Ideas

- *Joint Bilateral Upsampling*
  - Kopf *et al.*, SIGGRAPH 2007

Upsampling in Image Processing

- In many IP applications, computational and memory costs often require that a smaller solution be run over a downsampled image

- The final result is reconstructed through upsampling

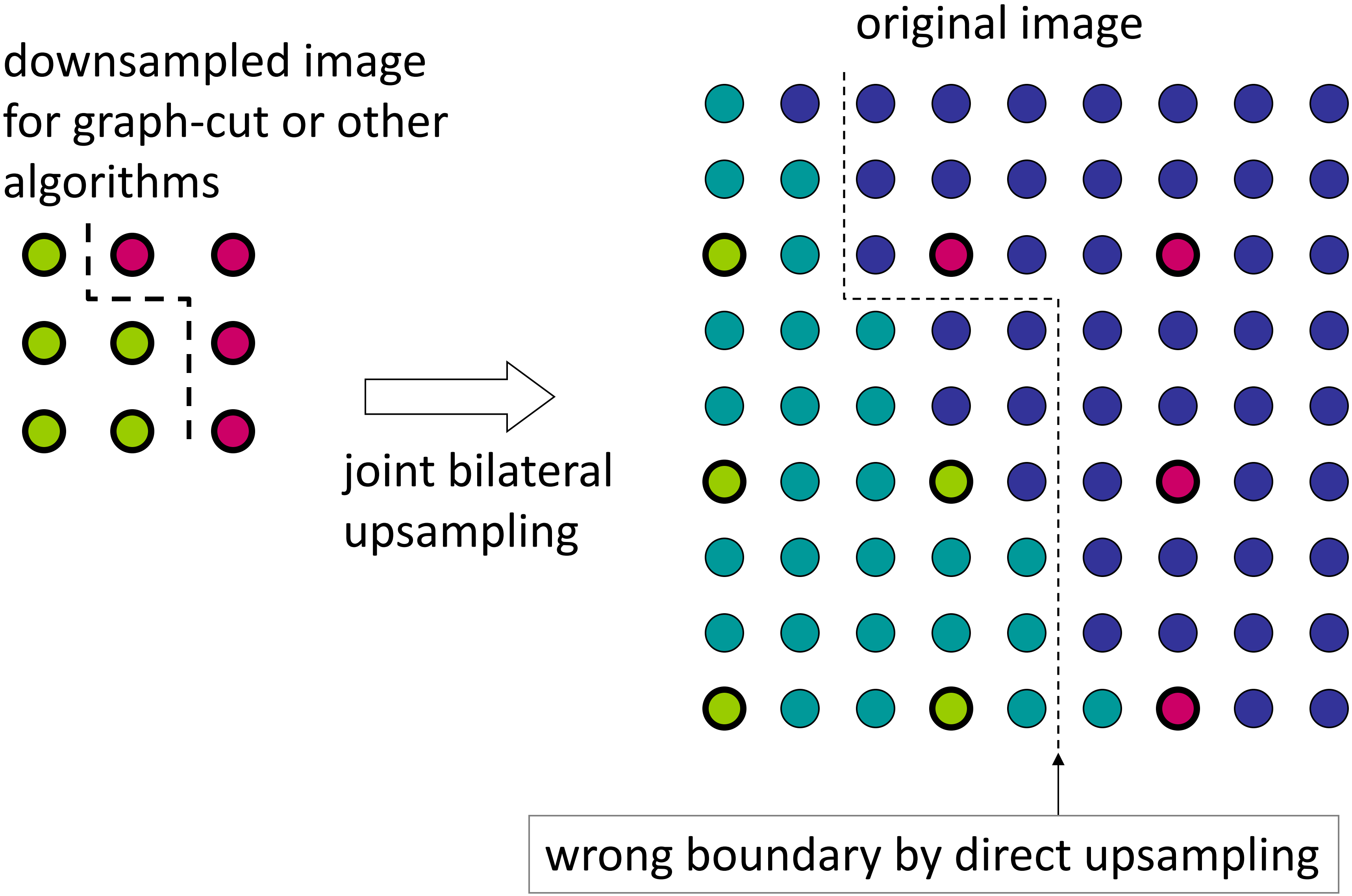
Joint Bilateral Upsampling

- **Idea**: Using the available high resolution input image as a prior in the context of a joint bilateral upsampling procedure to produce a better high resolution solution

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

low resolution

Use the Original Image as a Reference



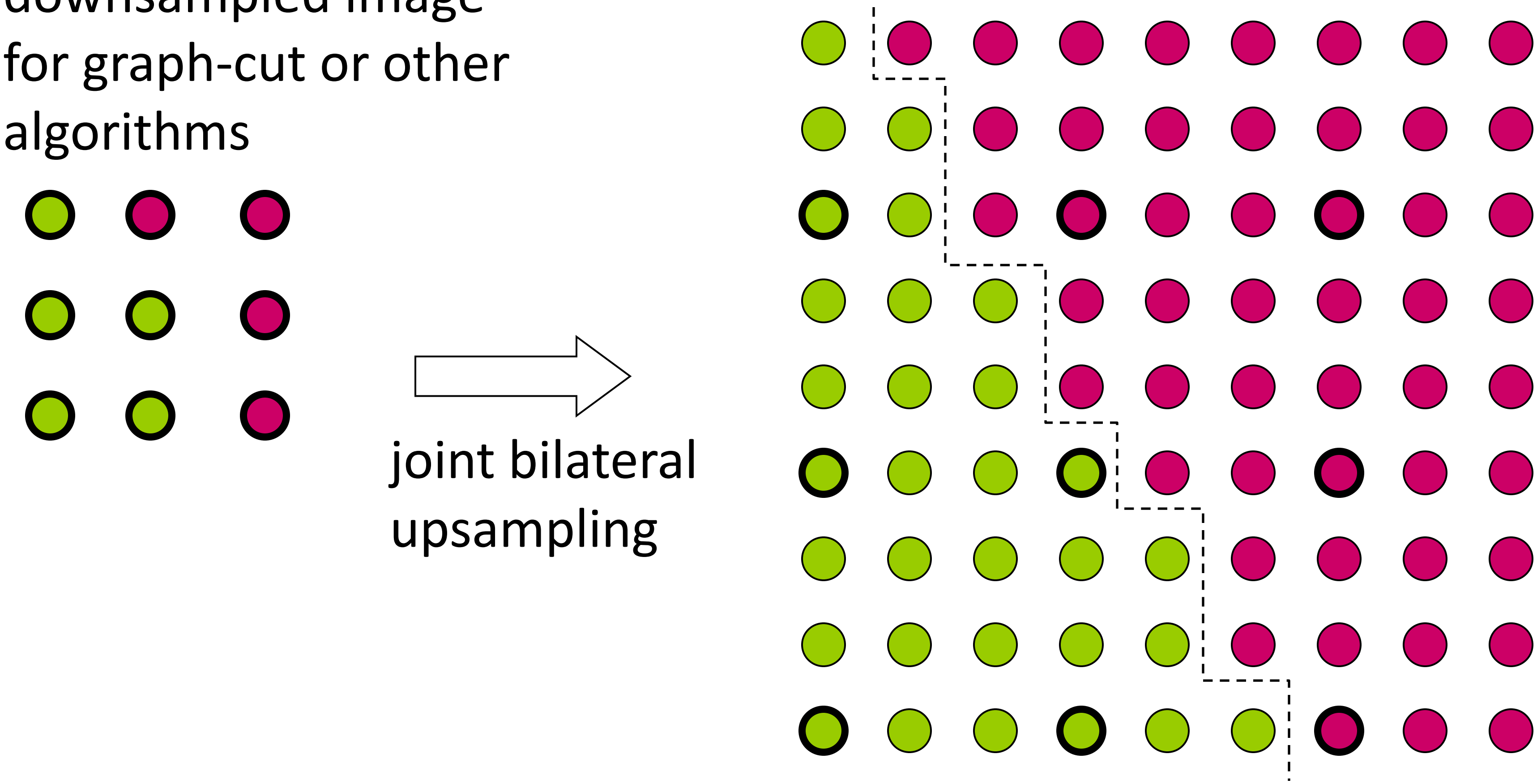
downsampled image for graph-cut or other algorithms

joint bilateral upsampling

original image

wrong boundary by direct upsampling

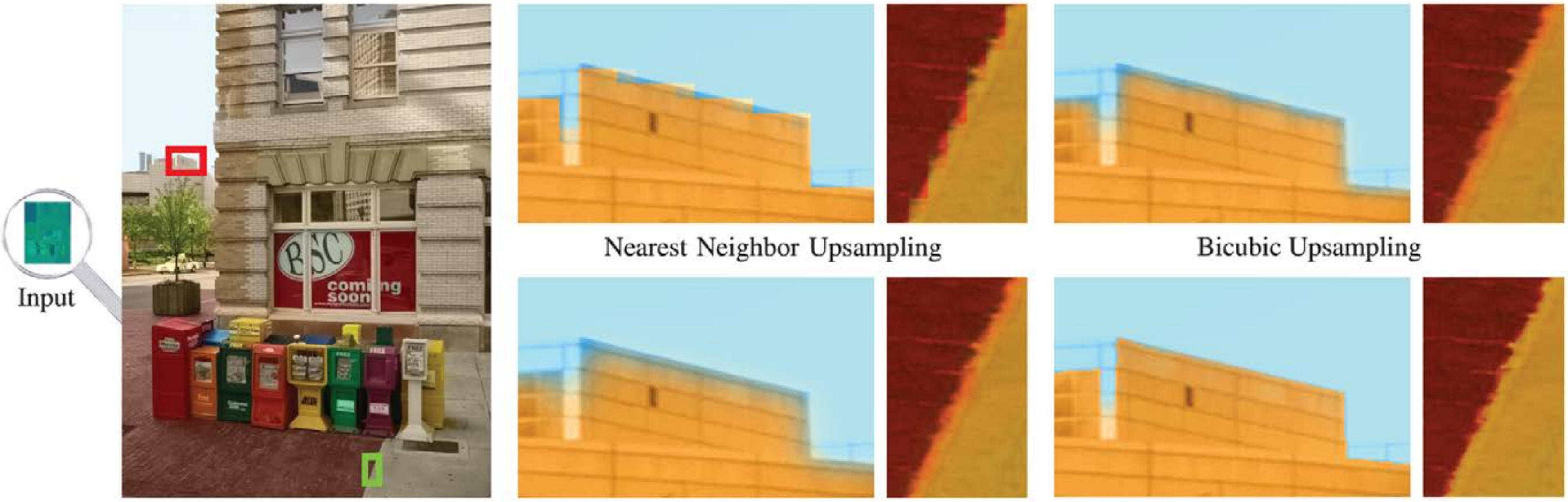
downsampled image for graph-cut or other algorithms



joint bilateral upsampling

upsampled result

Applications: Colorization



Input

Nearest Neighbor Upsampling

Bicubic Upsampling

Upsampled Result

Gaussian Upsampling

Joint Bilateral Upsampling

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

grayscale

A Simple Demo

- A few lines of code

```
I=double(imread('dog.bmp'))/255; % original image
Is=double(imread('dog_s.bmp'))/255; % low-res segmented image
Is=imresize(Is, [size(I,1), size(I,2)], 'nearest'); % enlarge
B(:,:,1) =bilateralFilter(Is(:,:,1), I(:,:,1)); % joint bilateral
B(:,:,2) =bilateralFilter(Is(:,:,2), I(:,:,2));
B(:,:,3) =bilateralFilter(Is(:,:,3), I(:,:,3));
figure; imshow(Is);
figure; imshow(B);
```