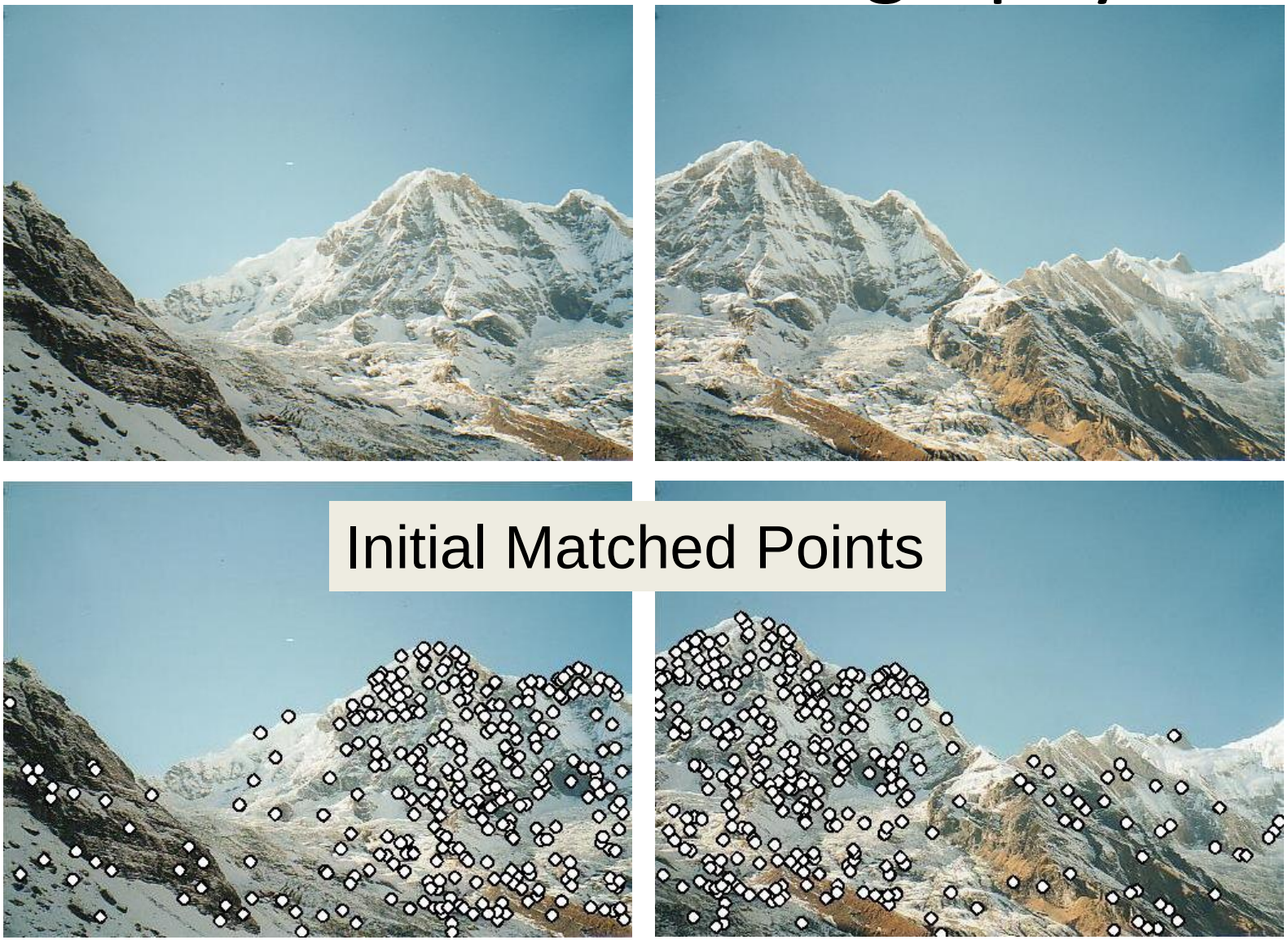
Image Stitching II

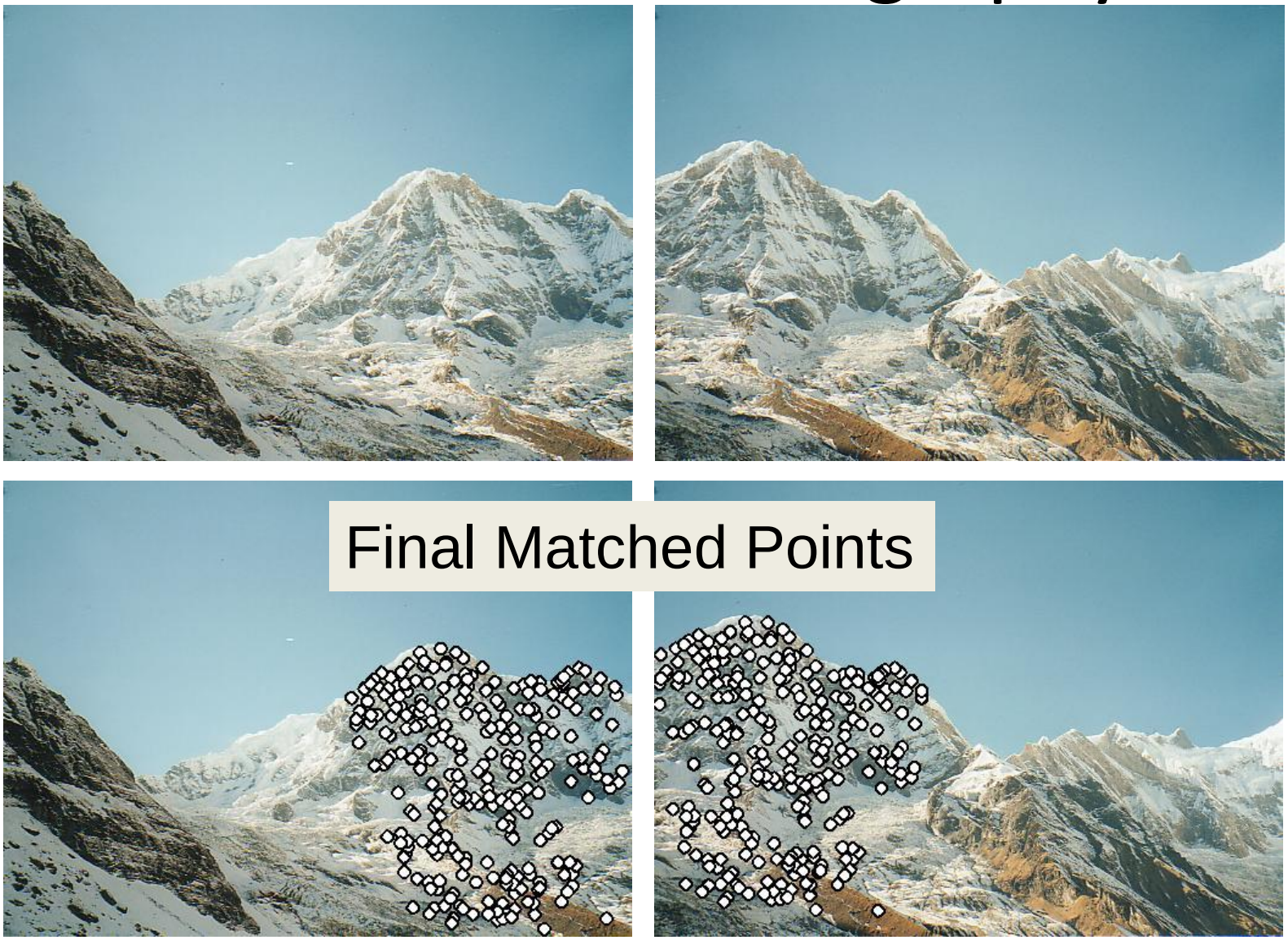
Linda Shapiro

ECE P 596

RANSAC for Homography



RANSAC for Homography



RANSAC for Homography

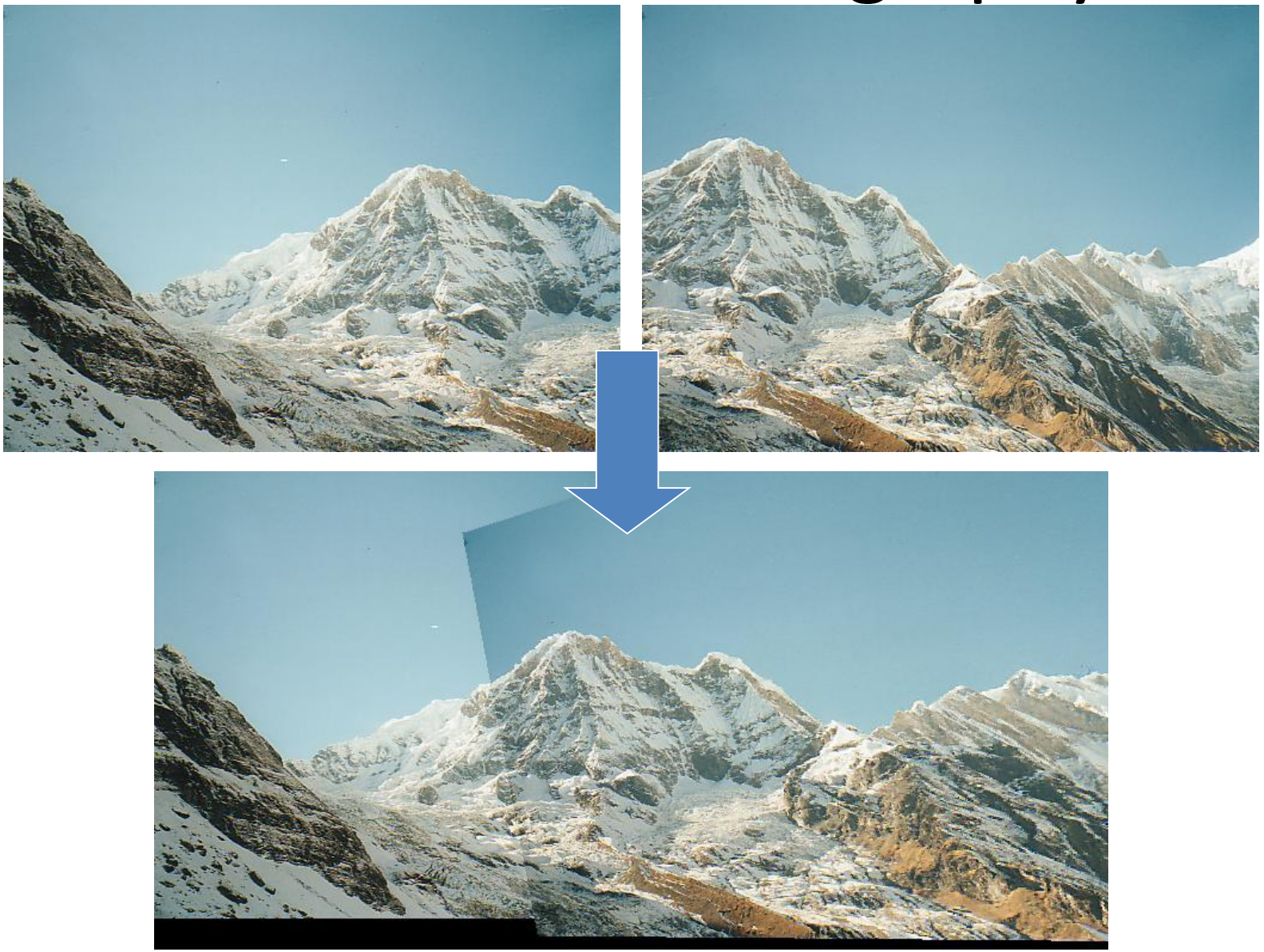
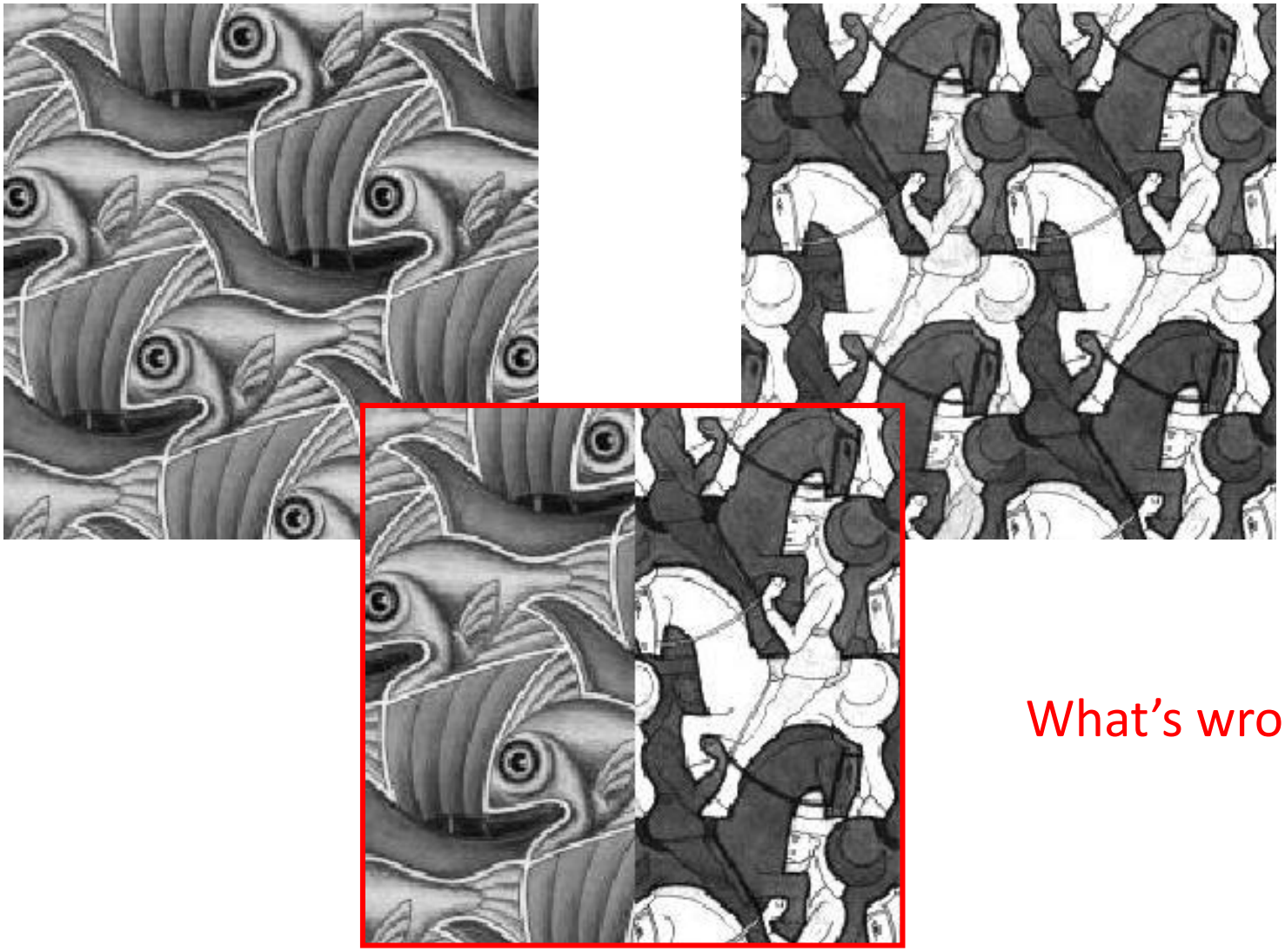
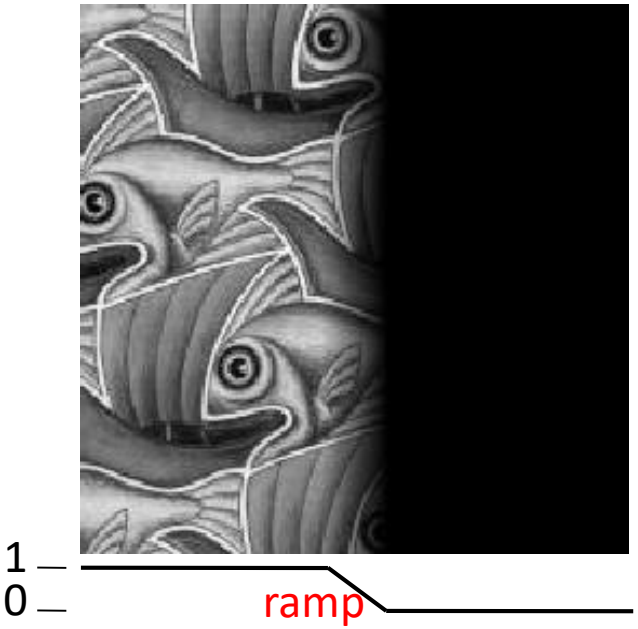


Image Blending

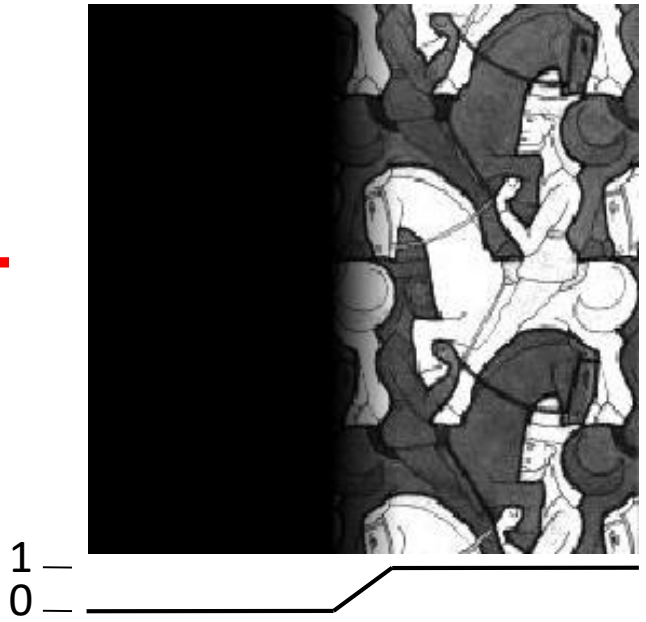


What's wrong?

Feathering

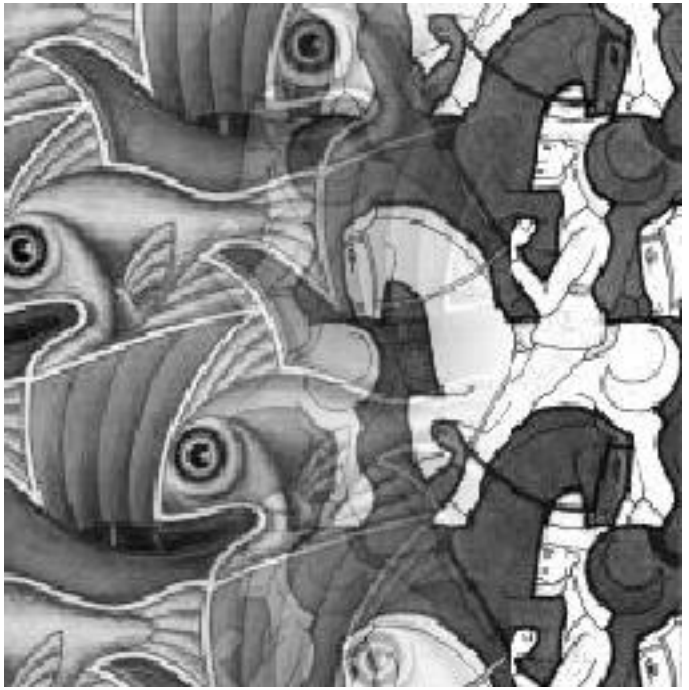


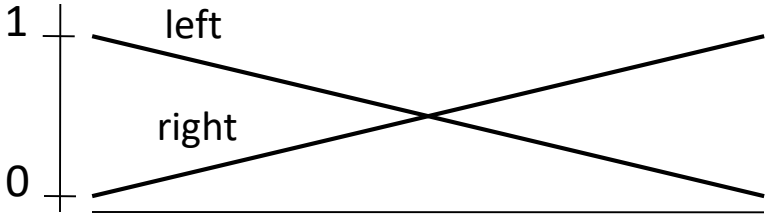
+

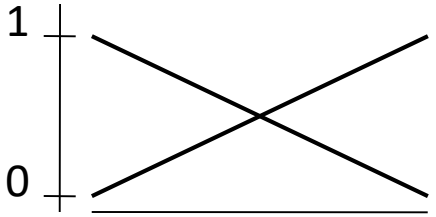


=

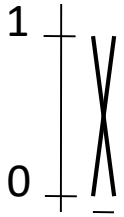
Effect of window (ramp-width) size

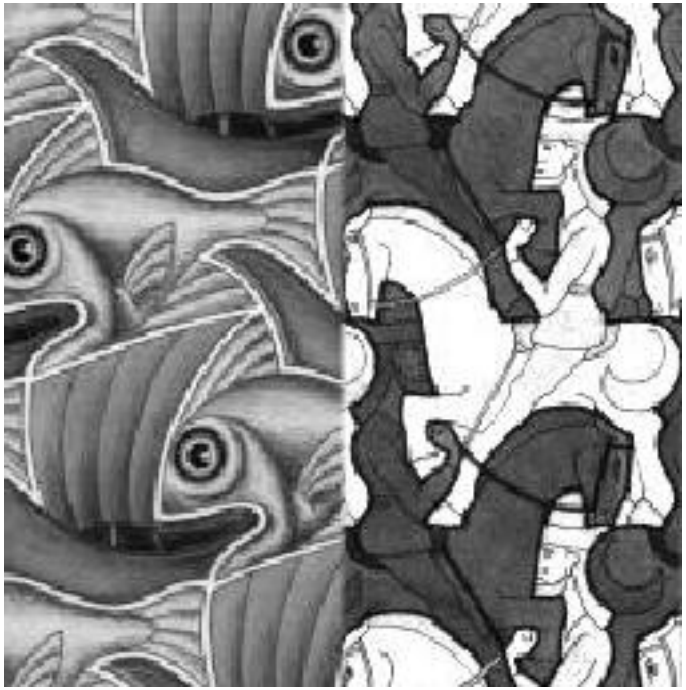


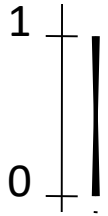




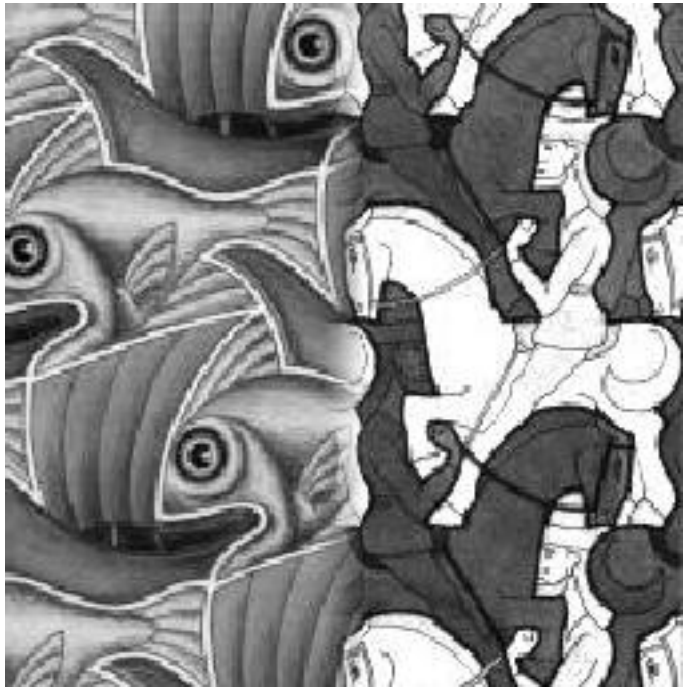
Effect of window size

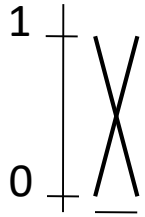






Good window size





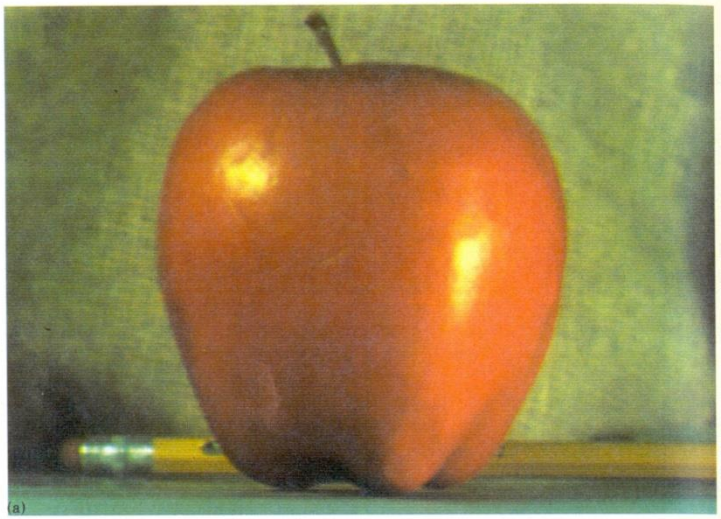
"Optimal" window: smooth but not ghosted

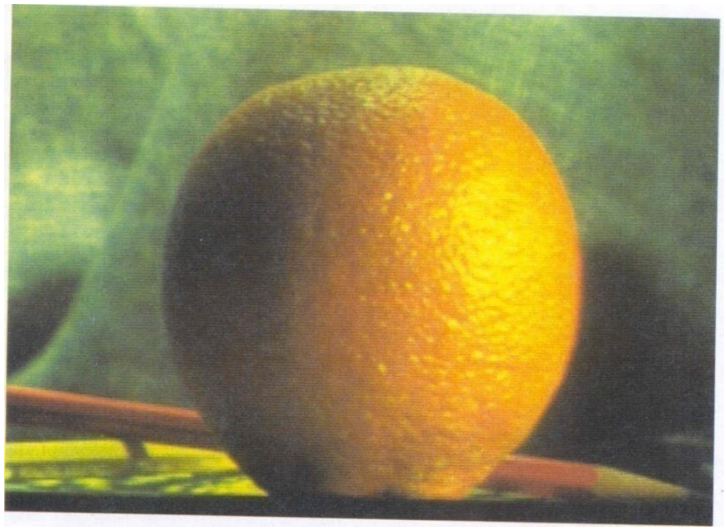
- Doesn't always work...

What can we do instead?

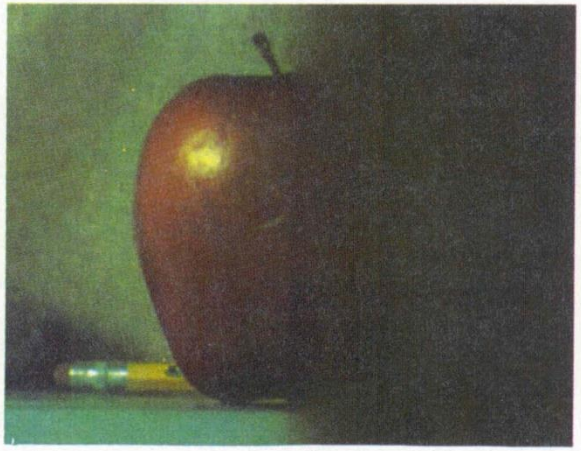
Pyramid blending

apple

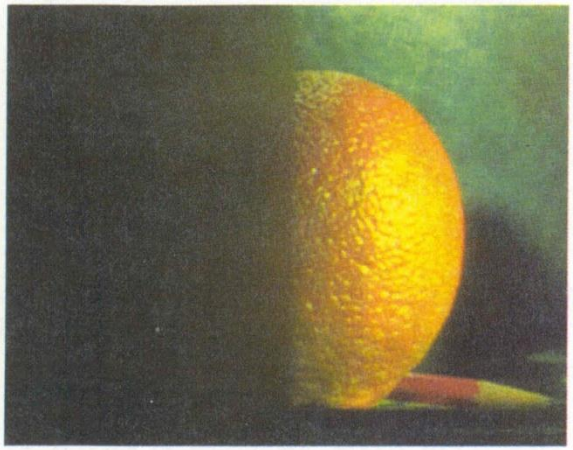




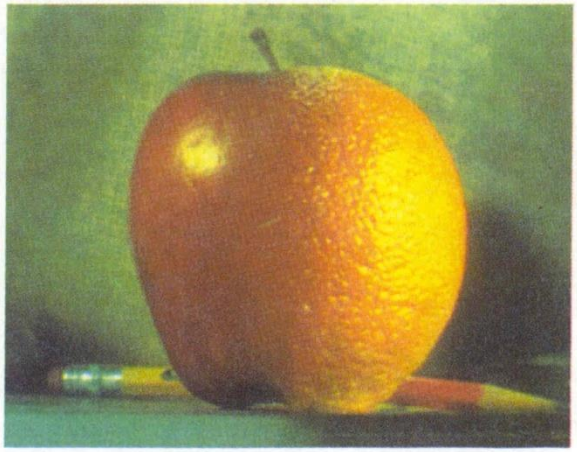
orange



(d)



(h)

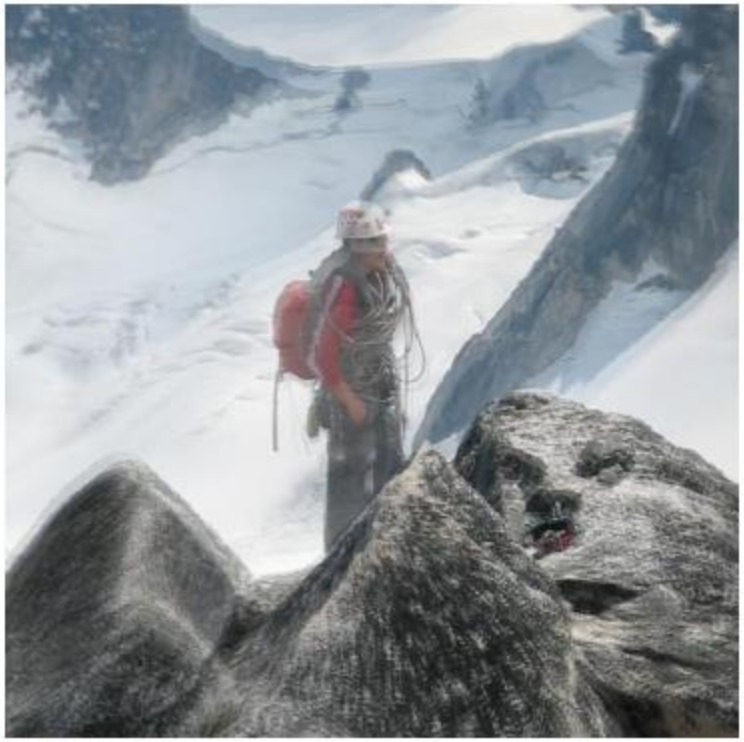


(l)

Create a Laplacian pyramid, blend each level

- Burt, P. J. and Adelson, E. H., A Multiresolution Spline with Application to Image Mosaics, ACM Transactions on Graphics, 42(4), October 1983, 217-236. http://persci.mit.edu/pub_pdfs/spline83.pdf

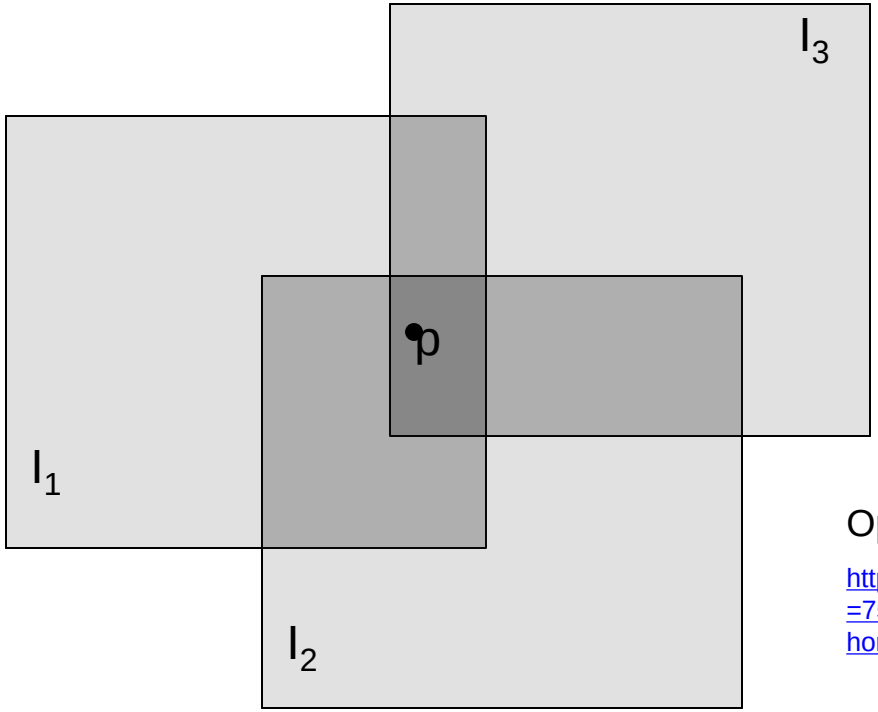
Blending comparison (IJCV 2007)



(a) Linear blending

(b) Multi-band blending

Alpha Blending



Optional: see Blinn (CGA, 1994) for details:

http://ieeexplore.ieee.org/iel1/38/7531/00310740.pdf?isNumber=7531&prod=JNL&arnumber=310740&arSt=83&ared=87&arAuthor=Blinn%2C+J.F.

Encoding blend weights: I(x,y) = (αR, αG, αB, α)

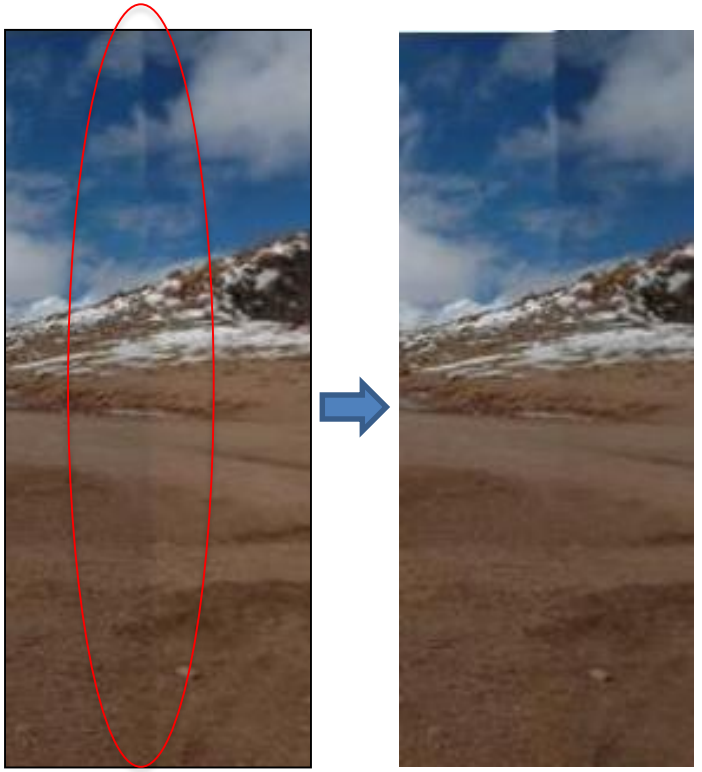
color at p = $$\frac{(\alpha_1 R_1,\ \alpha_1 G_1,\ \alpha_1 B_1) + (\alpha_2 R_2,\ \alpha_2 G_2,\ \alpha_2 B_2) + (\alpha_3 R_3,\ \alpha_3 G_3,\ \alpha_3 B_3)}{\alpha_1 + \alpha_2 + \alpha_3}$$

Implement this in two steps:

1. accumulate: add up the (α premultiplied) RGB values at each pixel
2. normalize: divide each pixel's accumulated RGB by its α value

Gain Compensation: Getting rid of artifacts

- Simple gain adjustment
  - Compute average RGB intensity of each image in overlapping region
  - Normalize intensities by ratio of averages



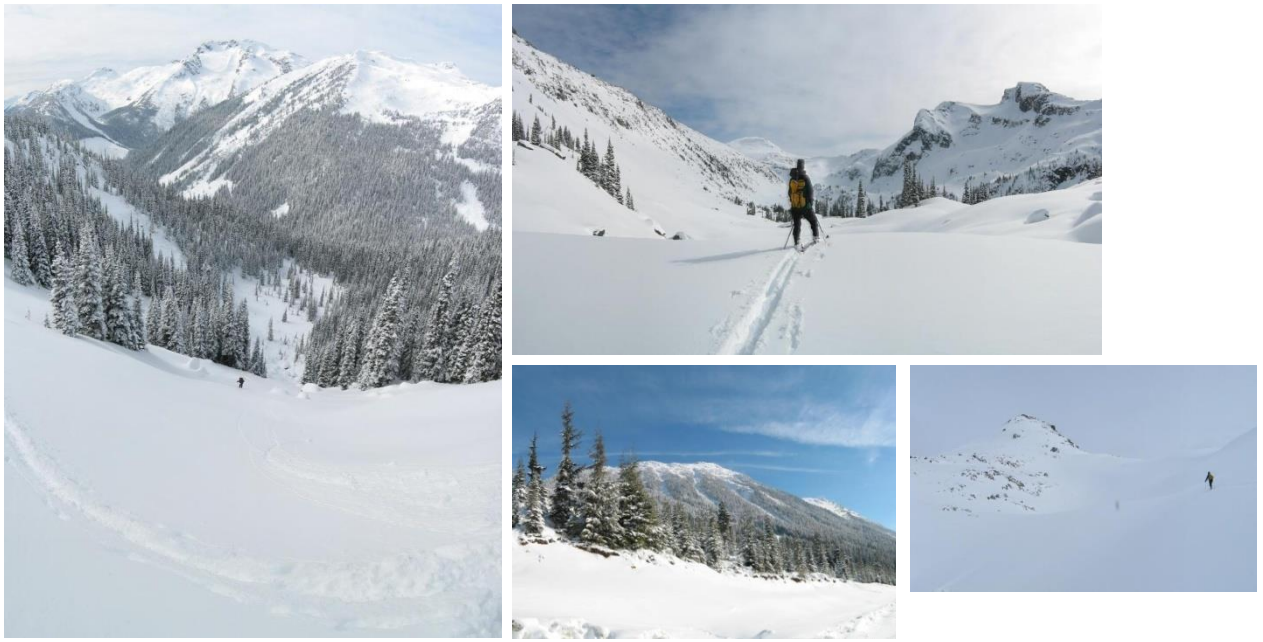
Blending Comparison

(b) Without gain compensation

(c) With gain compensation

(d) With gain compensation and multi-band blending

Recognizing Panoramas



Some of following material from Brown and Lowe 2003 talk

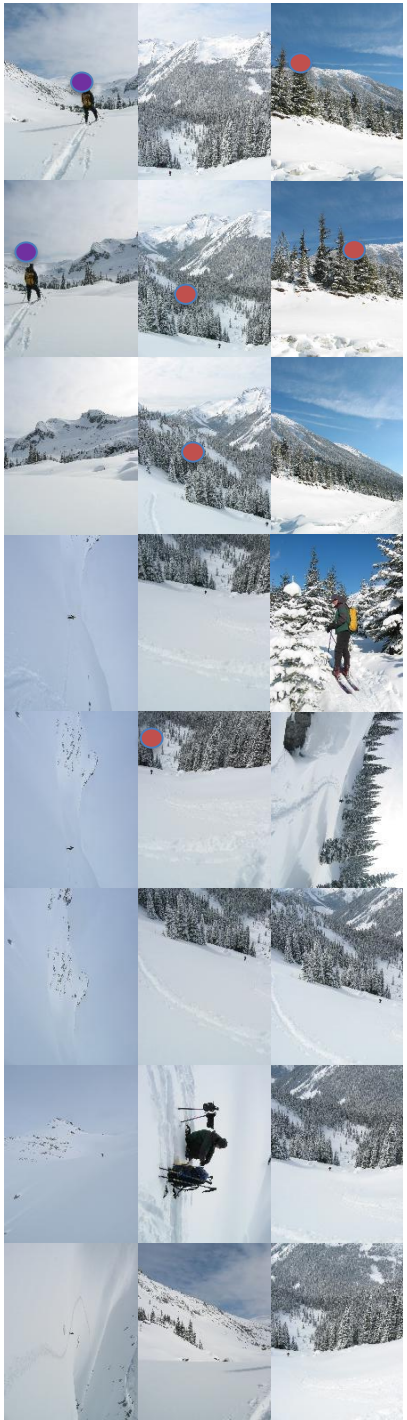
Brown and Lowe 2003, 2007

Recognizing Panoramas

Input: N images

1. Extract SIFT points, descriptors from all images
2. Find K-nearest neighbors for each point (K=4)

   (from the OTHER images)
3. For each image
   a) Select M candidate matching images by counting matched keypoints in other images (m=6)
   b) Solve homography $\mathbf{H}_{ij}$ for each matched image



What else matches?

Recognizing Panoramas

Input: N images

1. Extract SIFT points, descriptors from all images
2. Find K-nearest neighbors for each point (K=4)
3. For each image
   a) Select M candidate matching images by counting matched keypoints (m=6)
   b) Solve homography $\mathbf{H}_{ij}$ for each matched image
   c) Decide if match is valid ($n_i > 8 + 0.3\ n_f$)

n_i: # inliers

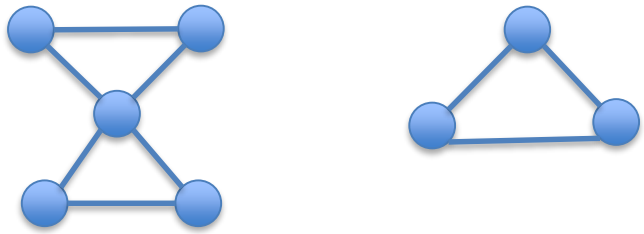
n_f: # keypoints in overlapping area

Recognizing Panoramas (cont.)

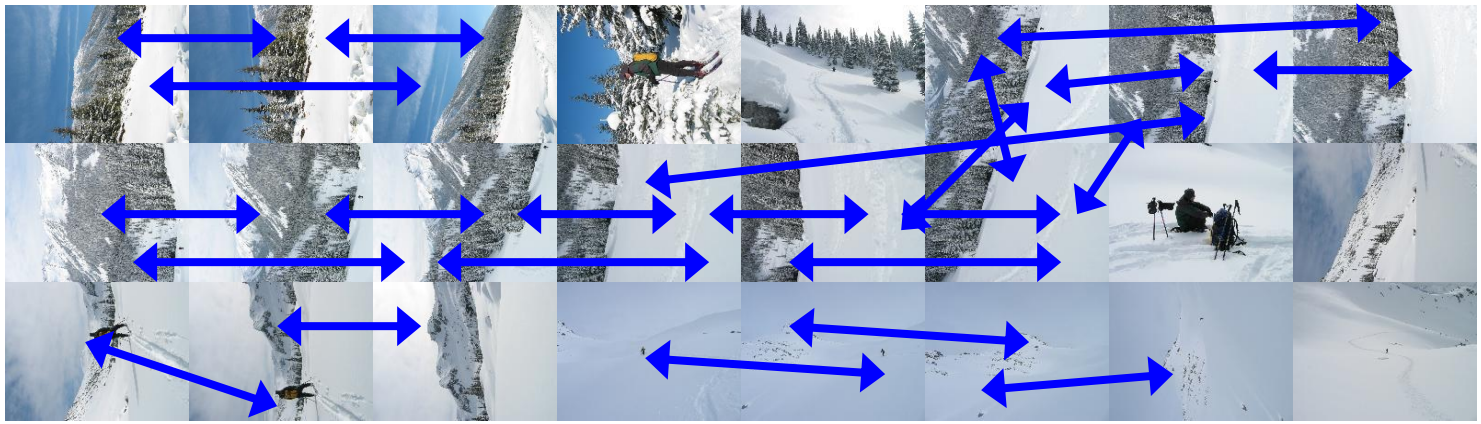
(now we have matched pairs of images)

4. Make a graph of matched pairs

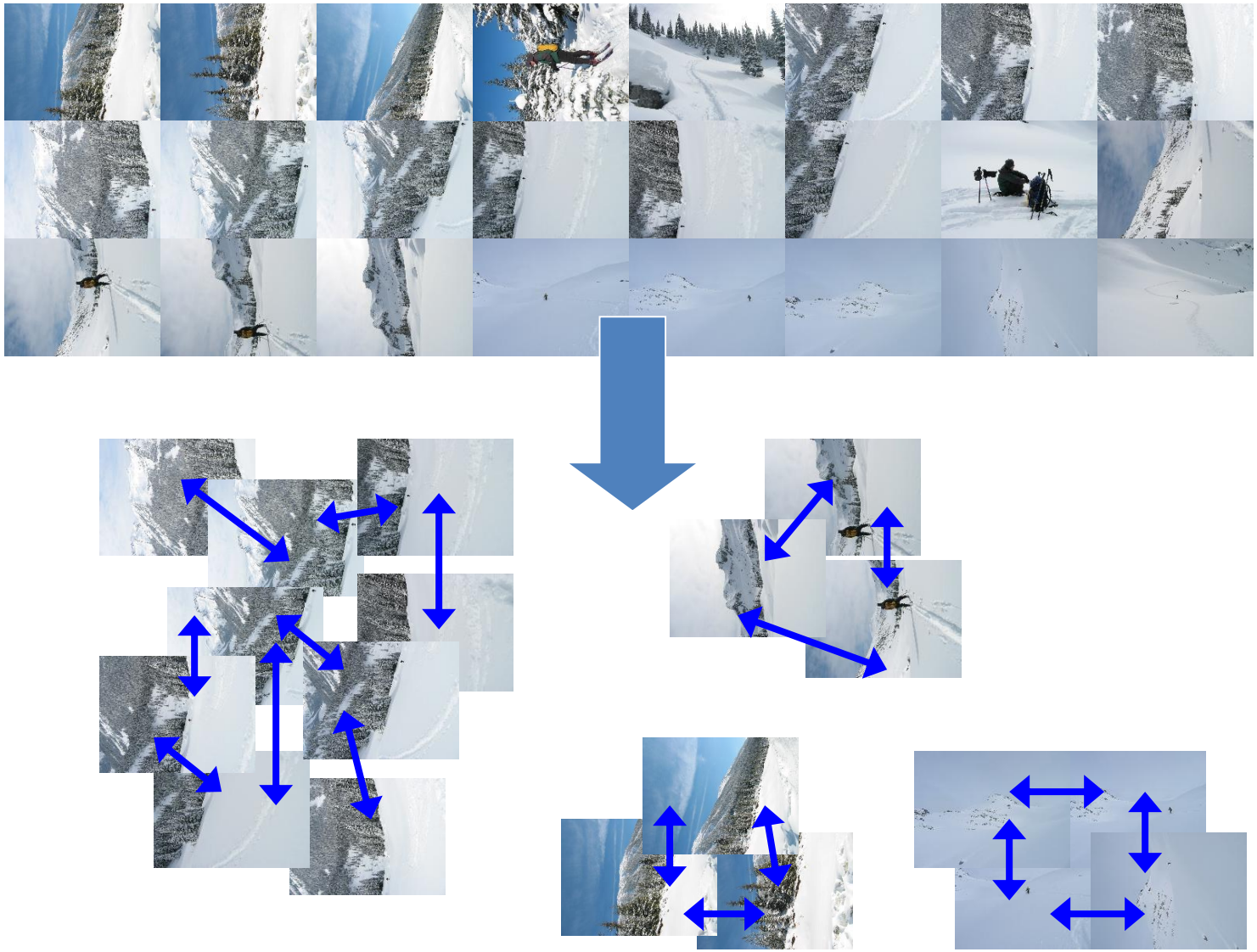
   Find connected components of the graph



Finding the panoramas



Finding the panoramas



Recognizing Panoramas (cont.)

(now we have matched pairs of images)

4. Find connected components
5. For each connected component
   a) Solve for rotation and f
   b) Project to a surface (plane, cylinder, or sphere)
   c) Render with multiband blending

Finding the panoramas