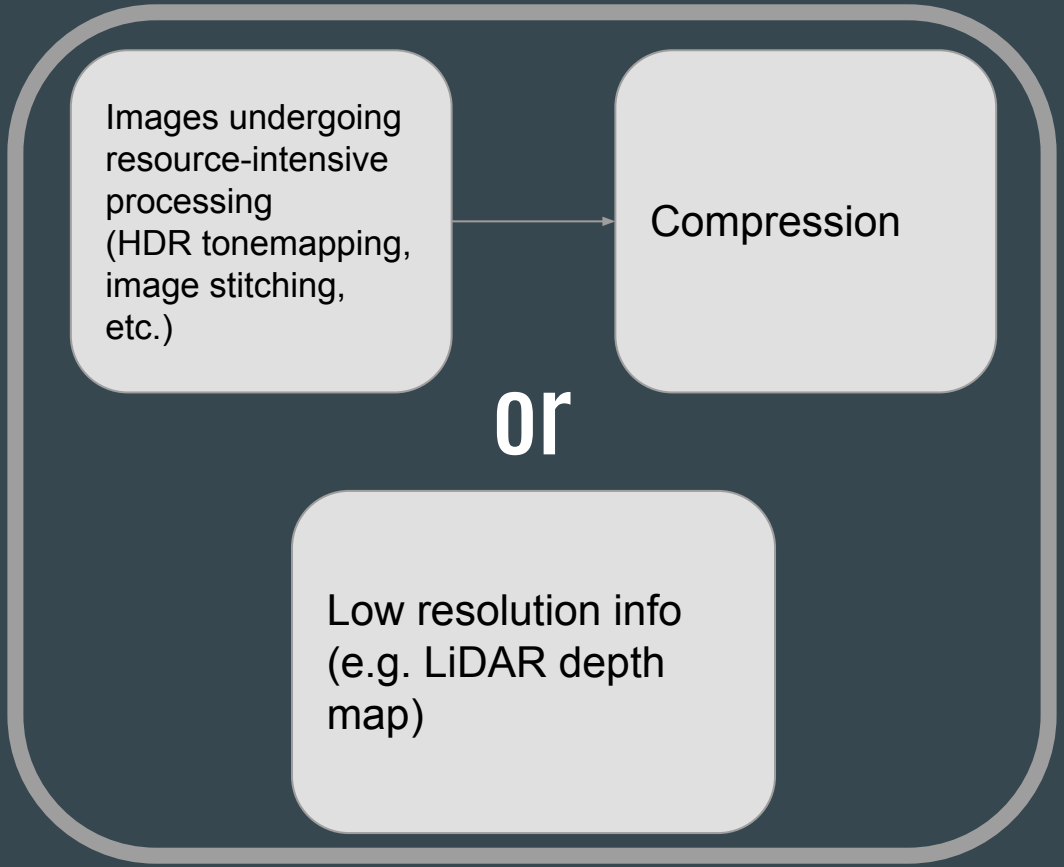
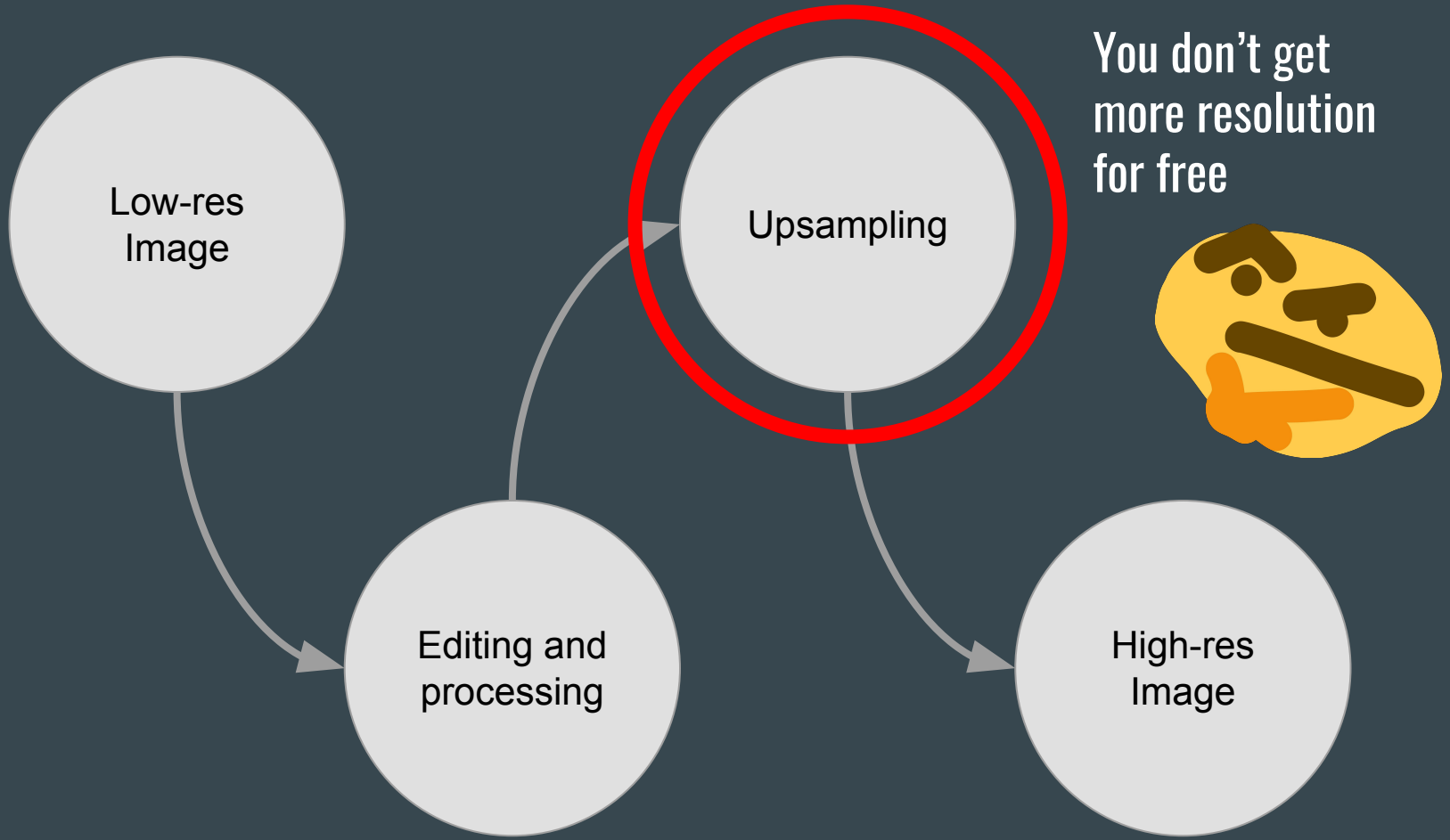
Joint Bilateral Upsampling

•••

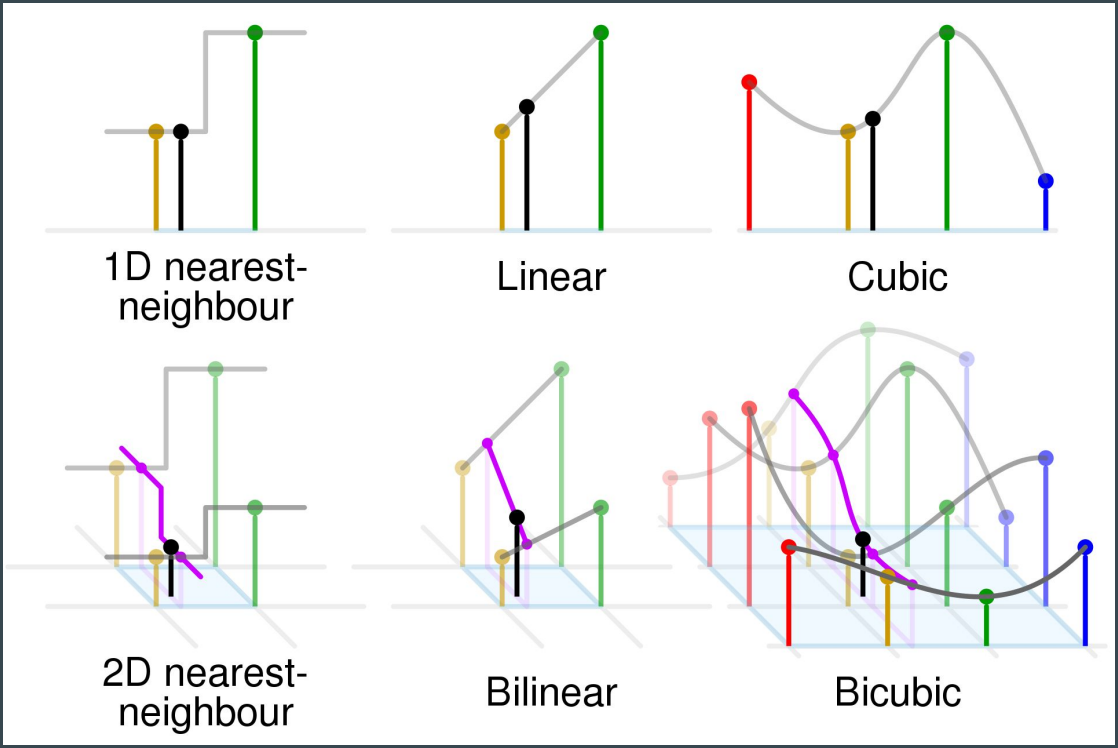
Joyce Zhang, Jason Xu

Problem Description


Images undergoing resource-intensive processing (HDR tonemapping, image stitching, etc.)
Compression
or
Low resolution info (e.g. LiDAR depth map)


Low-res Image
Upsampling
You don’t get more resolution for free
Editing and processing
High-res Image

Common Upsampling Methods



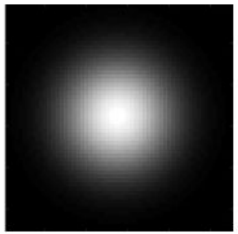
Bilateral & Joint Bilateral Filtering

Bilateral Filtering

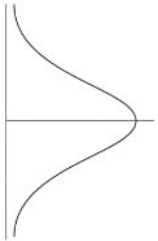
$$BF[I]_p = \frac{1}{W_p} \sum_{q \epsilon S} G_{\sigma_s}(||p-q||) G_{\sigma_r}(|I_p - I_q|) I_q$$

Normalization Factor

Space Weight



Range Weight



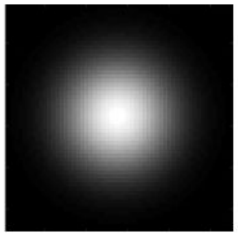
Maybe we can use the intensity kernel of another image?

Joint Bilateral Filtering

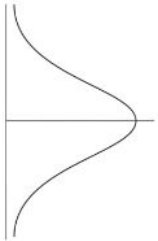
$$BF[I]_p = \frac{1}{W_p} \sum_{q \epsilon S} G_{\sigma_s}(||p-q||) G_{\sigma_r}(|I_p - I_q|) I_q$$

Normalization Factor

Space Weight



Range Weight



Replace with intensity kernel from another image

Bilateral filter denoising

Joint bilateral filter denoising
(Intensity kernel from flash image)

Method in Paper

High Level Idea: Get the color information from the low res, and the details from the original.

S Low-resolution image

$p_\downarrow$ and $q_\downarrow$ corresponding (possibly fractional) coordinates

$\tilde{I}$ full resolution image

p and q integer coordinates

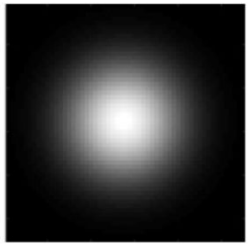
generate **I, S** → Expensive Image Operation → Appy same blur kernel to **I, S** → Joint Bilateral Upscaling

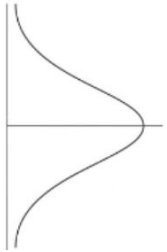
$$\tilde{S}_p = \frac{1}{k_p} \sum_{q_\downarrow \in \Omega} S_{q_\downarrow} \, f(||p_\downarrow - q_\downarrow||) \, g(||\tilde{I}_p - \tilde{I}_q||)$$

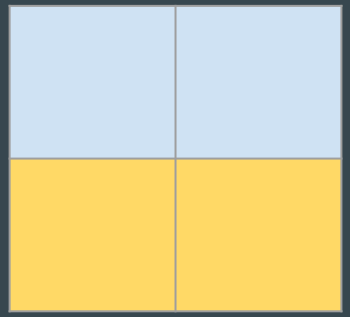
Spatial Range

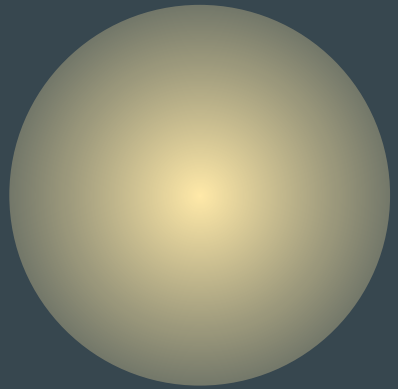
Range Weight

PNote, that q↓ takes only integer coordinates in the low resolution solution. Therefore the guidance image is only sparsely sampled process

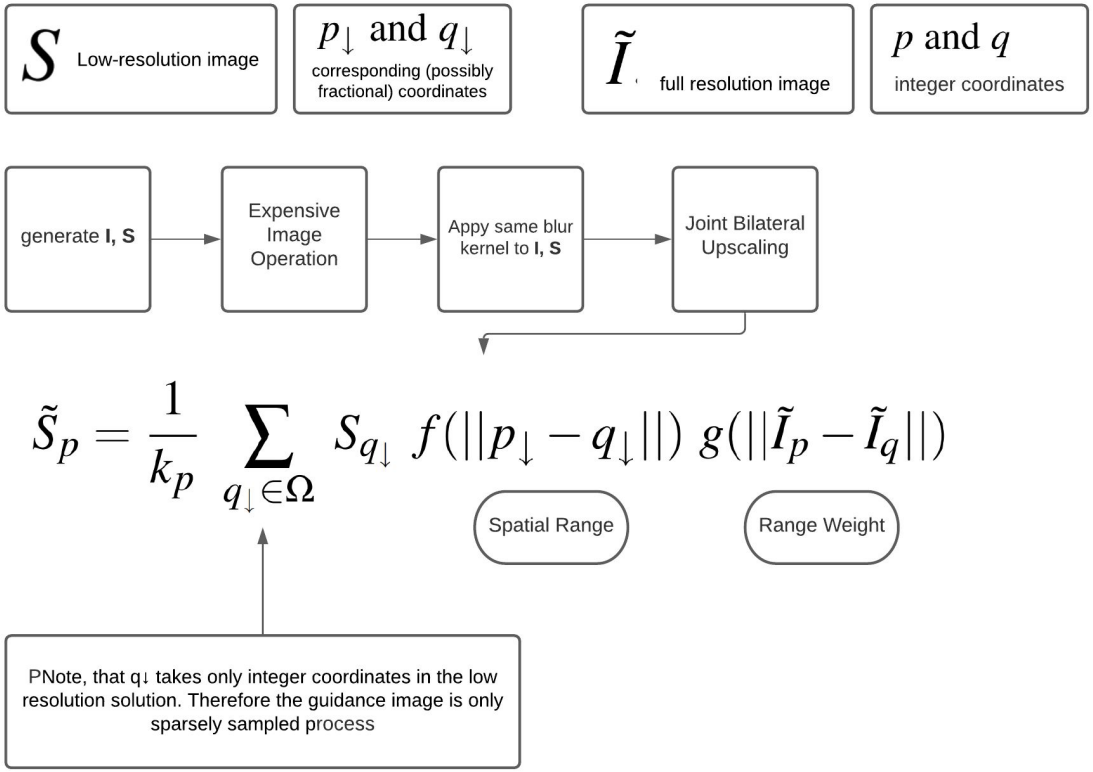








High Level Idea: Get the color information from the low res, and the details from the original.



O(Nr^2) where N is the output image size and r is the domain filter radius

Expensive Image operations such as...

- Stereo Depth
- Colorization
- Tone Mapping

Let’s look at some examples...

Results

Colorization (Levin, Colorization using optimization)

Colorized black and white image based on user input

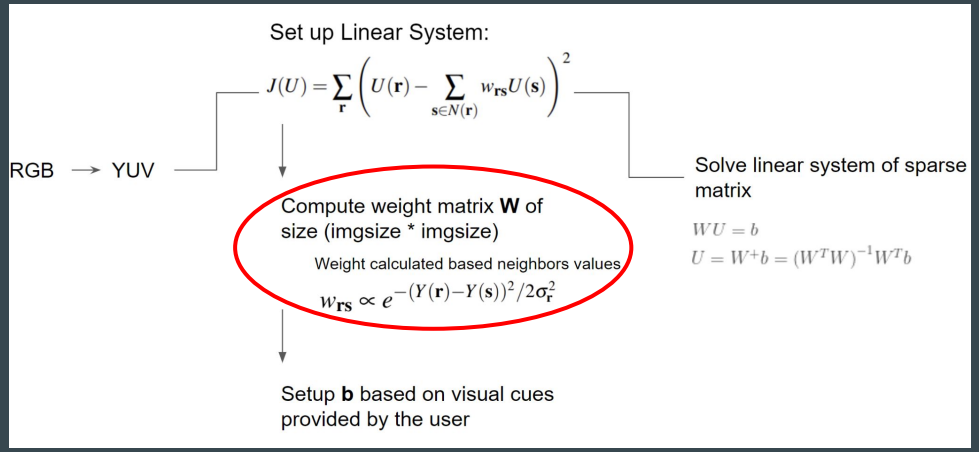
Algorithm:

Most important point:

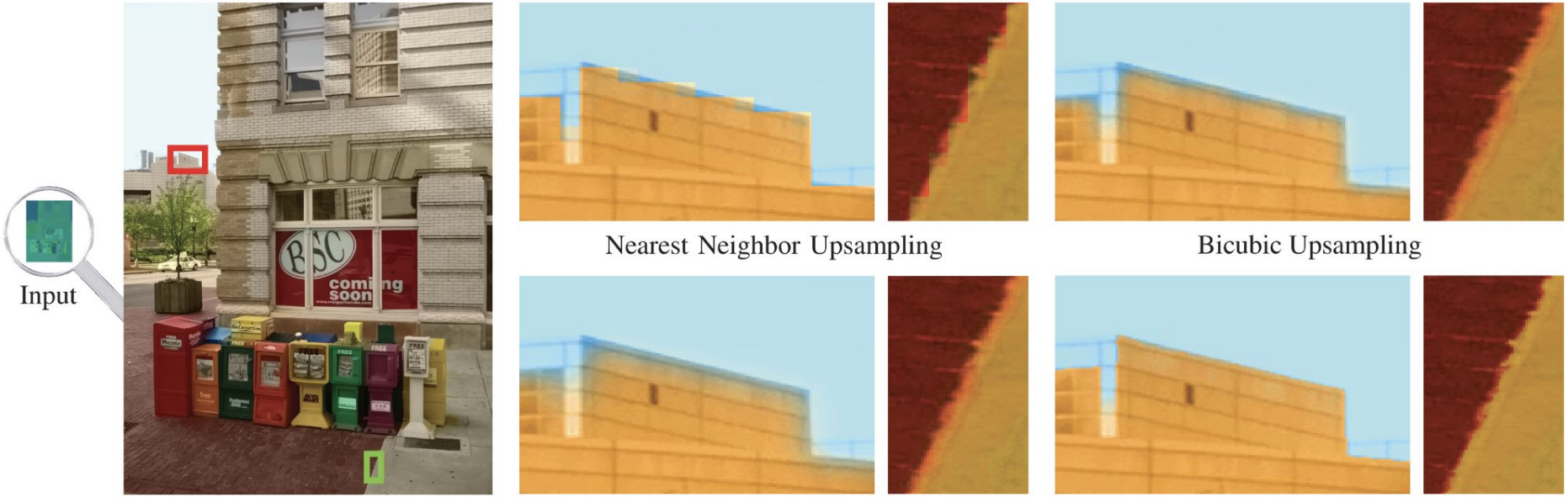
Problem gets a lot more complex the more pixels you have

But colorization results are better with more user input

Which is why upsampling is desired!



From Joyce's CompPhoto Project last semester.



The low resolution pseudo-colored chrominance solution is at left. Note that the JBU result does not exhibit color spills!

Tone Mapping

Discussion

- ➔ Simple and massively resource-saving
  - ◆ Reduces thrashing
  - ◆ Only requires single sweep
- ➔ Limitation: Can only be applied to image processing that doesn't alter the scene layout
  - ◆ Adding a cloud in the low resolution will not translate well through upscaling as there's no information/detail to support it
  - ◆ If applying perceptive transformations into low resolution, similar changes also need to be applied to high resolution

References

- Joint Bilateral Upsampling, SIGGRAPH 2007
  - https://johanneskopf.de/publications/jbu/paper/FinalPaper_0185.pdf
- Image scaling, Wikipedia
  - https://en.wikipedia.org/wiki/Image_scaling
- Bilateral Filter, Wikipedia
  - https://en.wikipedia.org/wiki/Bilateral_filter
- Python: Bilateral Filtering, GeeksForGeeks
  - https://www.geeksforgeeks.org/python-bilateral-filtering/
-