Display-aware Image Editing

Won-Ki Jeong[1], Micah K. Johnson[2],
Insu Yu[3], Jan Kautz[3],
Hanspeter Pfister[1], Sylvain Paris[4]

[1]Harvard, [2]MIT, [3]UCL, [4]Adobe

Very large images are easy to create

- We are interested in 100+ Mpixels.
- large choice of panorama software
- taking plenty of pictures is easy
  - hand-held with a telephoto lens
  - dedicated devices, e.g. GigaPan
  - medical scanners

Demo

A tour of Hong Kong in 300 megapixels

Challenges

- A lot of pixels to process
  - 100 Mpixels and above
  - everything is slow
- Scene changes completely with location and zoom level
  - no single adjustment works everywhere

Our approach
Display-aware Image Editing

- Only 1 to 4 Mpixels visible at a given time

- Our silver bullet: **Take into account and process only the visible pixels.**

Related work: Fast image filters

- Bilateral grid [Chen 07, Paris 09], edge-aware wavelets [Fattal 09], guided filtering [He 10], geodesic image editing [Criminisi 11]…
  - designed for a few tens of megapixels
  - slow down on 100 Mpixels and above
    - process every pixel

Related work: Bounding boxes

- Process only the affected regions [Shantzis 94]
  - global algorithms need touch every pixel
  - ignore zoom level
    - large regions can appear small when fully zoomed out

Related work: Adaptive tone mapping

- Viewer for HDR panoramas [Kopf 07]
  - adapt viewing parameters to visible image portion
  - does not allow user control

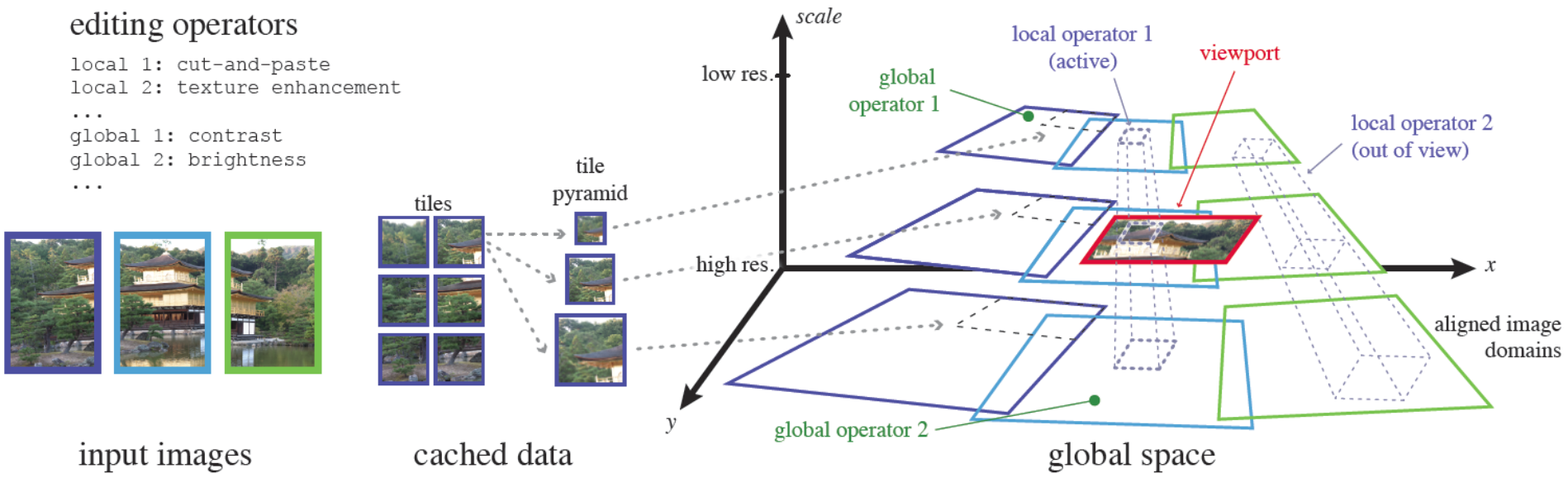
Talk outline

1. Efficient viewer
2. Display-aware algorithms
3. Display-aware viewing parameters

Our viewer

- Design choice: displayed image computed on-the-fly each time
  - dependencies straightforward to handle
- Efficient implementation
  - tile-based cache hierarchy
  - GPU-accelerated filters
  - bounding boxes

Viewer design



We load only the visible data.

We execute only the operators that affect the current view.

Extended in [Jeong et al. Vis'10].

Talk outline

1. Efficient viewer
2. Display-aware algorithms
3. Display-aware viewing parameters

Formal discussion

- A filter f is display-aware if the visible portion of its output can be computed only from the visible portion of the input I.

- Sufficient to process Laplacian pyramids locally
  - output coefficients depend only on nearby coefficients
  - proof in the paper
  - also a limitation, i.e. constrains algorithm design

Formal discussion

- A filter f is display-aware if the visible portion of its output can be computed only from the visible portion of the input I.
  - s is the operator that display an image on the screen
  - f is display-aware if $s(f(I)) = f(s(I))$

Formal discussion

- Displaying an image I on the screen, $s(I)$:
  1. low-pass filter followed by downsampling
     - equivalent to crop in Fourier domain
  2. crop in image space
- To commute with s, a filter f should not use data than may be cropped by s
  - use only local data in space and frequency
  - **Local processing on Laplacian pyramids**

Example application: Seamless image compositing

- Standard approach: gradient-domain + Poisson eq.
  - global optimization: does not scale up

Our approach

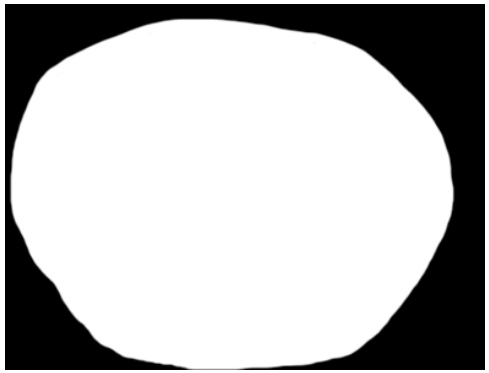
- Inspired from [Farbman 09]
  - core operation is computing a smooth interpolant of offset values defined at boundaries

- We use pyramid-based interpolant [Burt 88]
  - classical push-pull operation

Input data

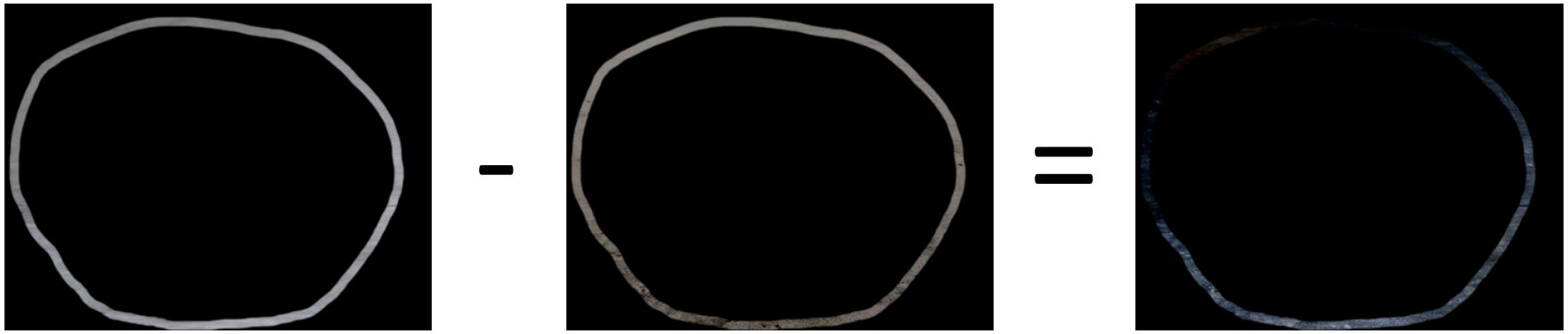
foreground

background



mask

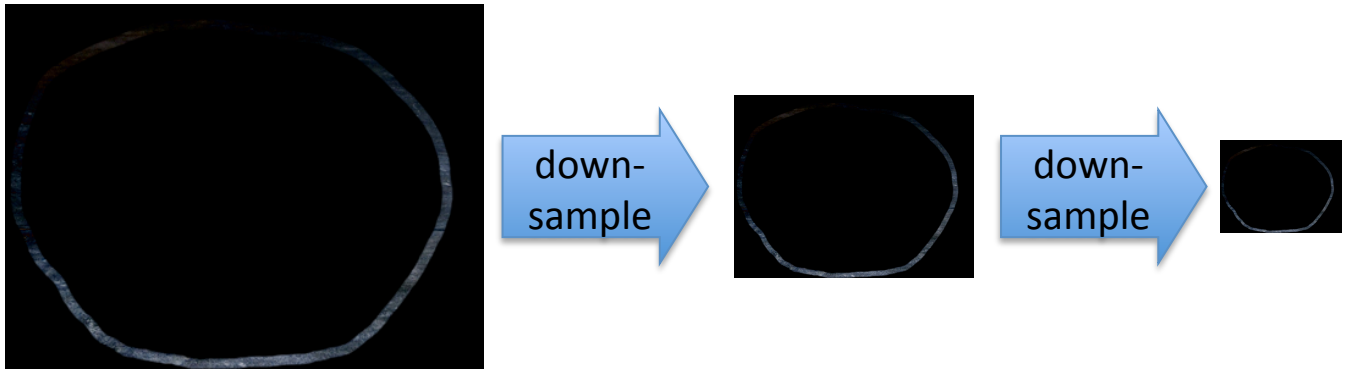
1- Compute offsets at boundary



foreground background offsets

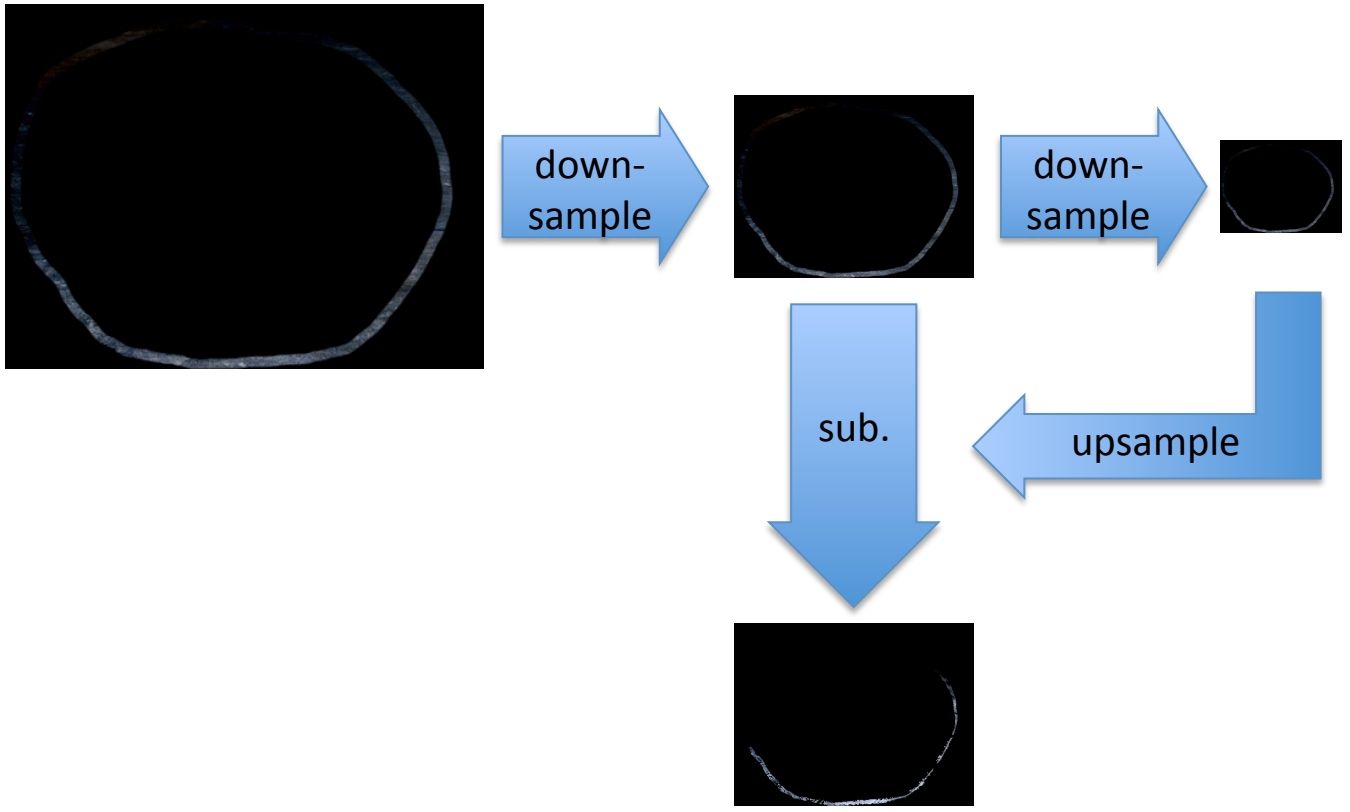
2- Build a Laplacian pyramid with the offsets

Gaussian pyramid:



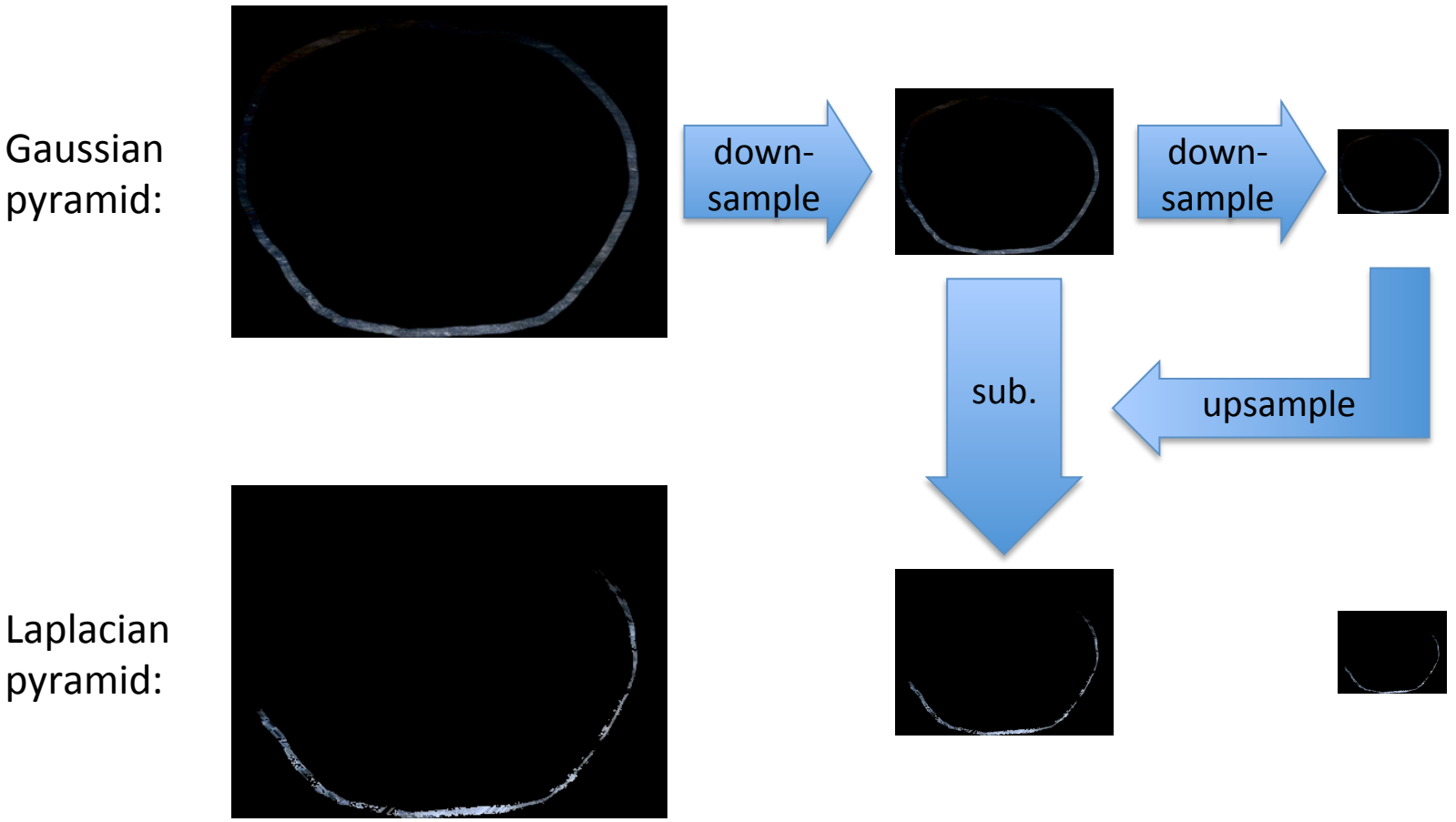
2- Build a Laplacian pyramid with the offsets

Gaussian pyramid:



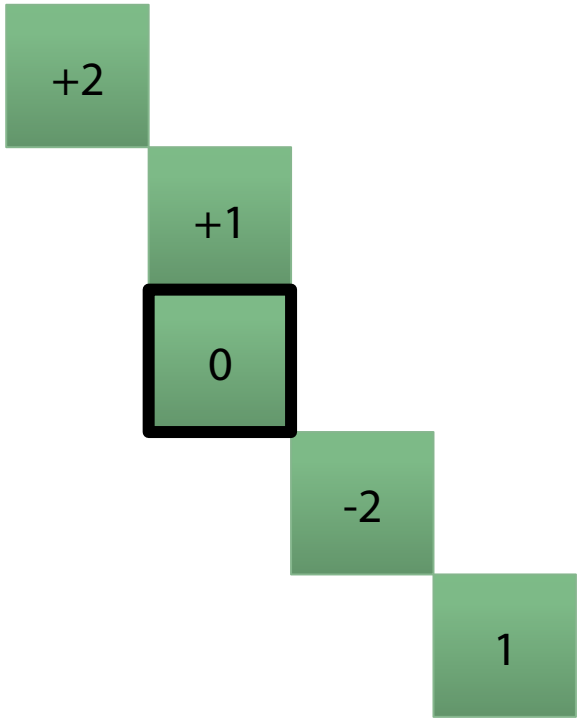
Laplacian pyramid:

2- Build a Laplacian pyramid with the offsets

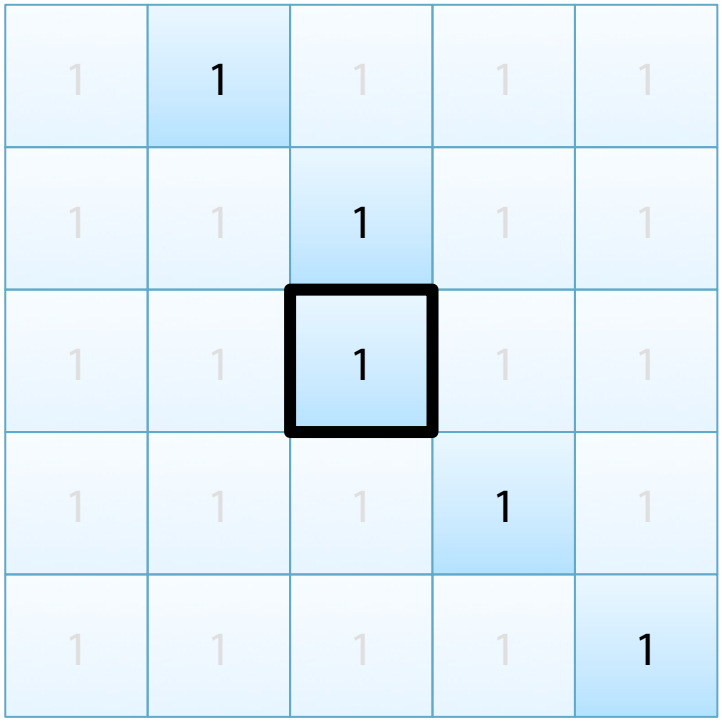


2- Build a Laplacian pyramid with the offsets

To downsample, keep only the values where there are data.



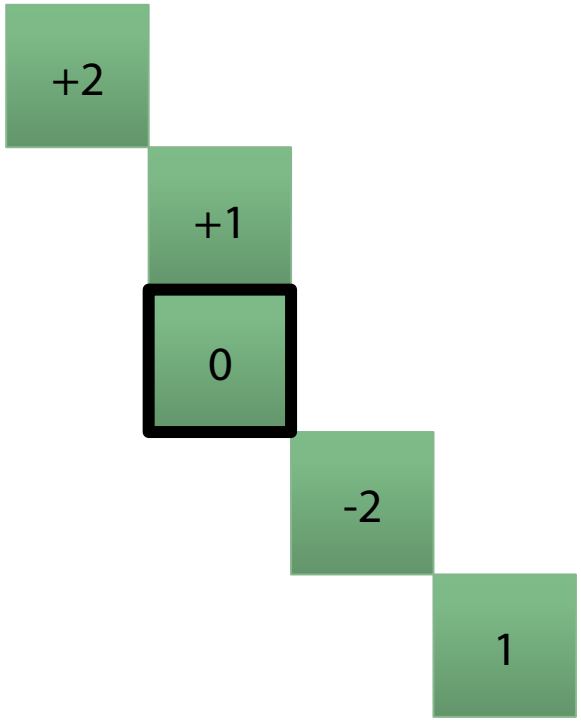
boundary offset values



blurring kernel
(box for illustration)

2- Build a Laplacian pyramid with the offsets

To upsample, keep all the values.



boundary offset values

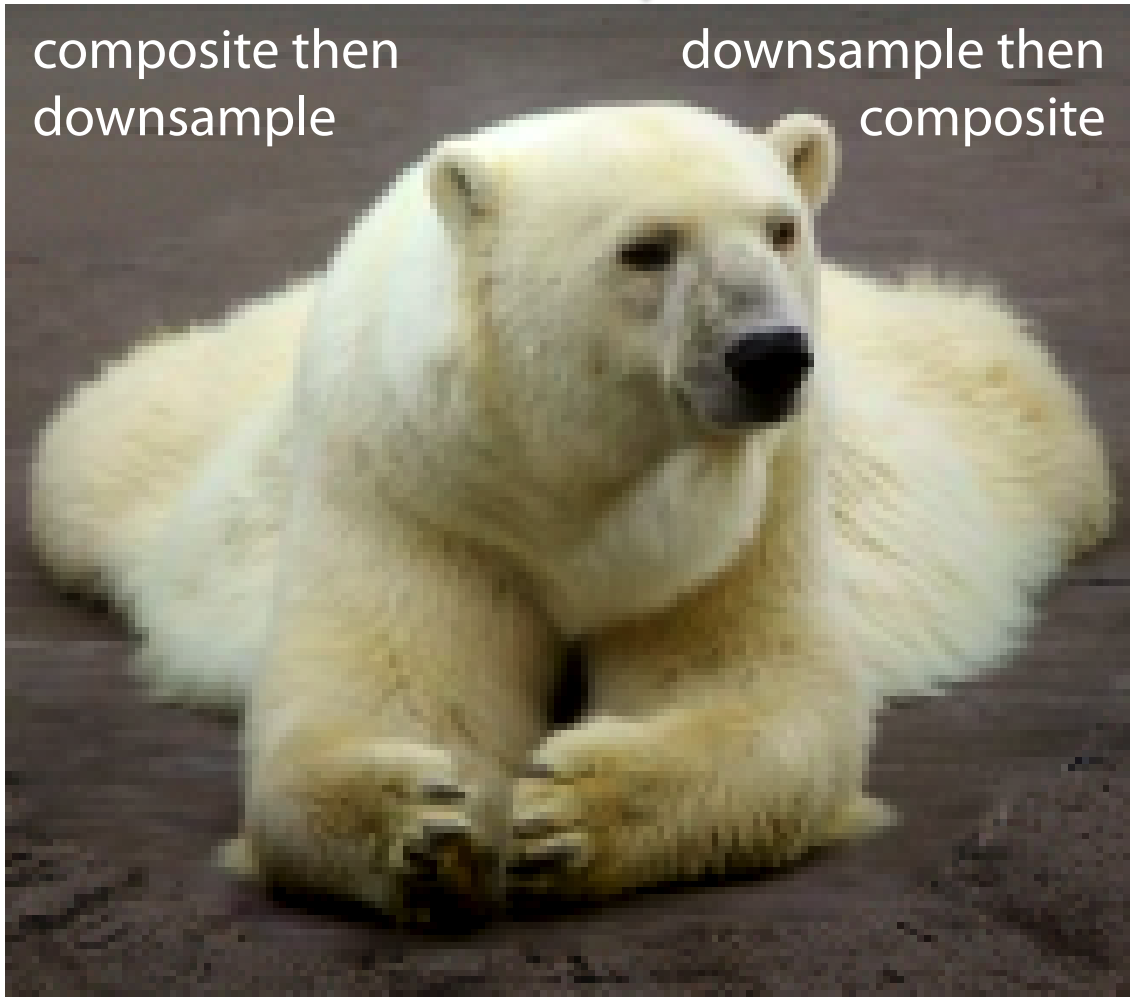
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

blurring kernel
(box for illustration)

3- Collapse the pyramid

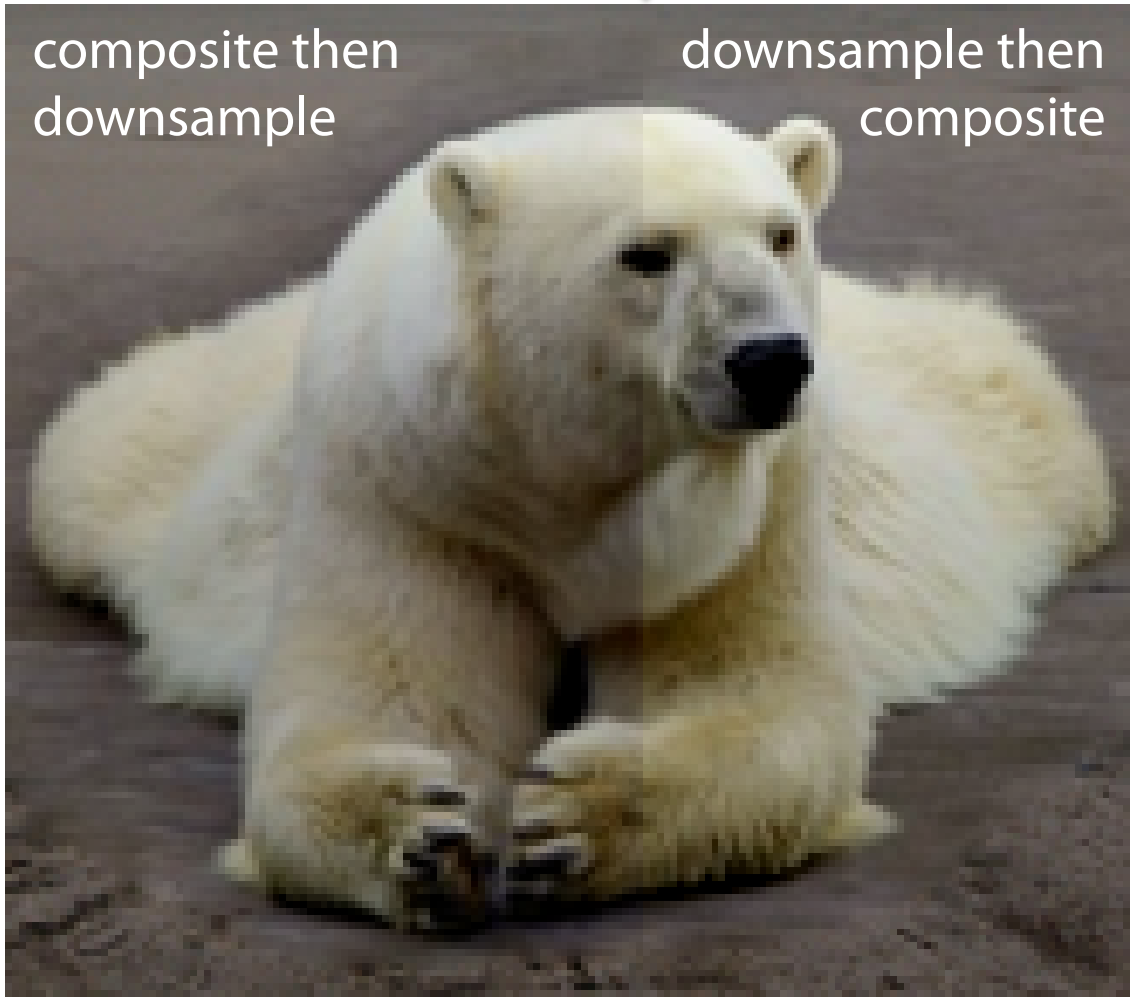
our result

Is it really display-aware?

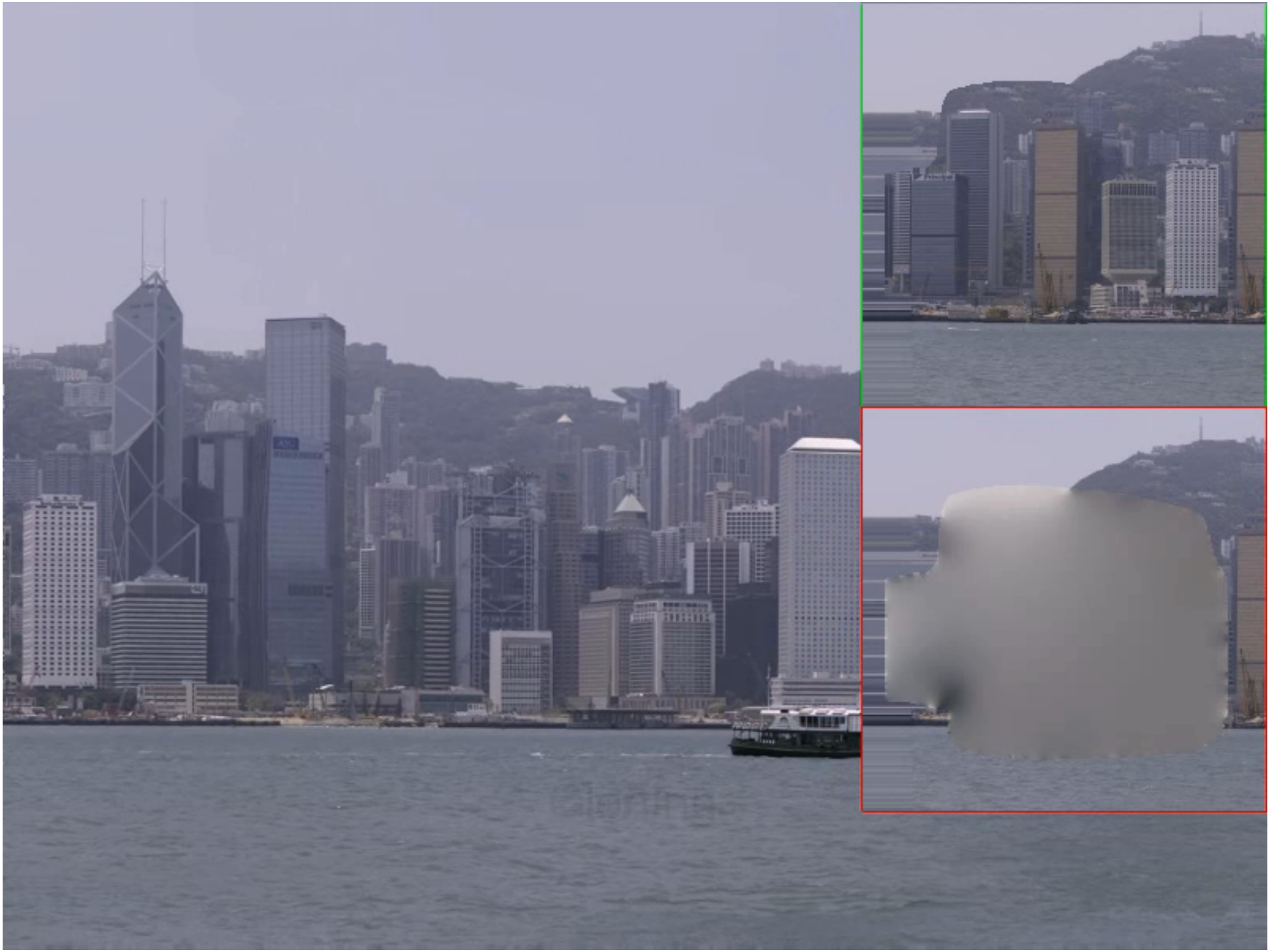


our result

Is it really display-aware?

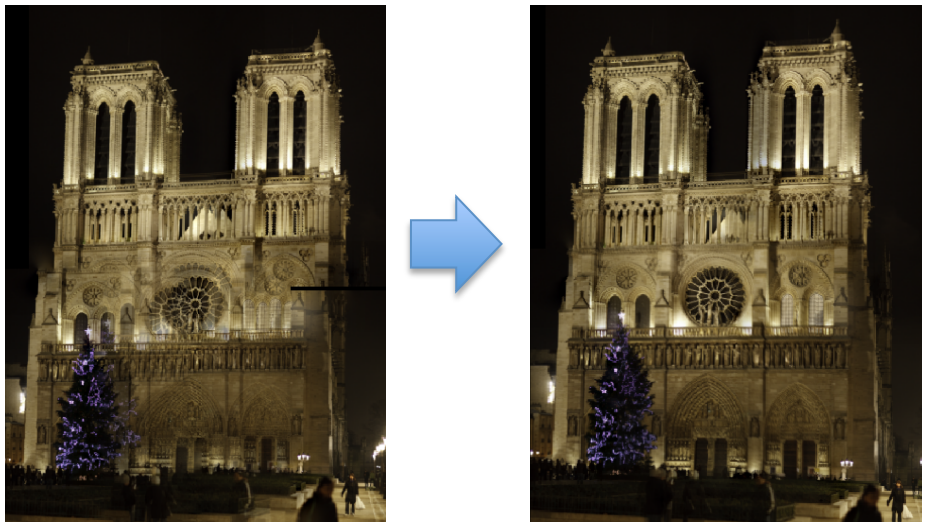


Photoshop result (gradient domain)



Other Applications

- Image alignment
  (see the paper)



- Edge-aware image editing
  (to appear at Siggraph)

Talk outline

1. Efficient viewer
2. Display-aware algorithms
3. Display-aware viewing parameters

Objective

- Adapting the viewing parameters (brightness, contrast…) to the current view

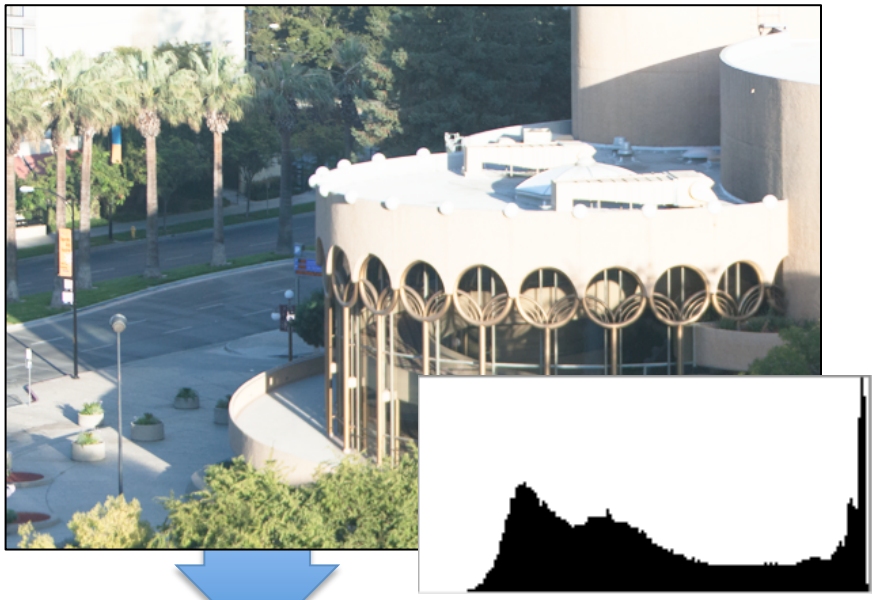
- Automatic by default, user can override

User study (Mechanical Turks)

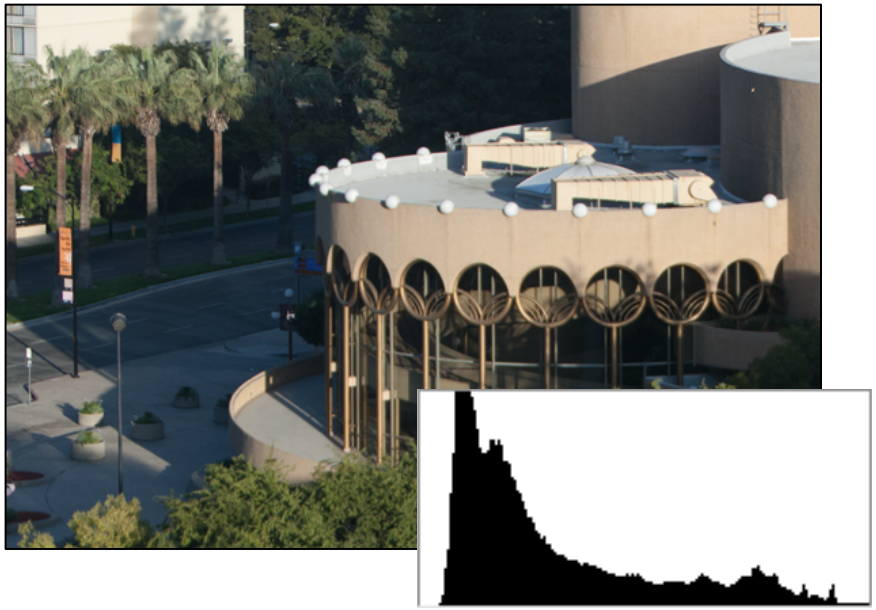
- Users adjust photos.
  - instruction: neutral pleasing rendition (postcard)

- We compare before and after.
  - **Image histograms are more similar after adjustment than before**.
  - Confirms the results of [Bae 06]

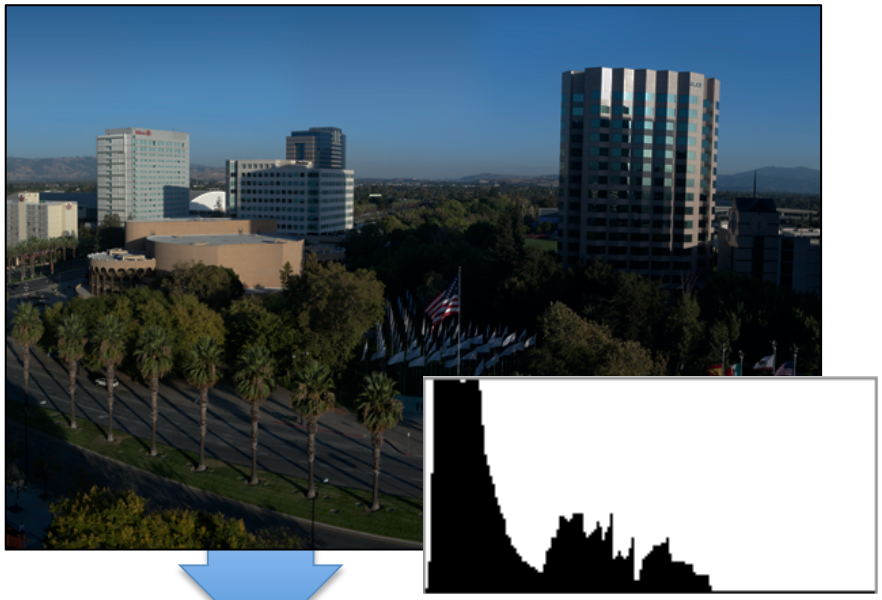
before adjustment



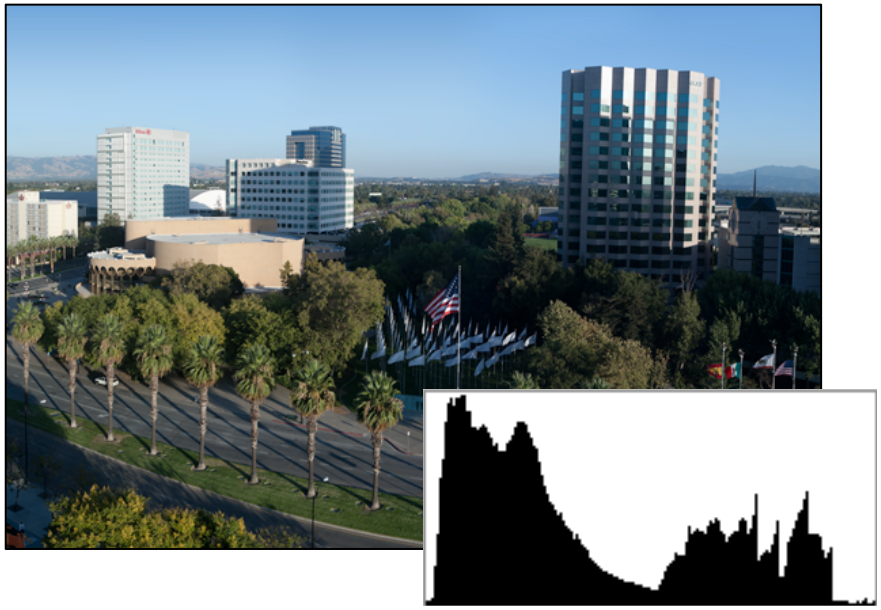
after



before adjustment



after



Our strategy

Keep histogram as constant as possible.

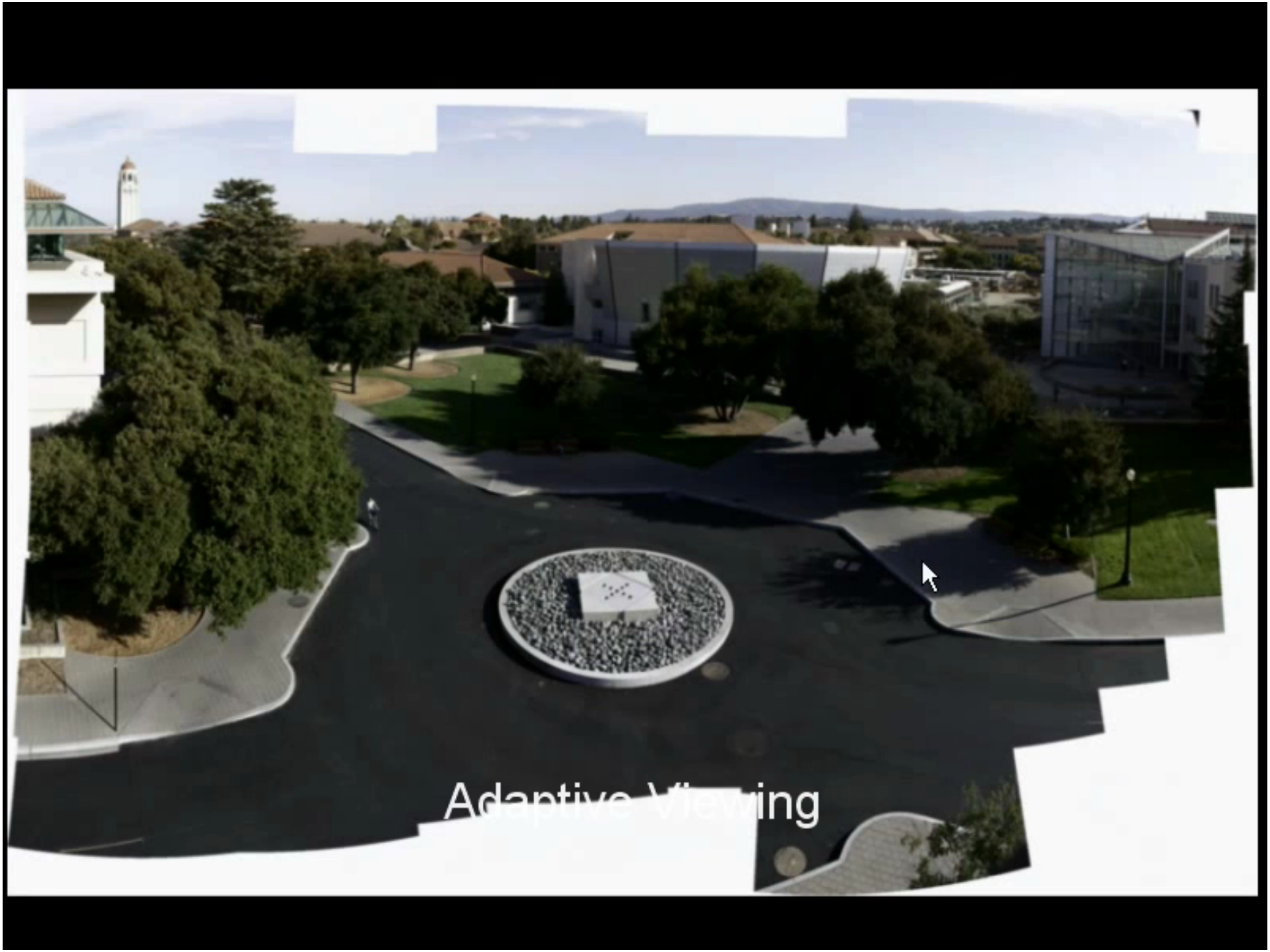
Algorithm overview

1. User specifies parameters at a few locations and scales.

2. For unspecified views, find k nearest neighbors with edits.
   - distance = EMD on unedited image histogram

3. Compute average edited histogram of the k neighbors

4. Estimate viewing parameters to best match that histogram


Adaptive Viewing

Conclusion

- We enable the editing of
  very large images (1+ Gpixel).
- Laplacian pyramids are key.
- Histogram distance
  characterizes adjustments.
- Well-suited for mobile devices with small screens and limited computational power.

Acknowledgments

We thank Min H. Kim for his help producing the supplementary video.

Insu Yu was in part supported by EPSRC grant EP/E047343/1. Won-Ki Jeong and Hanspeter Pfister were partially supported by the Intel Science and Technology Center for Visual Computing, Nvidia, and the National Institute of Health under Grant No. HHSN276201000488P. This material is based upon work supported by the National Science Foundation under Grant No. 0739255.


HAPPY BIRTHDAY!
HAPPY BIRTHDAY!

Conclusion

- We enable the editing of
  very large images (1+ Gpixel).
- Laplacian pyramids are key.
- Histogram distance
  characterizes adjustments.
- Well-suited for mobile devices with small screens and limited computational power.