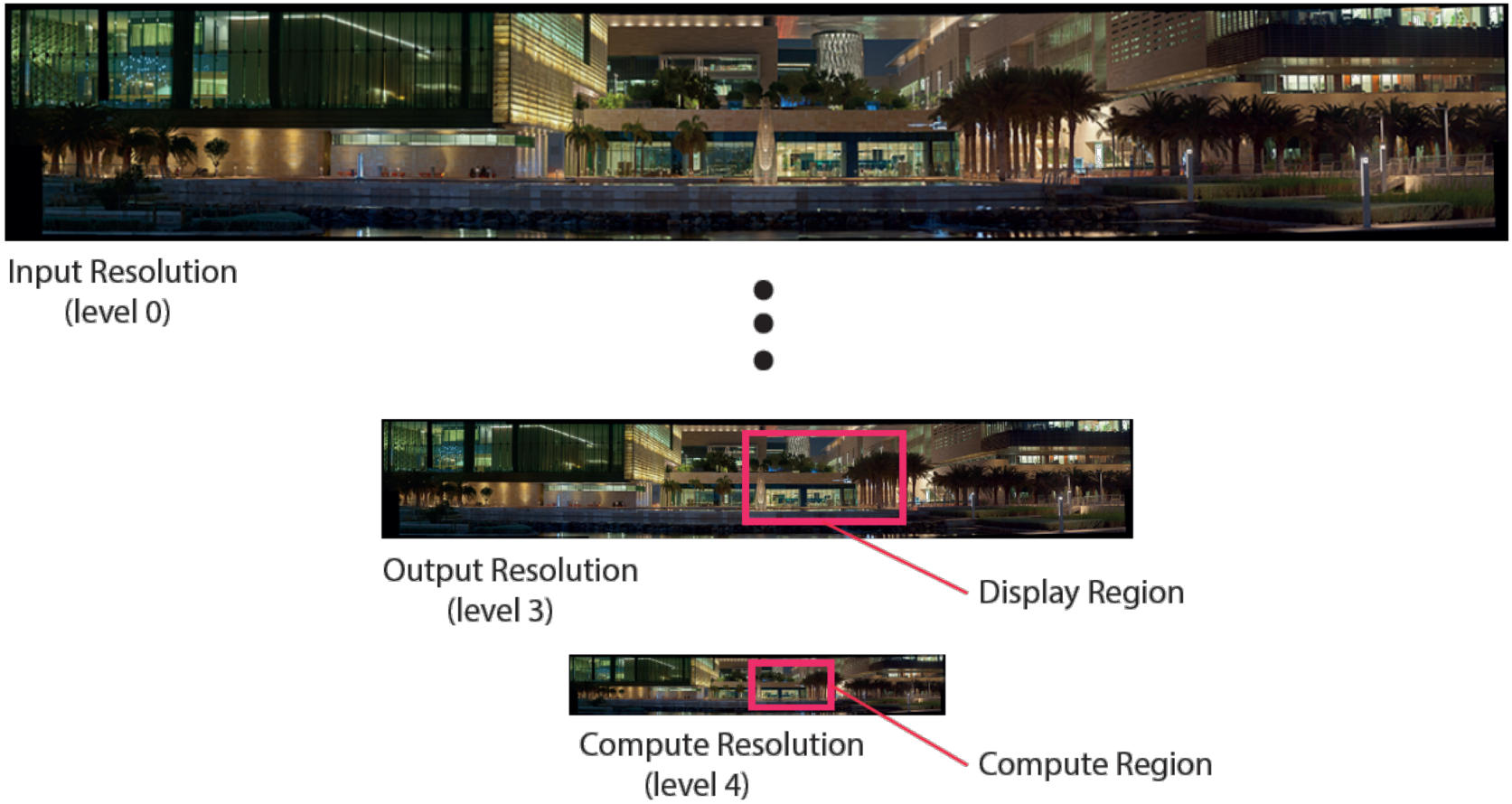


IMAGE PYRAMIDS

- Dyadic image pyramids
  - Mipmaps [Williams 1983]: texture mapping (standard on GPUs)
  - Gaussian/Laplacian pyramids [Burt and Adelson 1983]: image processing/compression

level 0

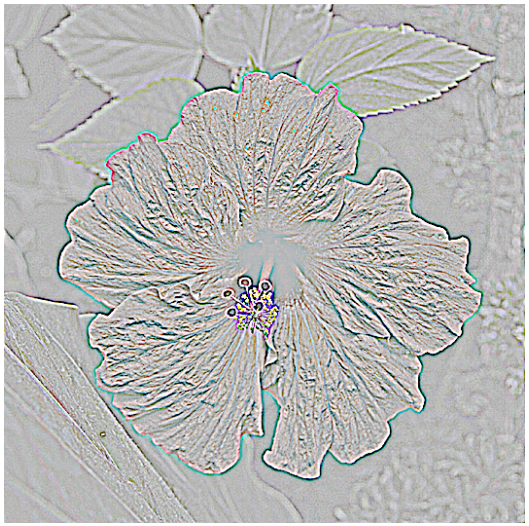
level 1

level 2

level 3

IMAGE PYRAMIDS

- Dyadic image pyramids
  - Mipmaps [Williams 1983]: texture mapping (standard on GPUs)
  - Gaussian/Laplacian pyramids [Burt and Adelson 1983]: image processing/compression



level 0

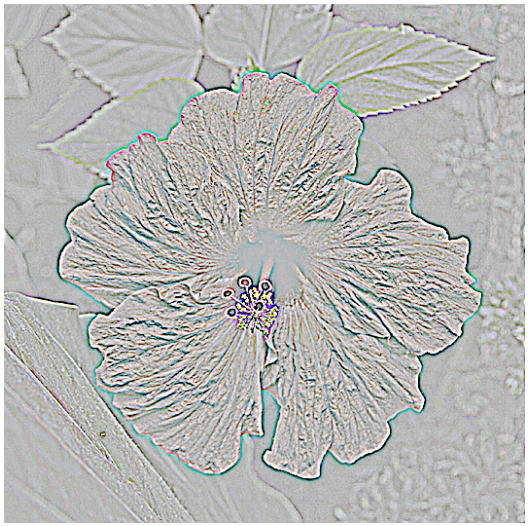
level 1

level 2

level 3

IMAGE PYRAMIDS

- Dyadic image pyramids
  - Mipmaps [Williams 1983]: texture mapping (standard on GPUs)
  - Gaussian/Laplacian pyramids [Burt and Adelson 1983]: image processing/compression



level 0

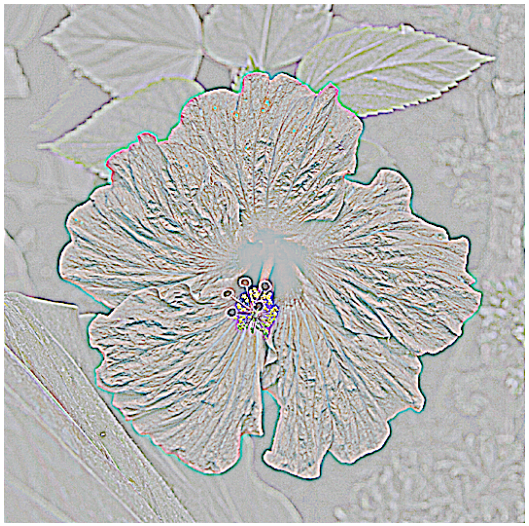
level 1

level 2

level 3

IMAGE PYRAMIDS

- Dyadic image pyramids
  - Mipmaps [Williams 1983]: texture mapping (standard on GPUs)
  - Gaussian/Laplacian pyramids [Burt and Adelson 1983]: image processing/compression
  - Sparse pdf maps [Hadwiger et al. 2012]



level 0

Laplacian pyramid

level 1

level 2

level 3

IMAGE PYRAMIDS

- Dyadic image pyramids
  - Mipmaps [Williams 1983]: texture mapping (standard on GPUs)
  - Gaussian/Laplacian pyramids [Burt and Adelson 1983]: image processing/compression
  - Sparse pdf maps [Hadwiger et al. 2012]

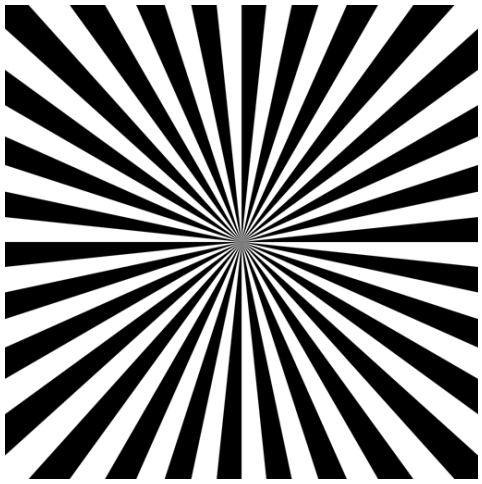
level 0

level 1

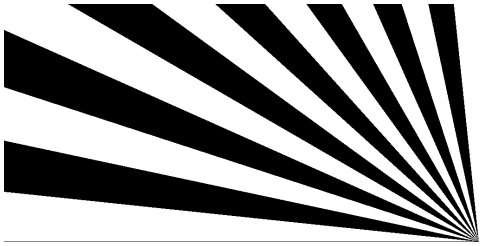
level 2

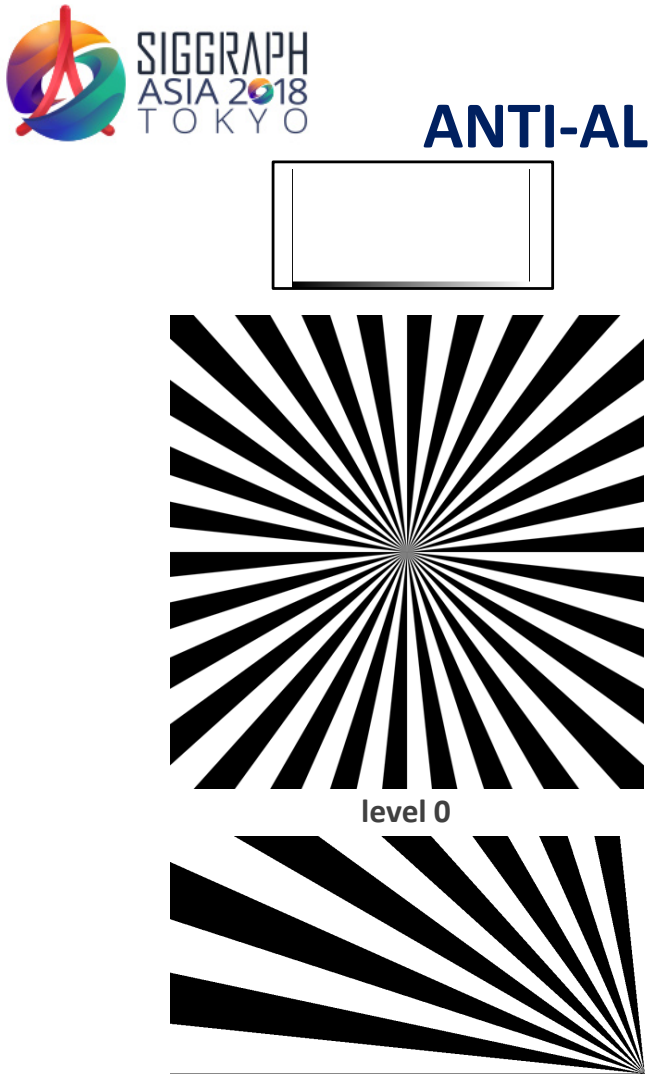
level 3

ANTI-ALIASING IN IMAGE PYRAMIDS



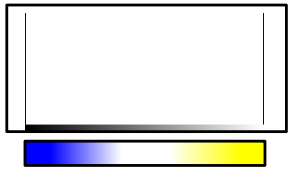
level 0

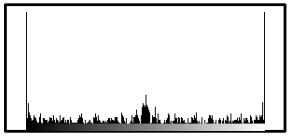


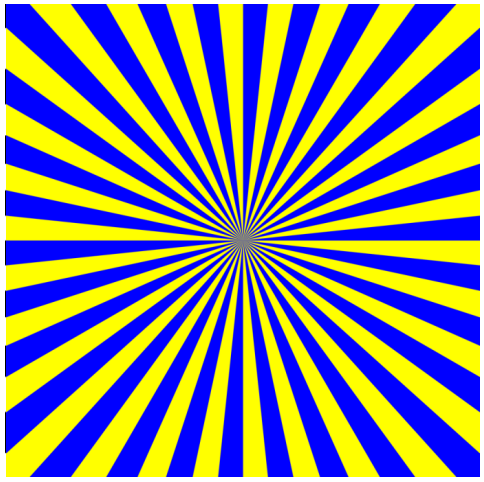


ANTI-ALIASING IN IMAGE PYRAMIDS

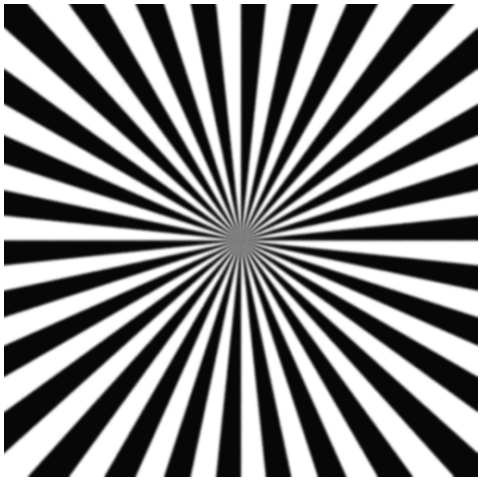
ANTI-ALIASING IN IMAGE PYRAMIDS



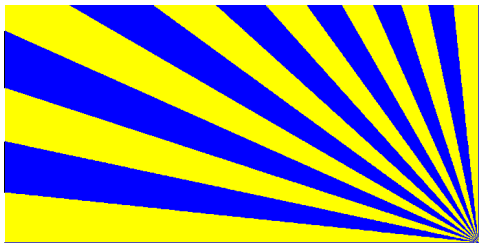




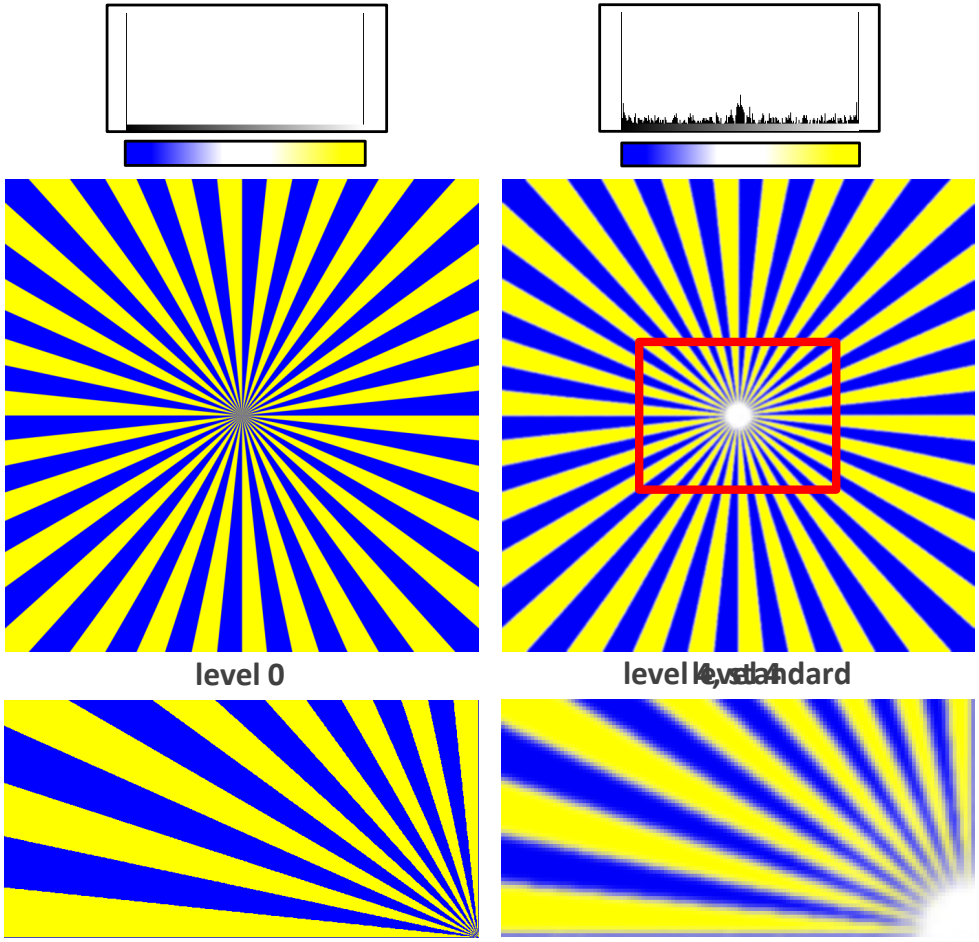
level 0



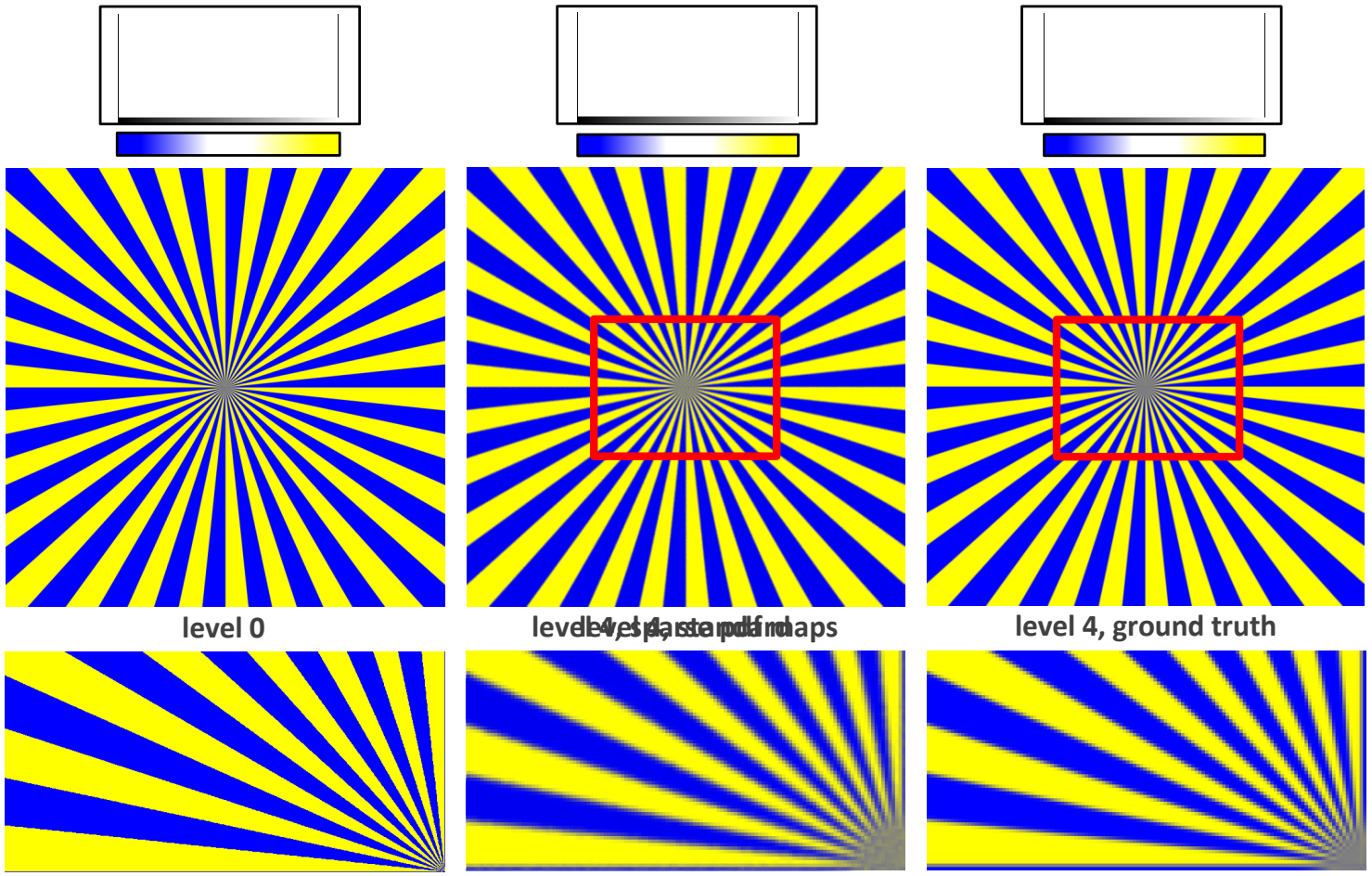
level 4



ANTI-ALIASING IN IMAGE PYRAMIDS



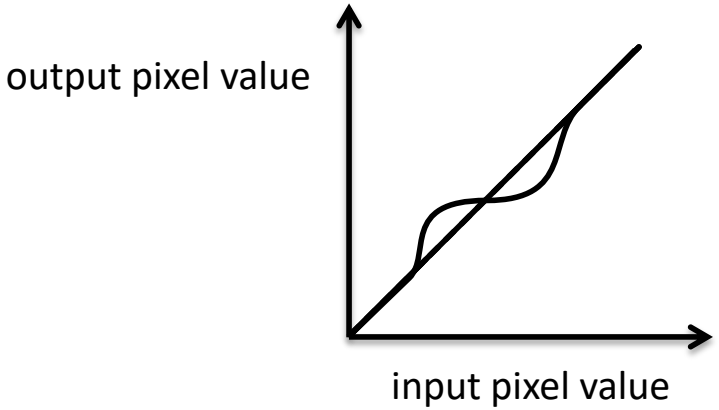
ANTI-ALIASING IN IMAGE PYRAMIDS



NON-LINEAR IMAGE OPERATORS

Apply non-linear operation to each pixel

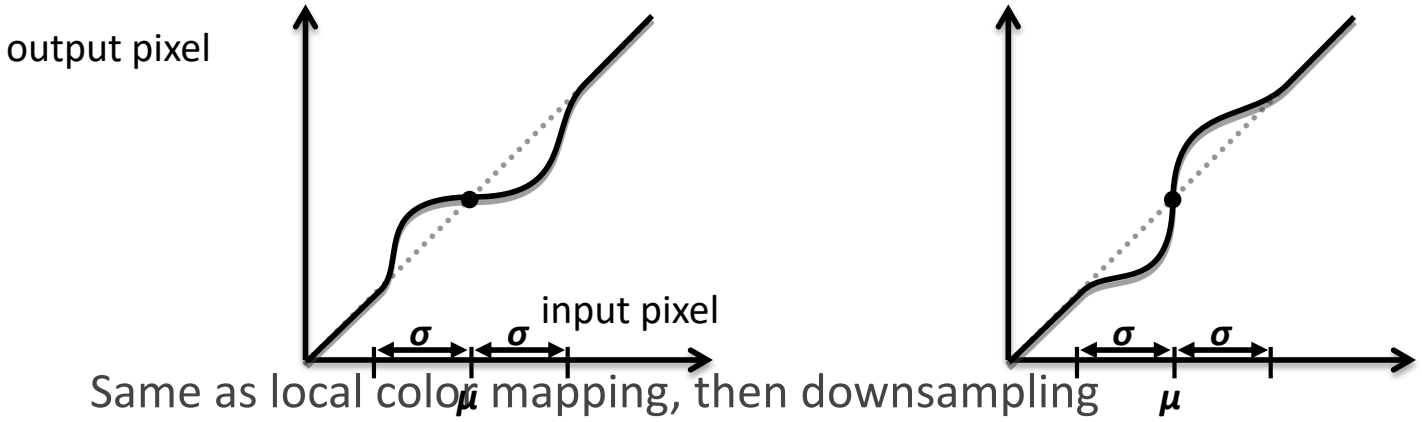
- Color map or non-linear contrast adjustment
- Bilateral filtering: range weight
- Smoothed local histogram filtering [Kass and Solomon 2010]
- Local Laplacian filtering [Paris et al. 2011]: point-wise, non-linear re-mapping



LOCAL LAPLACIAN FILTERING [PARIS ET AL. 2011]

Compute Laplacian pyramid coefficient

- Adjust local contrast via point-wise non-linearity; then downsample

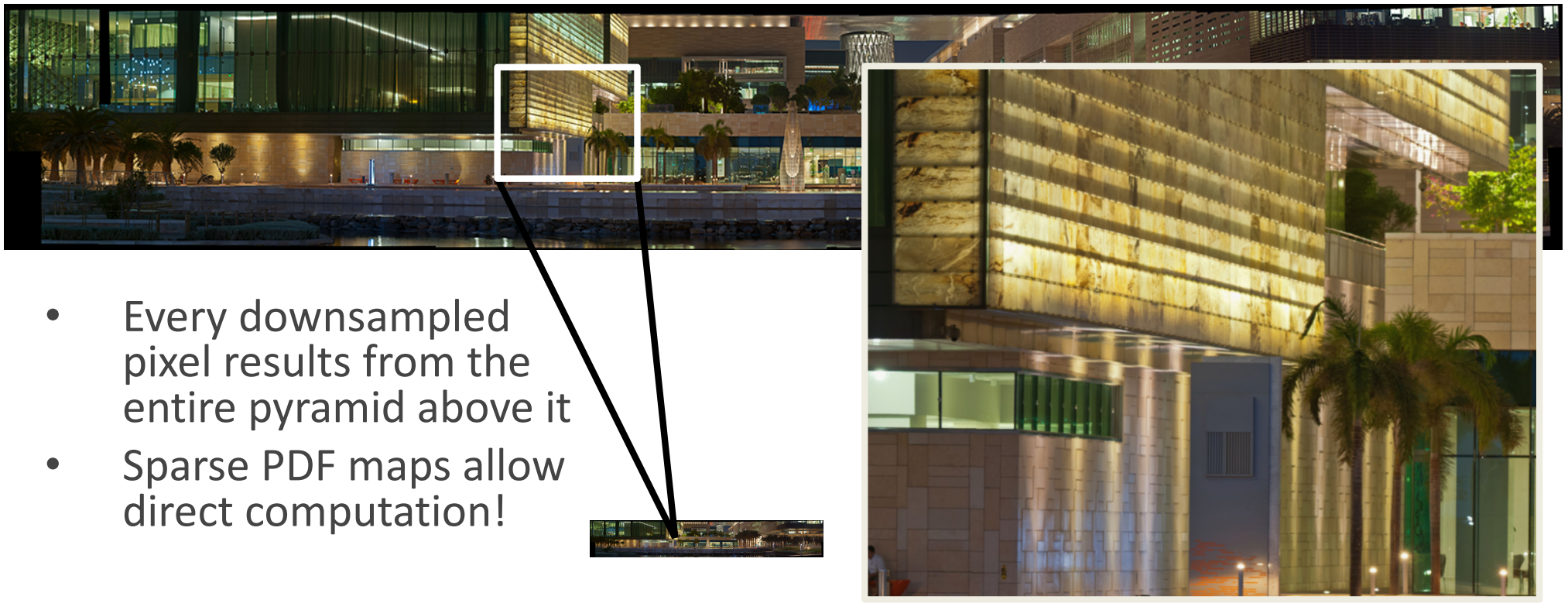


Same as local color mapping, then downsampling

- Cannot apply the re-mapping function to the downsampled image!
- Need to compute ground truth (pyramid!) or proper “anti-aliasing”

LOCAL LAPLACIAN FILTERING: SCALABILITY

- Night Scene Panorama: 47,908 x 7,531 pixels (361 Mpixels)

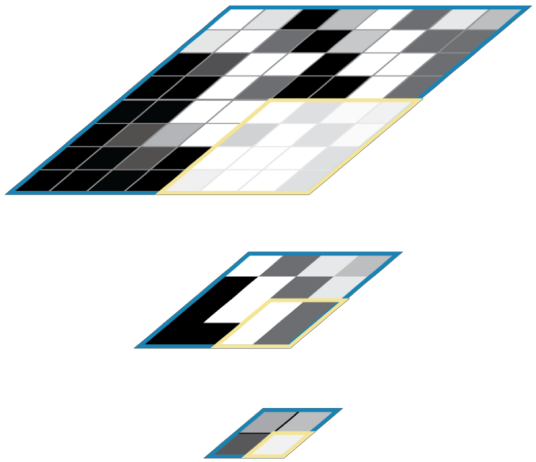


- Every downsampled pixel results from the entire pyramid above it
- Sparse PDF maps allow direct computation!

Sparse PDF Maps: Concept

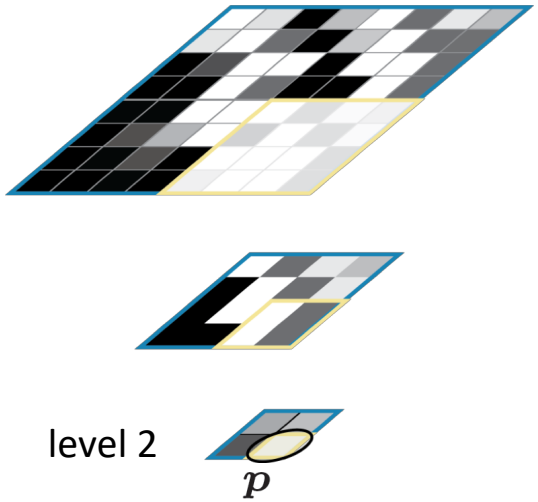
SPARSE PDF MAPS

Represent distribution of pixel values in footprint in original image



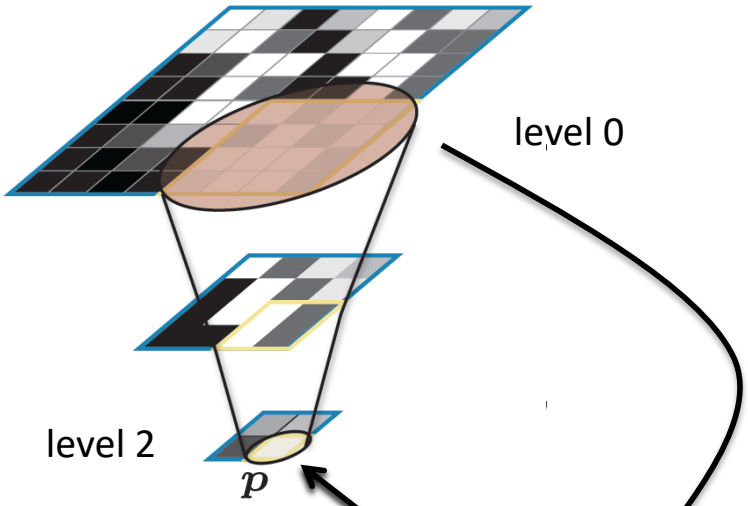
SPARSE PDF MAPS

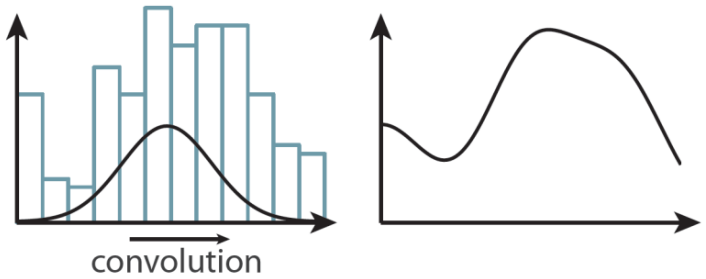
Represent distribution of pixel values in footprint in original image



SPARSE PDF MAPS

Represent distribution of pixel values in footprint in original image

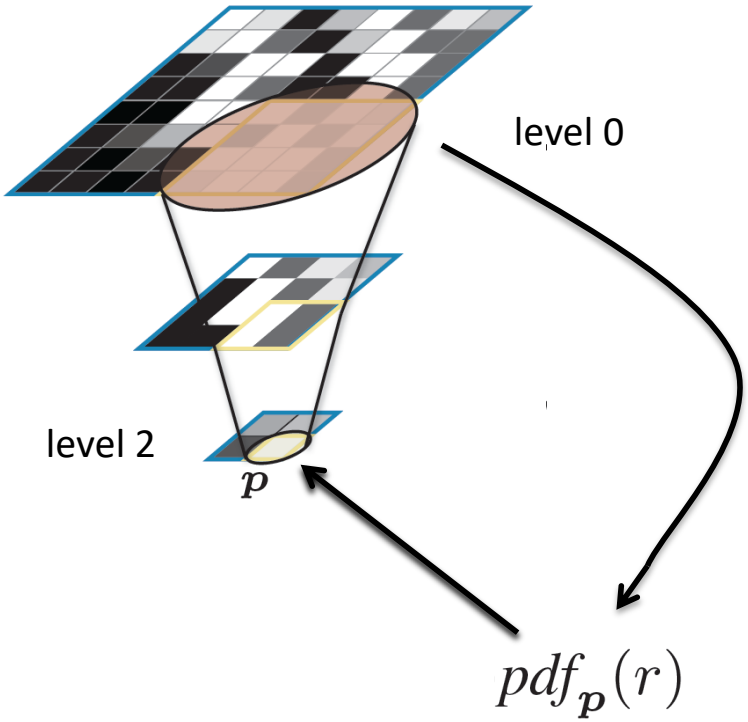


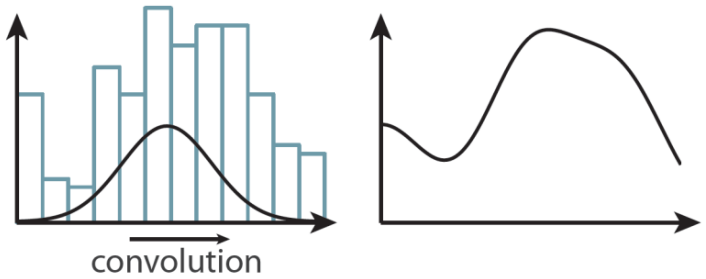


$$pdf_{\boldsymbol{p}}(r)$$

SPARSE PDF MAPS

Represent distribution of pixel values in footprint in original image





SPARSE PDF MAPS

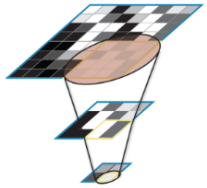
Represent distribution of pixel values in footprint in original image

Apply non-linear operation

$$E\left[t_{\boldsymbol{p}}\left(X_{\boldsymbol{p}}\right)\right] = \frac{1}{w_{\boldsymbol{p}}} \int_0^1 t_{\boldsymbol{p}}(r)\, pdf_{\boldsymbol{p}}(r)\, dr$$

EXAMPLE 1: DOWNSAMPLED IMAGE

$$E\left[t_{\boldsymbol{p}}\left(X_{\boldsymbol{p}}\right)\right] = \frac{1}{w_{\boldsymbol{p}}} \int_0^1 t_{\boldsymbol{p}}(r) pdf_{\boldsymbol{p}}(r)\, dr$$

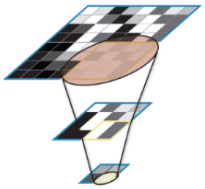


$$t_{\boldsymbol{p}}(r) = r$$

$$w_{\boldsymbol{p}} = 1$$

EXAMPLE 2: COLOR MAPPING

$$E\left[t_{\boldsymbol{p}}\left(X_{\boldsymbol{p}}\right)\right] = \frac{1}{w_{\boldsymbol{p}}} \int_0^1 t_{\boldsymbol{p}}(r) pdf_{\boldsymbol{p}}(r)\, dr$$

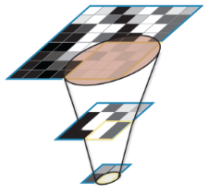


$$t_{\boldsymbol{p}}(r) = \text{color map}$$

$$w_{\boldsymbol{p}} = 1$$

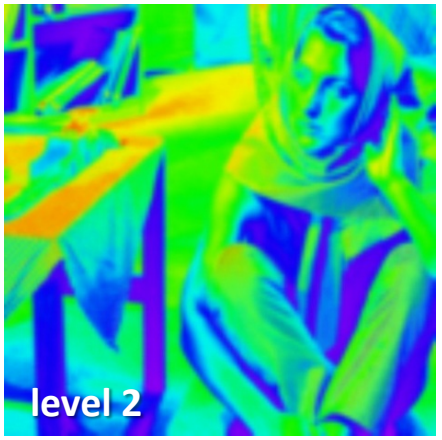
EXAMPLE 2: COLOR MAPPING

$$E\left[t_{\boldsymbol{p}}\left(X_{\boldsymbol{p}}\right)\right] = \frac{1}{w_{\boldsymbol{p}}} \int_0^1 t_{\boldsymbol{p}}(r) pdf_{\boldsymbol{p}}(r)\, dr$$



$$t_{\boldsymbol{p}}(r) = \text{color map}$$

$$w_{\boldsymbol{p}} = 1$$

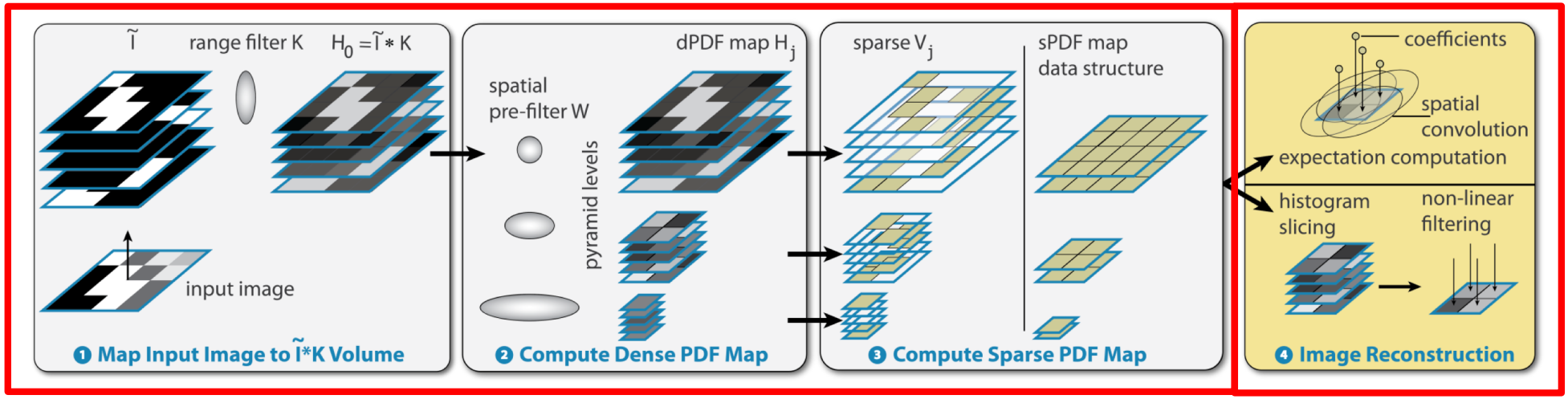


plus: bilateral filtering, local Laplacian filtering in linear time, …

INTERACTIVE GIGAPIXEL FILTERING

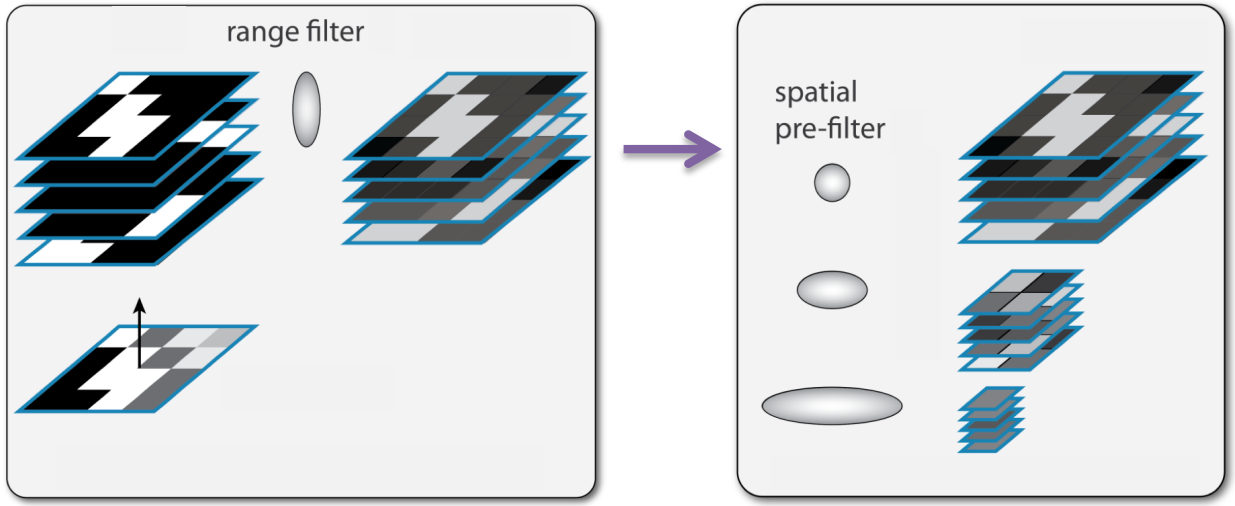
Sparse PDF Map Computation

PIPELINE


Ĩ
range filter K
$H_0 = \tilde{I} * K$
input image
1 Map Input Image to Ĩ*K Volume
dPDF map H_j
spatial
pre-filter W
pyramid levels
2 Compute Dense PDF Map
sparse V_j
sPDF map
data structure
3 Compute Sparse PDF Map
coefficients
spatial
convolution
expectation computation
histogram
slicing
non-linear
filtering
4 Image Reconstruction

STEP 1: DENSE PDF MAP

DENSE PDF MAP COMPUTATION

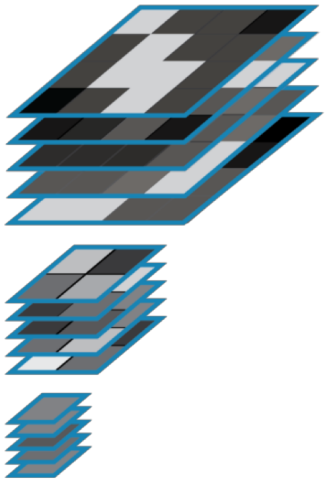


H is similar to a pyramid of bilateral grids [Chen et al. 2007]

STEP 2: SPARSE PDF MAP

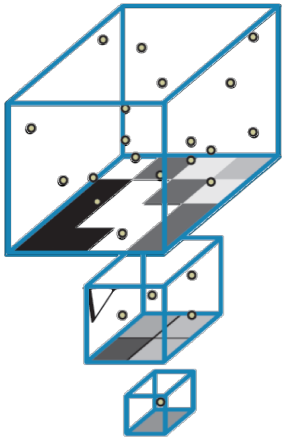
SIGGRAPH ASIA 2018 TOKYO

SPARSE REPRESENTATION

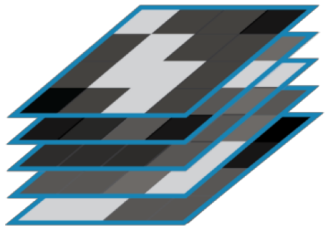


$$H \approx V * (W \otimes K)$$

$$G_{\boldsymbol{\sigma}_s} \otimes G_{\sigma_r}$$

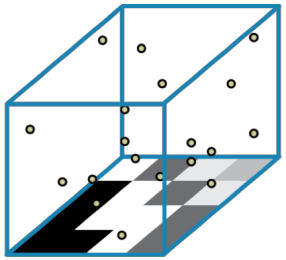


SPARSE REPRESENTATION



$$H \approx V * (W \otimes K)$$

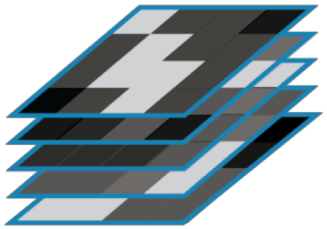
$$G_{\boldsymbol{\sigma}_s} \otimes G_{\sigma_r}$$



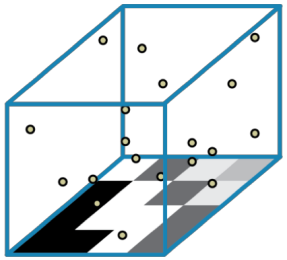
Compute via Matching Pursuit [Mallat and Zhang 1993]

SIGGRAPH ASIA 2018 TOKYO

SPARSE REPRESENTATION



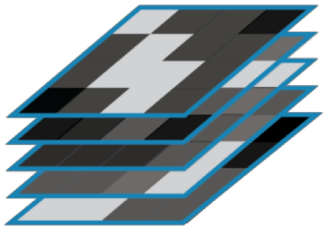
$$H \approx V * (W \otimes K)$$



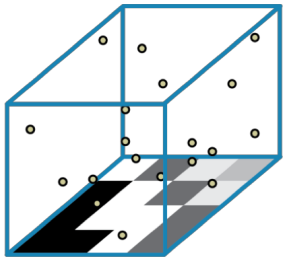
$$V(\boldsymbol{p}_n, r_n) = c_n \qquad c_n \neq 0$$

SIGGRAPH ASIA 2018 TOKYO

SPARSE REPRESENTATION



$$H \approx V * (W \otimes K)$$

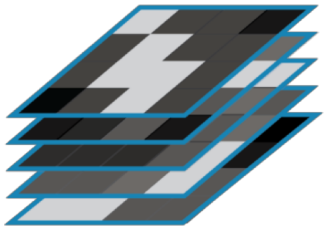


$$V(\boldsymbol{p}_n, r_n) = c_n \qquad c_n \neq 0$$

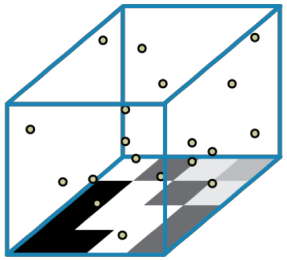
$$(\boldsymbol{p}_n, r_n, c_n)$$

SIGGRAPH ASIA 2018 TOKYO

SPARSE REPRESENTATION



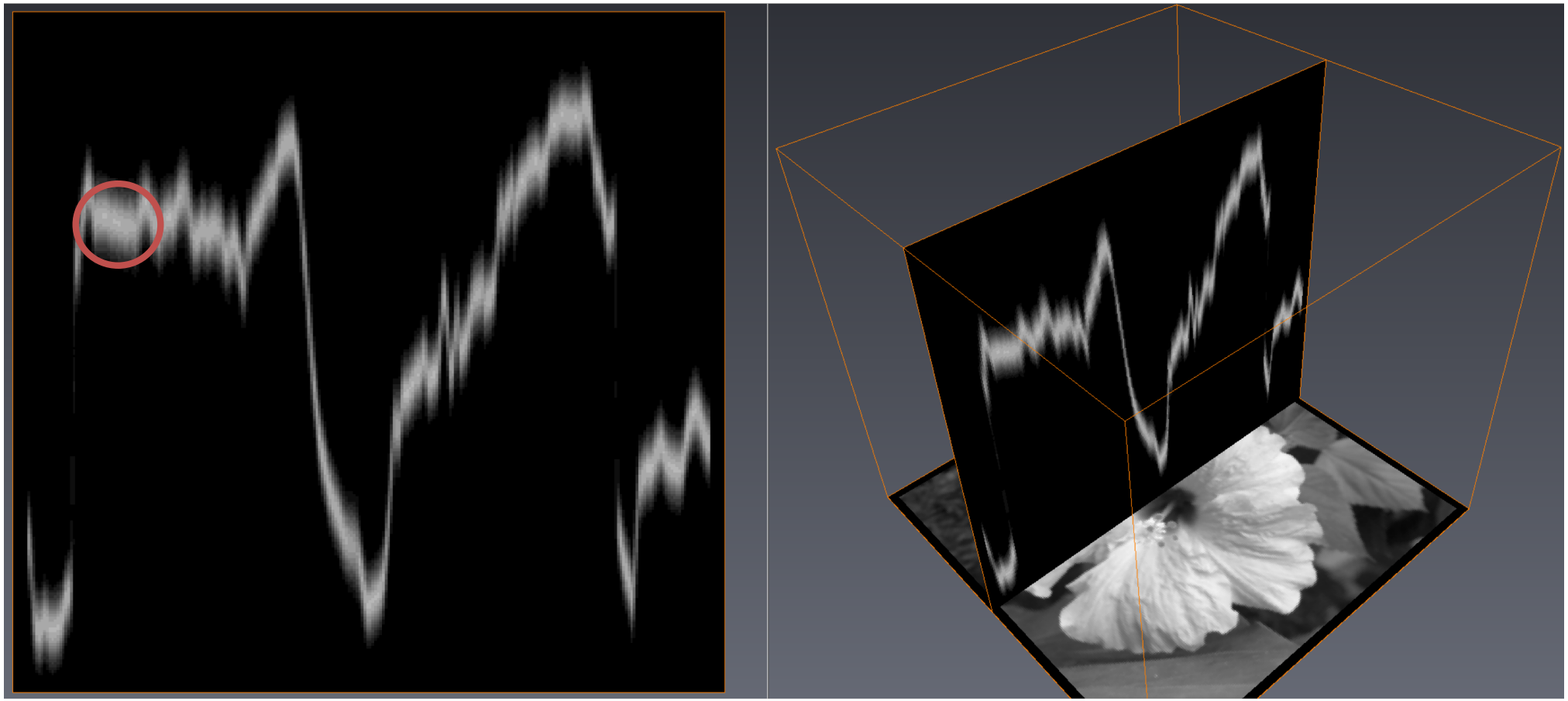
$$H \approx V * (W \otimes K)$$



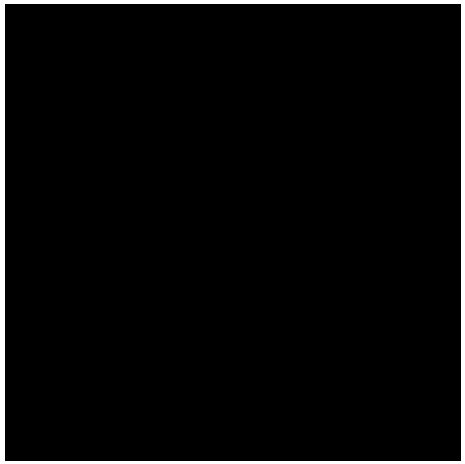
$$V(\boldsymbol{p}_n, r_n) = c_n \qquad c_n \neq 0$$

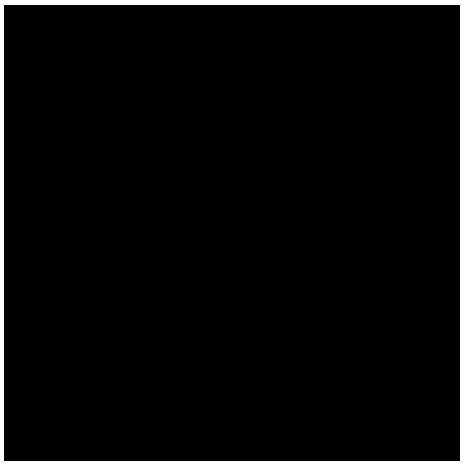
$$(\boldsymbol{p}_n, r_n, c_n)$$

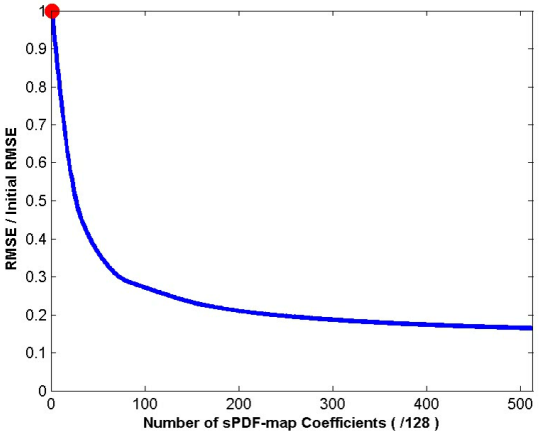
SPATIAL AND RANGE COHERENCE

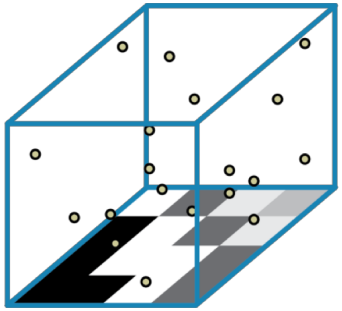


GREEDY APPROXIMATION







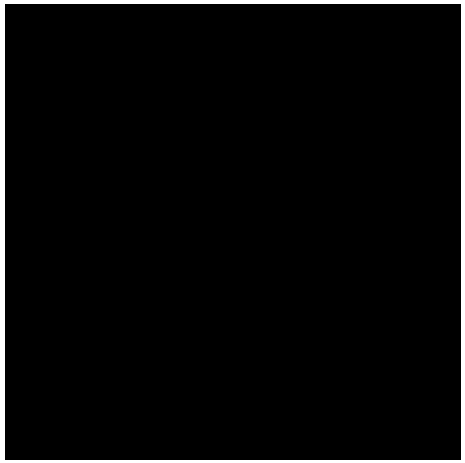


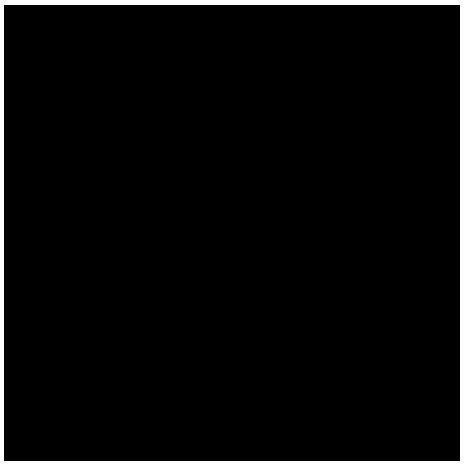
Spatial filter W : 5 x 5

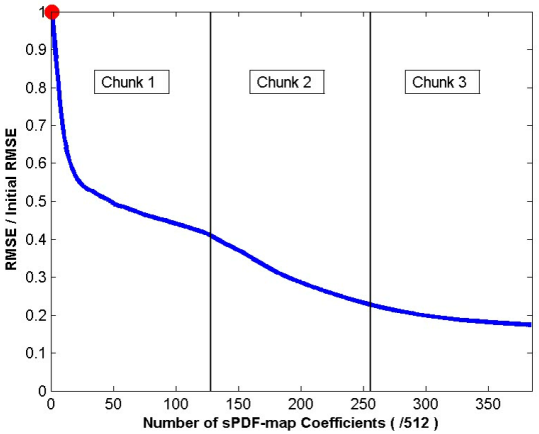
1 coefficient chunk

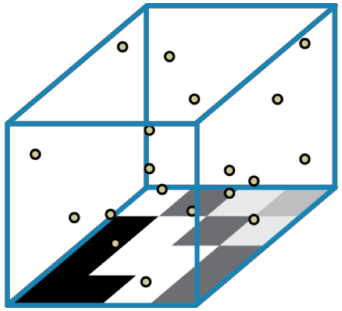
(# coefficients == 1 * # pixels)

GREEDY APPROXIMATION









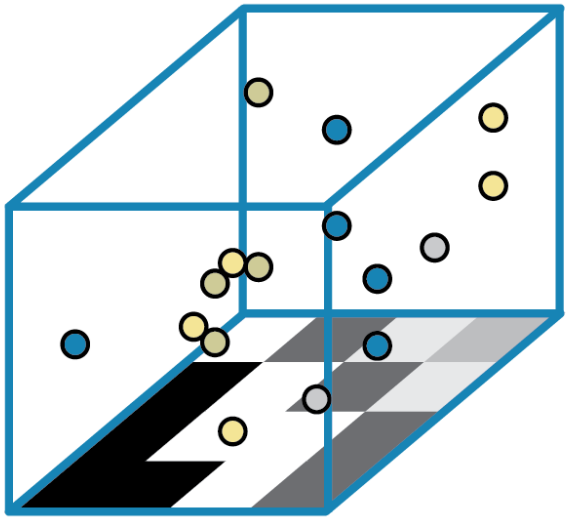
Spatial filter W : 3 x 3

1-3 coefficient chunks

(# coefficients == 1-3 * # pixels)

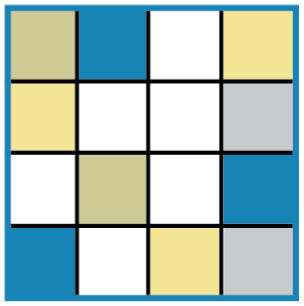
sPDF-Maps Data Structure

SPDF-MAPS DATA STRUCTURE



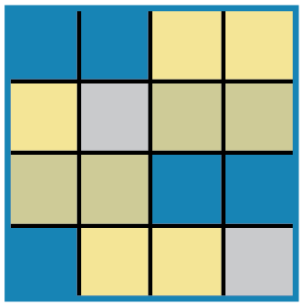
conceptual

$$V(\boldsymbol{p}_n, r_n) = c_n$$



index image

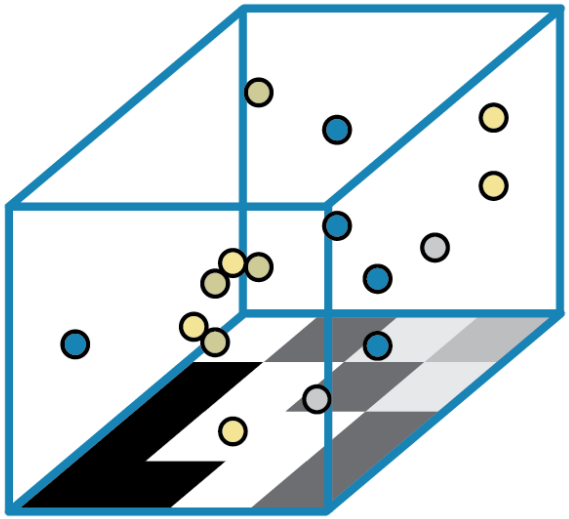
$$(index, count)_{\boldsymbol{p}}$$



coefficient image

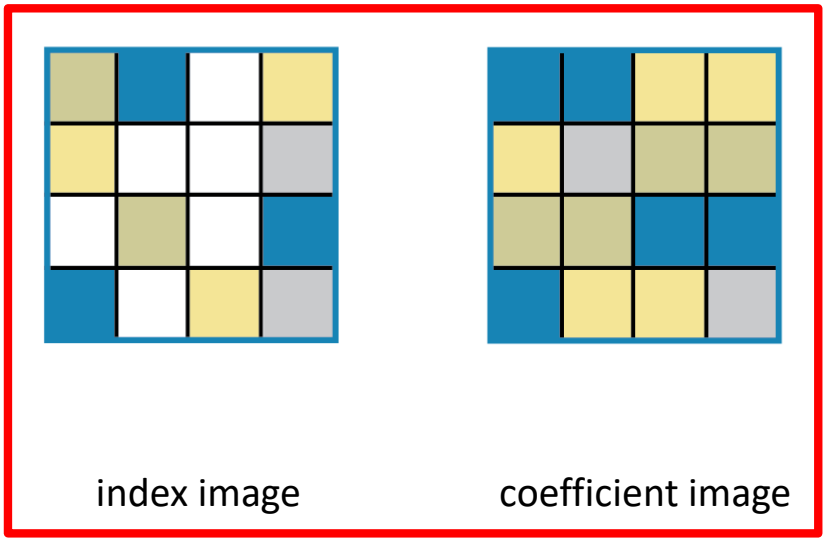
$$(r_n, c_n)$$

SPDF-MAPS DATA STRUCTURE



conceptual

$$V(\boldsymbol{p}_n, r_n) = c_n$$



index image

$$(index, count)_{\boldsymbol{p}}$$

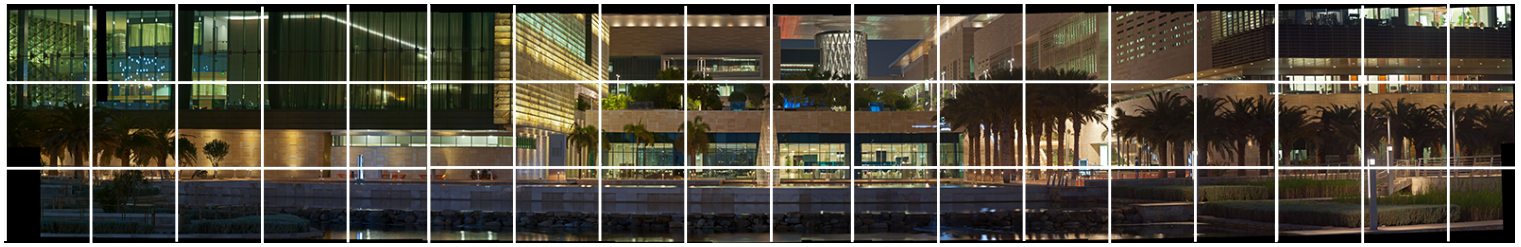
coefficient image

$$(r_n, c_n)$$

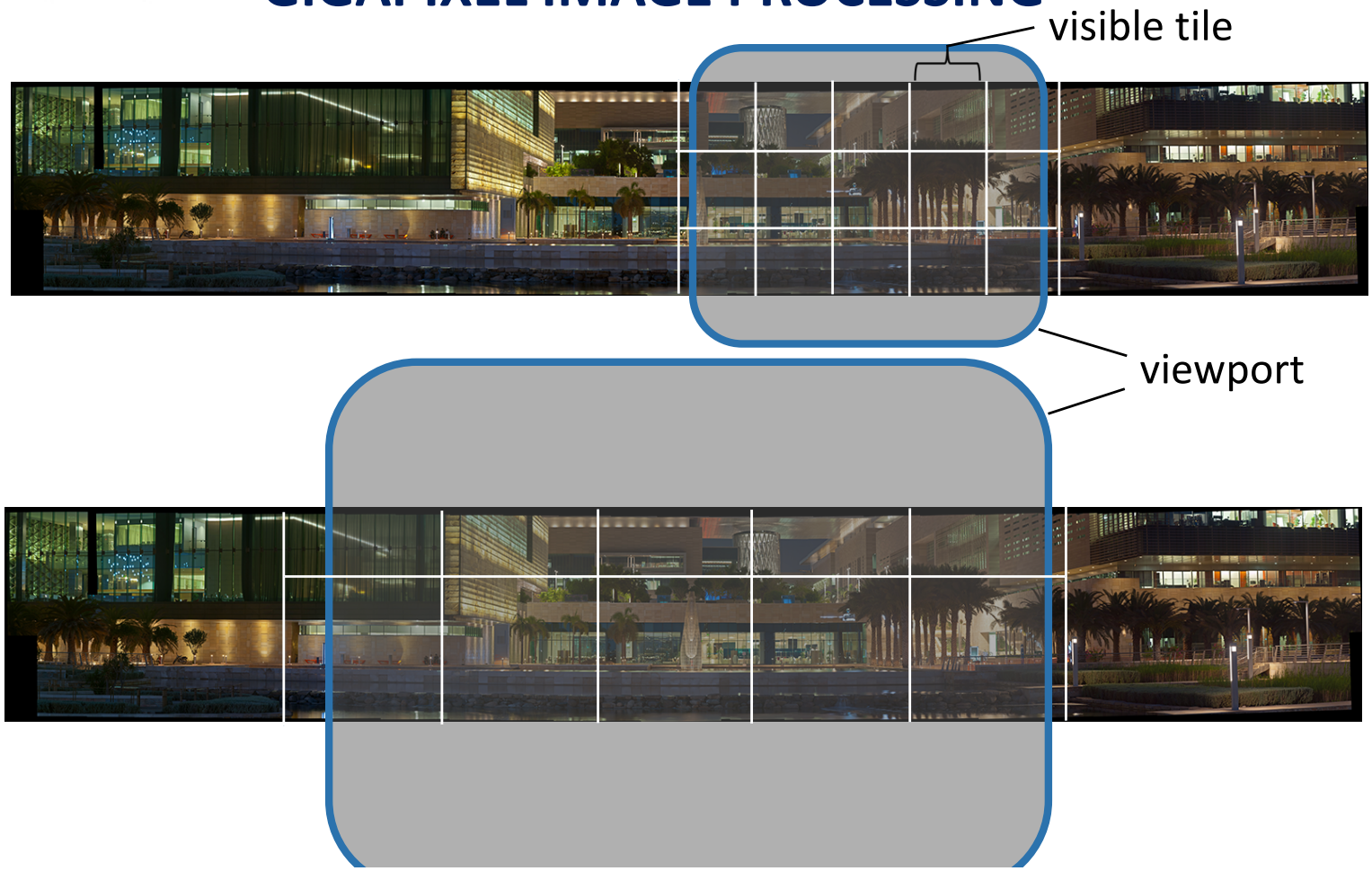
Display-Aware Gigapixel Image Processing

GIGAPIXEL IMAGE PROCESSING

- Out-of-Core Processing
  - Divide data into smaller tiles, process each tile independently (e.g., 256x256)
  - Image operations are performed only on requested sub-tiles (display-aware)
  - Rendering based on tiled data, using GPU-based virtual memory approach



GIGAPIXEL IMAGE PROCESSING



GIGAPIXEL IMAGE PROCESSING

- GPU-based virtual memory architecture [Hadwiger et al. 2012]

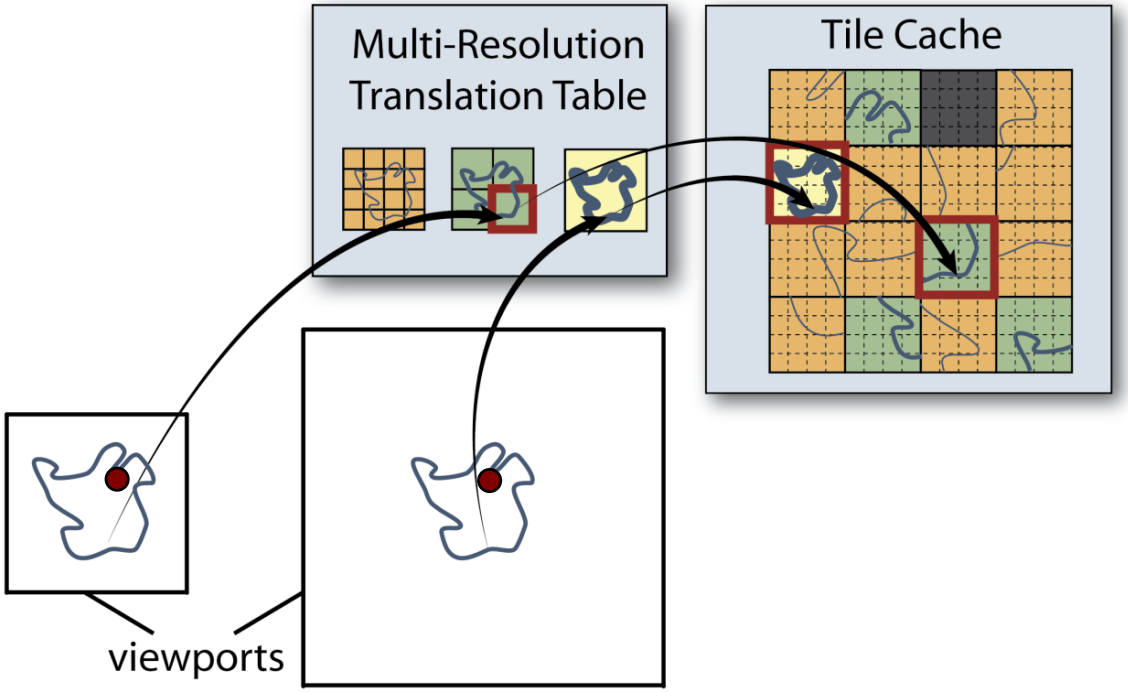
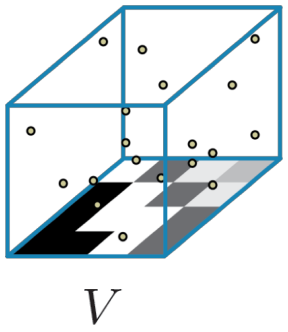


Image Reconstruction

IMAGE RECONSTRUCTION

$$E\left[t_{\boldsymbol{p}}\left(X_{\boldsymbol{p}}\right)\right] = \frac{1}{w_{\boldsymbol{p}}} \int_0^1 t_{\boldsymbol{p}}(r) pdf_{\boldsymbol{p}}(r)\ dr$$



- Key idea # 1

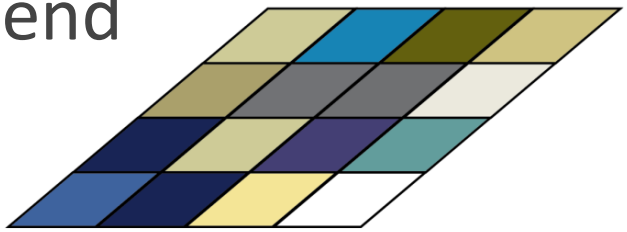
  Use $t_{\boldsymbol{p}}(r)$ instead of $pdf_{\boldsymbol{p}}(r)$

- Key idea # 2

  Pre-convolve V with K: $\tilde{t}_{\boldsymbol{p}} = t_{\boldsymbol{p}} * K$

COLOR MAPPING

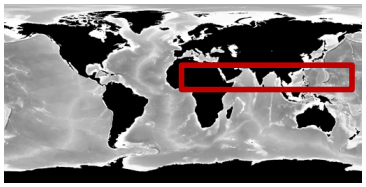
- Pre-convolve color map (with range kernel)
- For each pixel go over its coefficients
  - Apply color map to coefficient and sum up (not spatially convolved yet!)
- One spatial convolution in the end

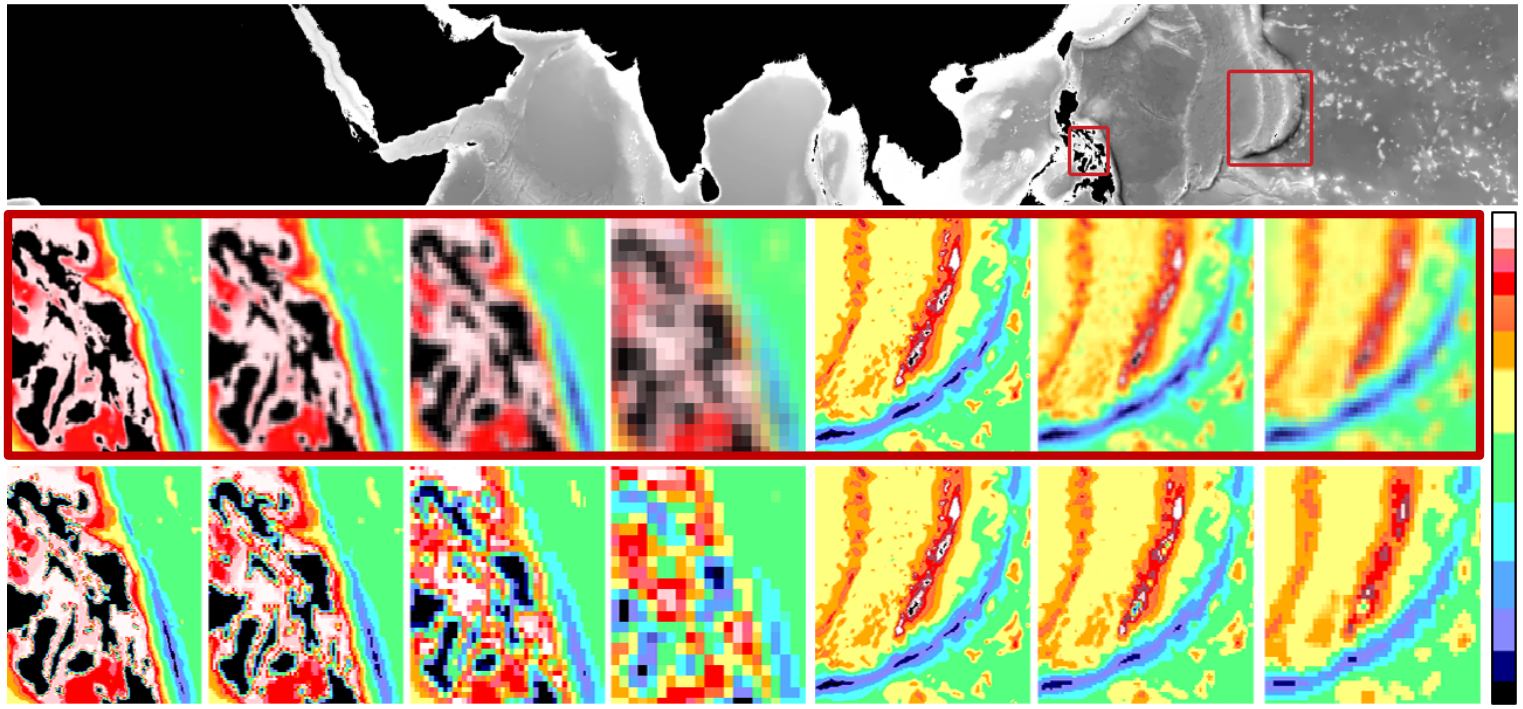


Results

COLOR MAPPING GIGAPIXEL IMAGES

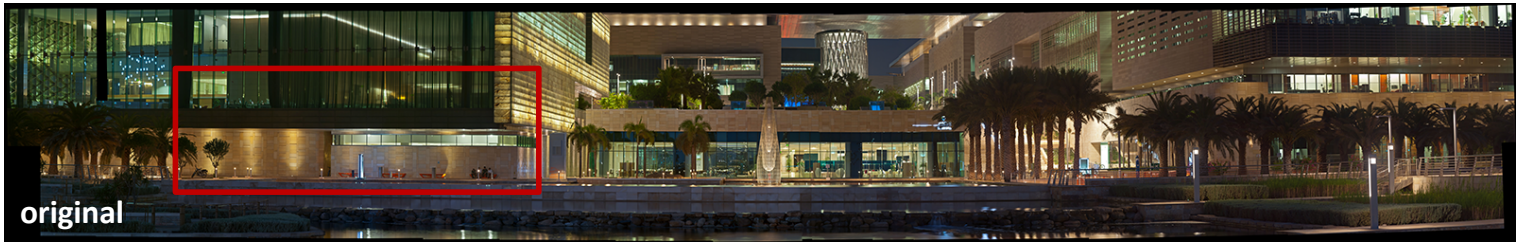
NASA Blue Marble bathymetry: 21,601 x 10,801 pixels (233 Mpixels)

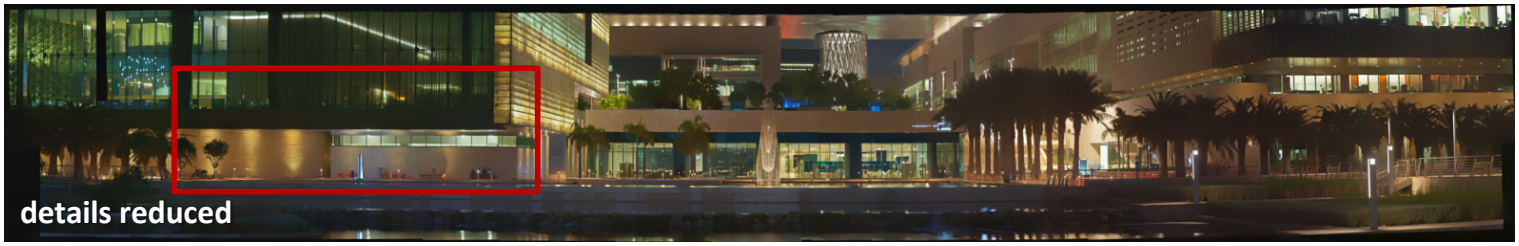


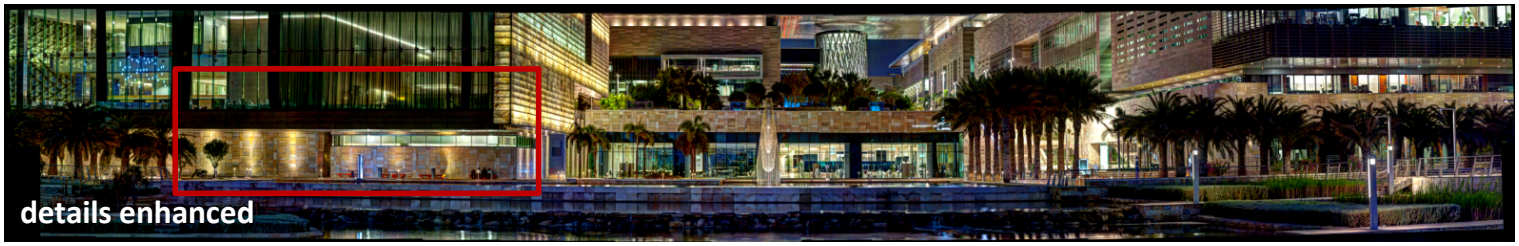



details enhanced
original
details reduced
details enhanced
original
details reduced

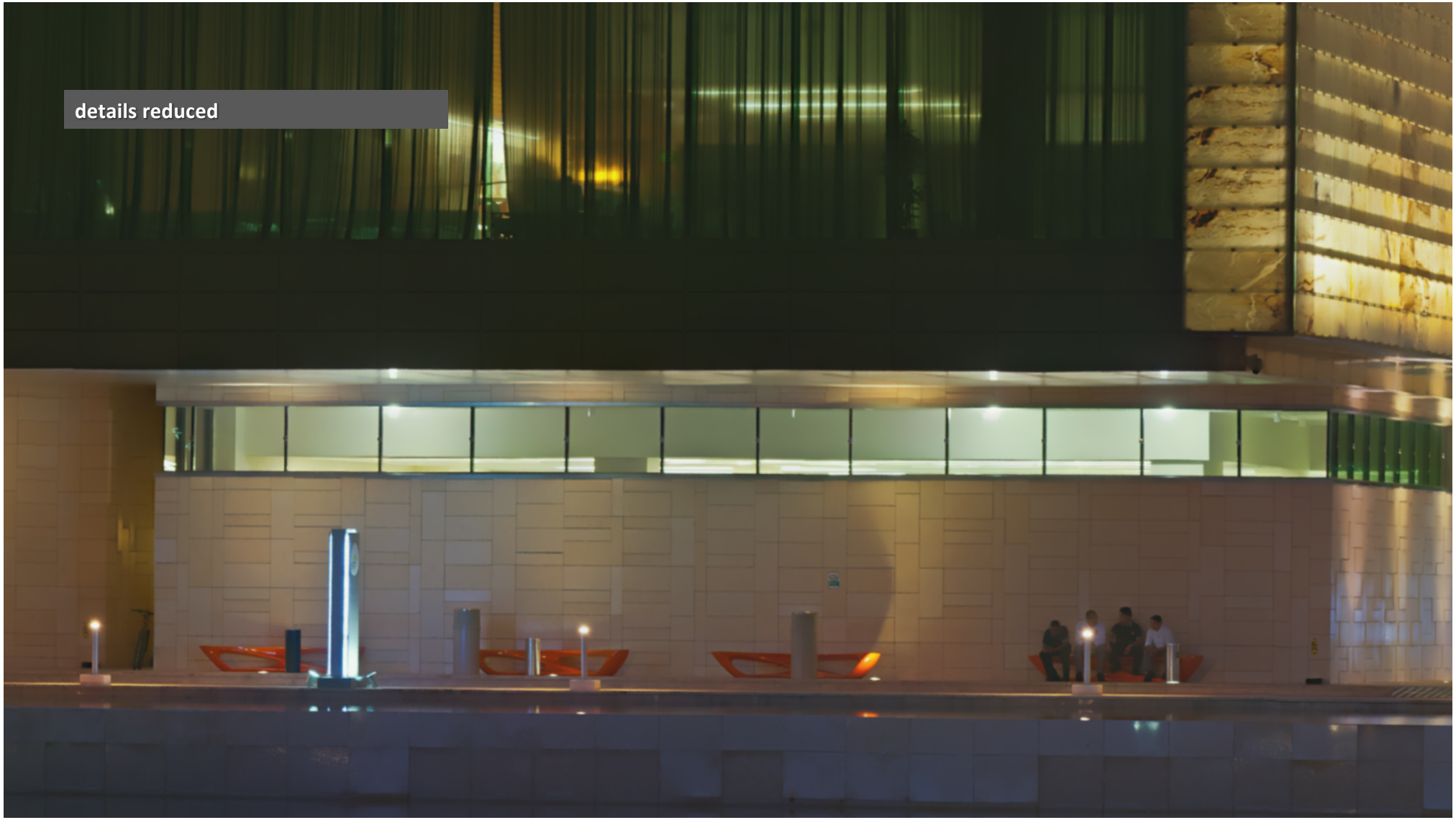
GIGAPIXEL LOCAL LAPLACIAN FILTERING

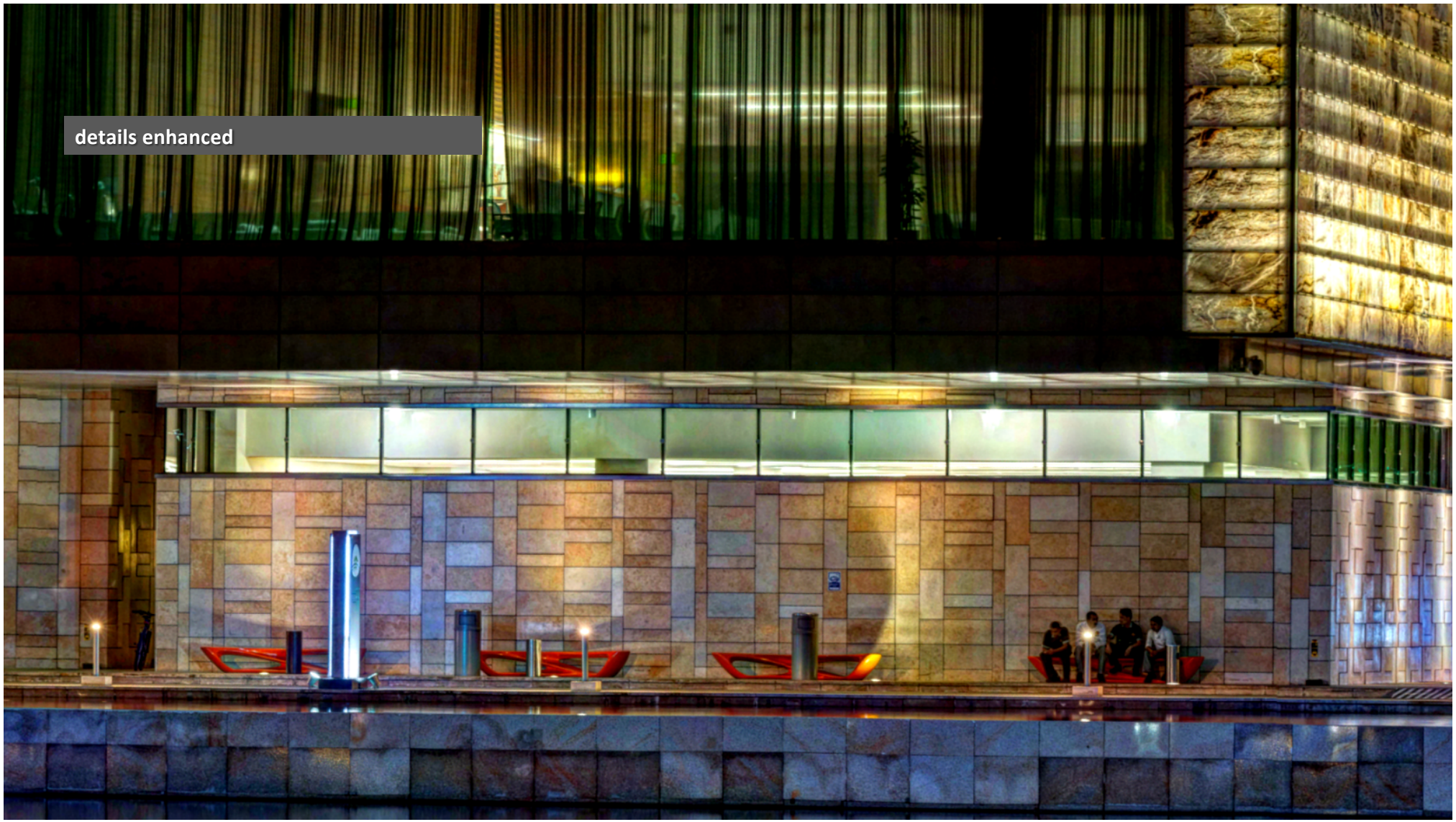







original


details reduced


details enhanced

SUMMARY

Display-aware processing with flexible new image pyramid (spdf map)

- Consistent, sparse representation of pixel footprint pdfs

Unified evaluation of many important non-linear image operations

- Local Laplacian filtering for gigapixel images

Efficient CUDA implementation

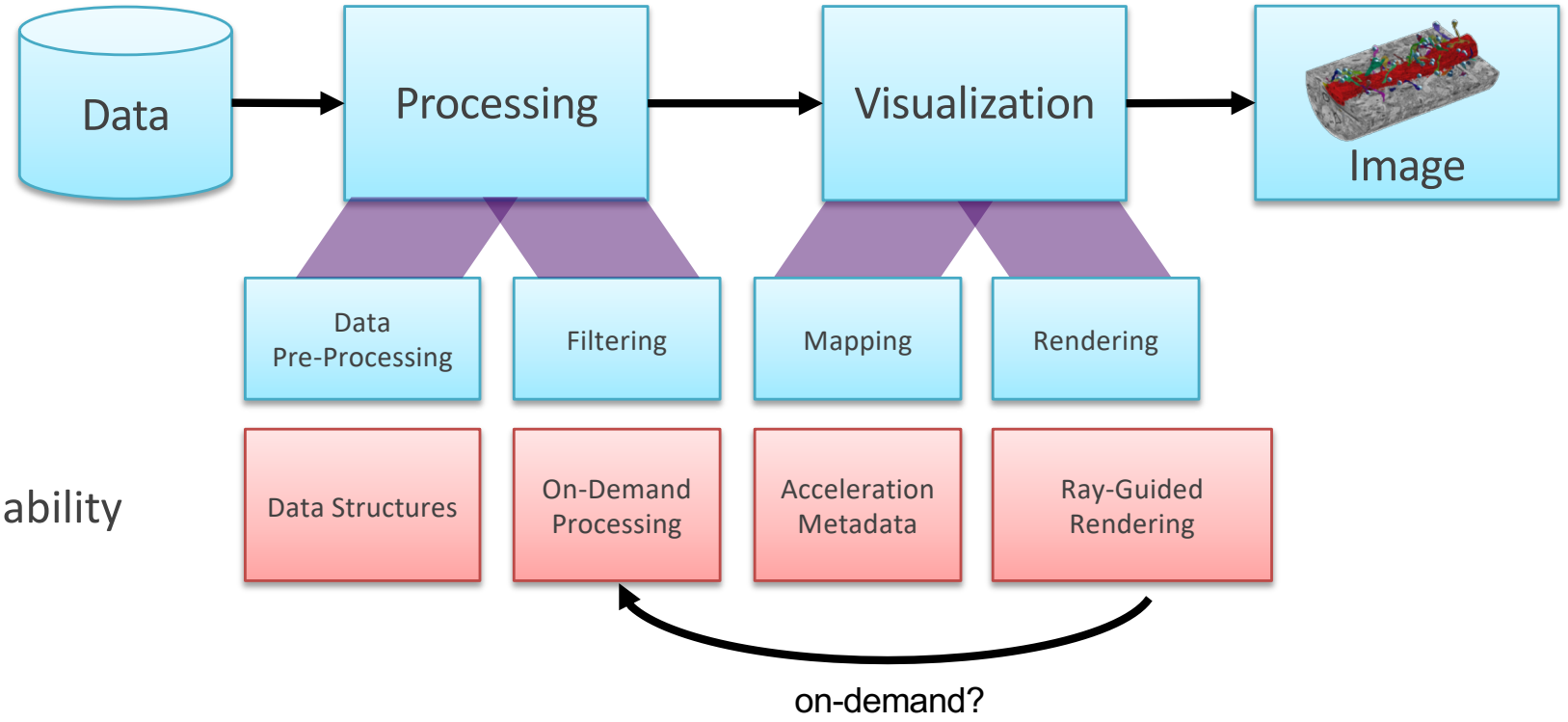
Pre-computation costly, but only performed once

Run time storage and computation similar to standard pyramids

Hadwiger, Sicat, Beyer, Krüger, Möller,
Sparse PDF Maps for Non-Linear Multi-Resolution Image Operations,
Siggraph Asia 2012

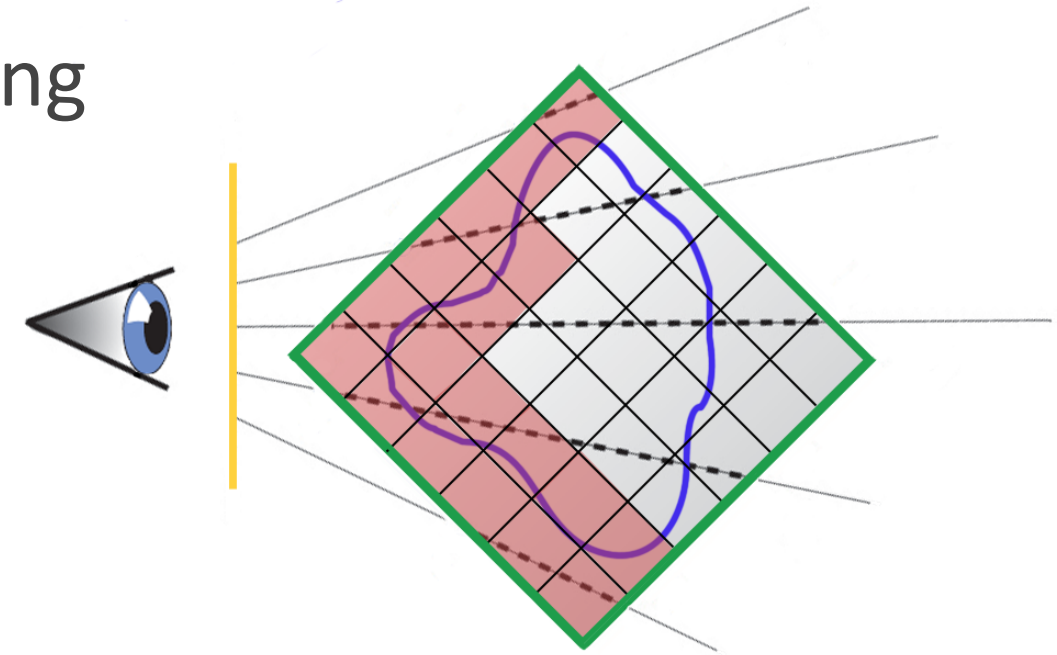
Summary & Outlook

LARGE-SCALE VISUALIZATION PIPELINE



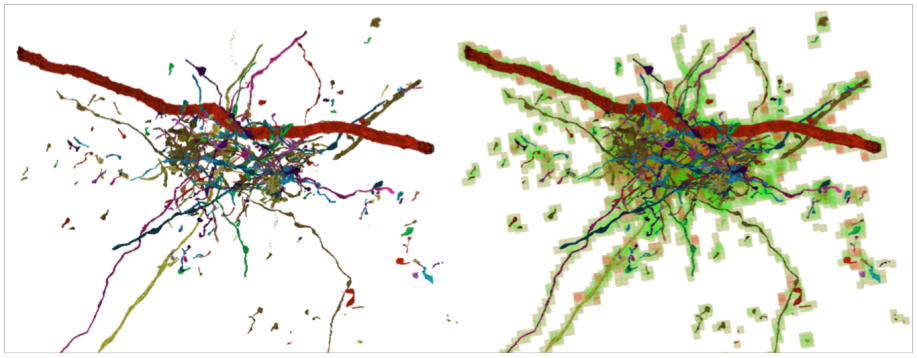
RAY-GUIDED VOLUME RENDERING

- Working set determination on GPU
- Single-pass rendering
- Traversal on GPU
- Virtual texturing



VOLUME RENDERING FOR SEGMENTED DATA

- Empty space skipping essential
- Efficient culling is basis for empty space skipping
  - Compact and scalable data structure (to millions of objects)
  - Hierarchical culling algorithm
- Hybrid approaches
  - Image-order vs. object-order
  - Deterministic vs. probabilistic



SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

Sponsored by

THANK YOU!

Johanna Beyer, Harvard University
Markus Hadwiger, KAUST

Course Website:
http://johanna-b.github.io/LargeSciVis2018/index.html