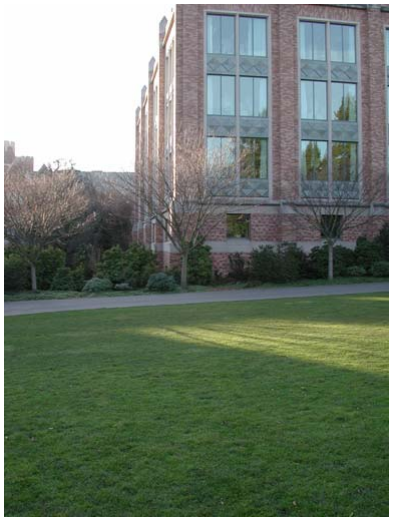
Making Panoramas

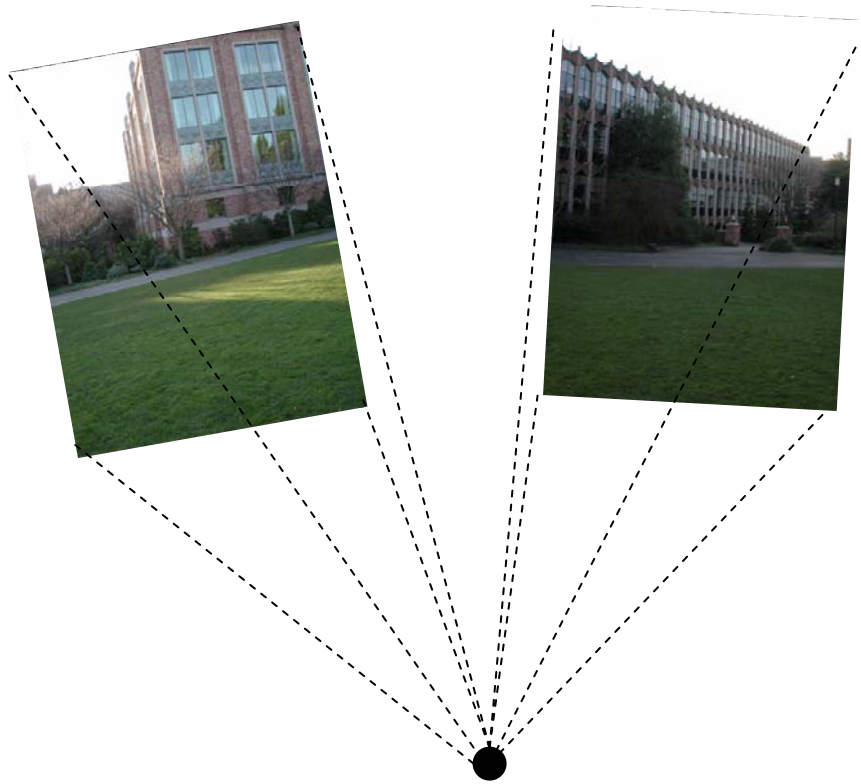
Input:

. . .

Output:

- Input:
  - A set of images taken from the same optical center.
  - For this project, the images will also have the same horizontal orientation.

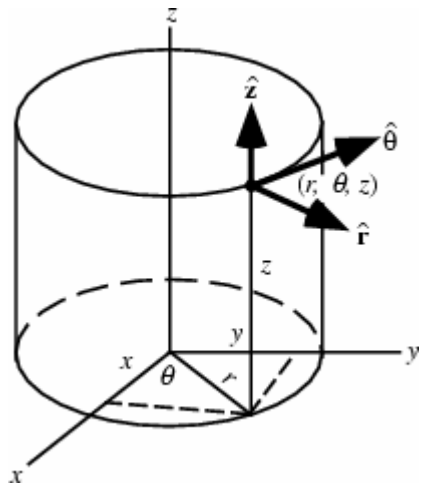


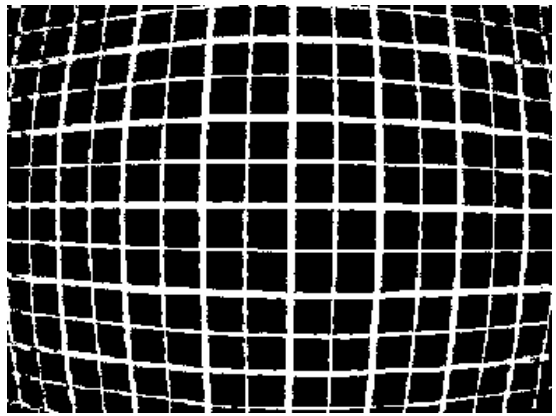
Steps

1. Convert each image to cylindrical coordinates.
    a. Remove radial distortion.
2. Find the alignment parameters between each adjacent pair of images.
3. Blend the images into a single panorama.

1. Convert each image to cylindrical coordinates.

- The image plane is $z = 1$.
- We compute the inverse transformation of $(\theta, y, 1)$ onto the image plane:
  - $x = \tan \theta$
  - $y = y / \cos \theta$
  - $z = 1$



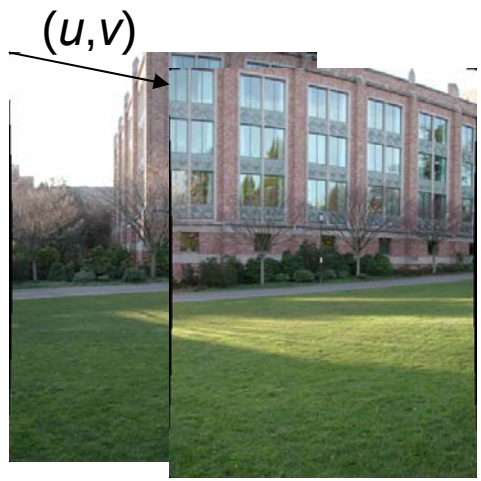


a. Remove radial distortion.

- Again, perform the inverse transformation on (x,y):
  - $x' = x + \kappa_1 r^2 x + \kappa_2 r^4 x$
  - $y' = y + \kappa_1 r^2 y + \kappa_2 r^4 y$

2. Find the alignment parameters between each adjacent pair of images.

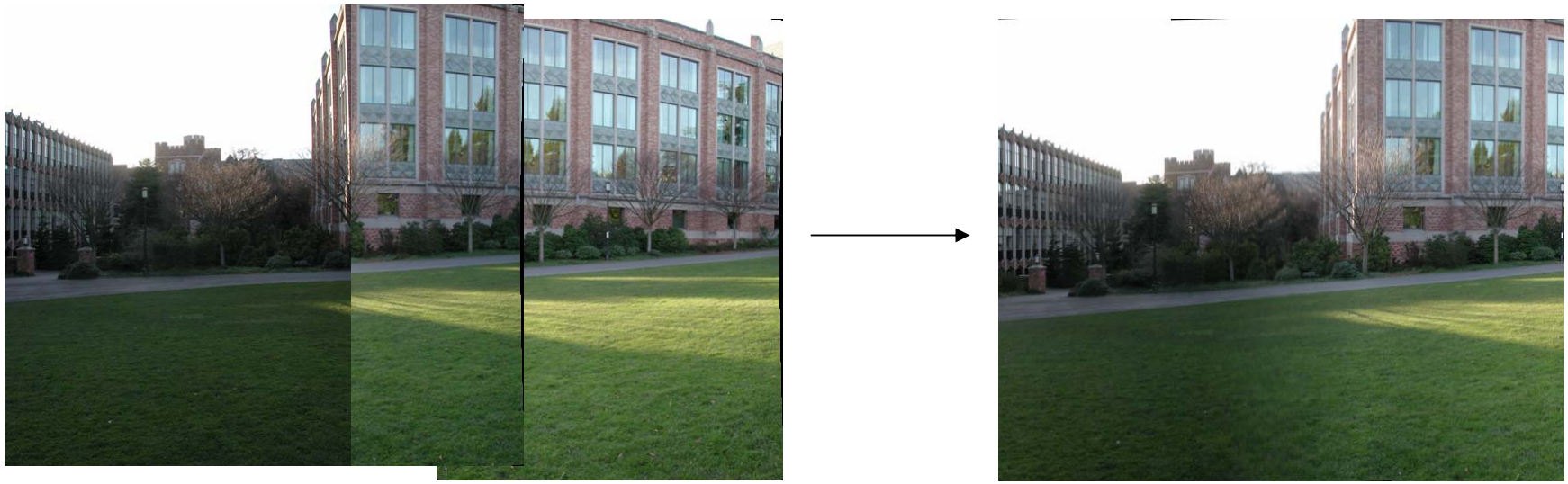
- The images lie on a cylinder and have the same horizontal orientation.
- Therefore we can represent the alignment by a single (u,v) offset.



2. Find the alignment parameters between each adjacent pair of images.

- The Lucas-Kanade optical flow algorithm can discover this offset.
  - $I_x u + I_y v = -It$ at each pixel.
  - We have two unknowns and many equations, so we can solve this with a 2-by-2 least-squares system.
- We do this for each level of the image pyramid, traversing coarse-to-fine.

3. Blend the images into a single panorama.



- What do we do with pixels shared by multiple images?

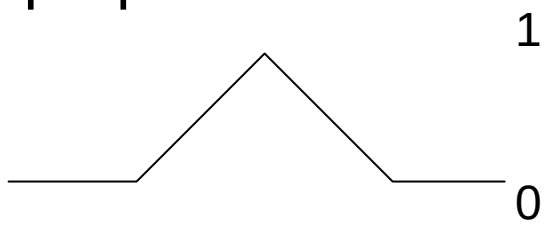
3. Blend the images into a single panorama.

- Have each image *i* assign a weight α_i to each pixel. Then, the color of a pixel (*r*,*g*,*b*) in the panorama is:

$$(r, g, b) = \frac{\sum_i \alpha_i (r_i, g_i, b_i)}{\sum_i a_i}$$

3. Blend the images into a single panorama.

- Assigning weights:
  i. Uniform weights ($\alpha_i = 1$ for all i).
  ii. Horizontal hat function (α_i inversely proportional to distance from horizontal center, within some window.



  iii. Something else?

Project 4

- For Project 4, you will make a panorama.
- Most of the code has been given to you. You only need to write code for:
  - Forming and solving the matrix equation in Lucas-Kanade flow estimation.
  - Image blending.
- This project should be less time-consuming than the last.