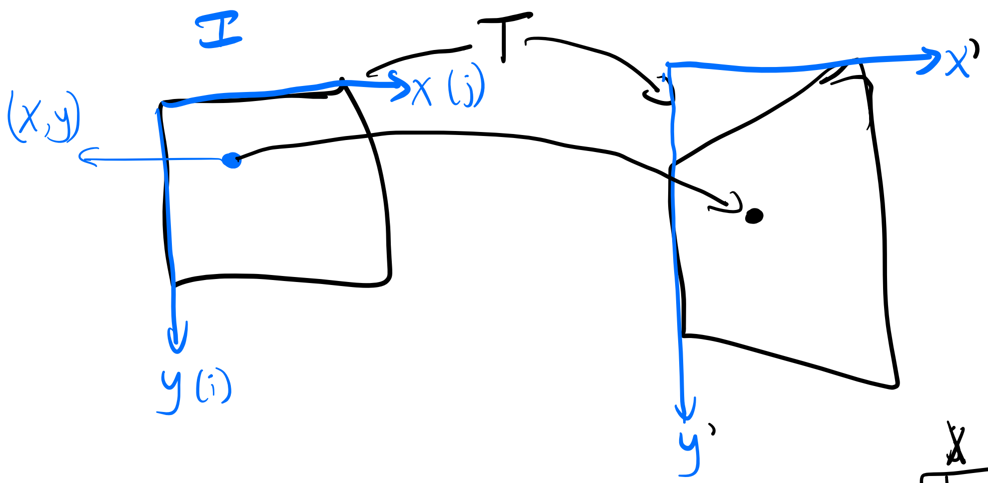
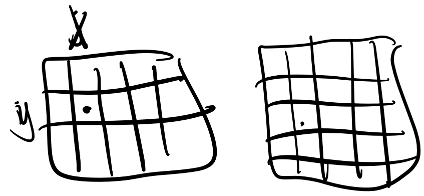
Lecture 10 - Warping

$$T\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \rightarrow \begin{matrix} x' \\ y' \\ 1 \end{matrix}$$



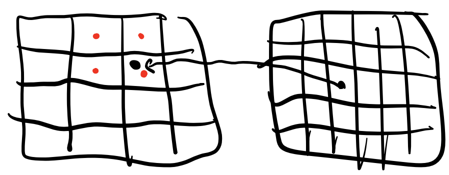
Code: forward warping:




```
for (x, y) in I:
    x', y' ← T(x,y)
    I'[x', y'] = I[x,y]
```

type(x')

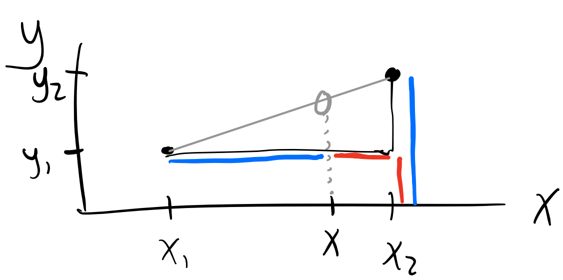
Inverse Warping:



```
for (x', y') in I':
    x, y ← T^-1(x', y')
    I'[x', y'] ← interpolate(I, x, y)
```

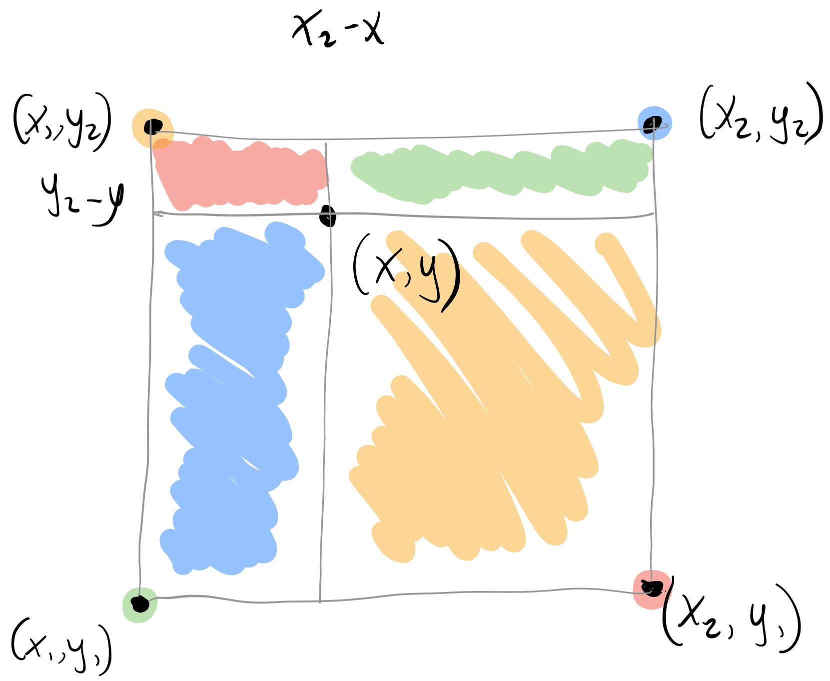
Linear Interpolation

$$y = y_1(x_2 - x) + y_2(x - x_1)$$

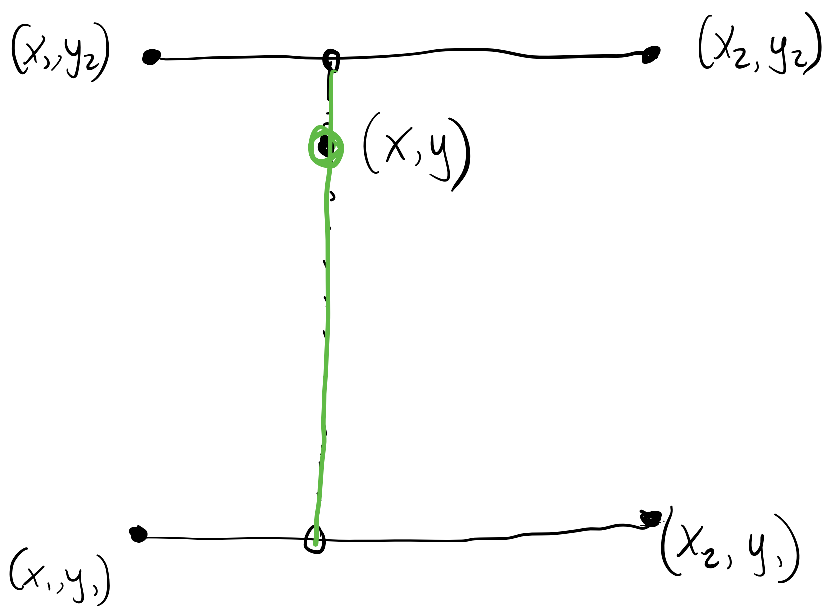


Bilinear Interpolation

- A tent filter
- Weights are areas of opposite-corner rectangles



- Interpolate in x, then interpolate in y



Panorama Stitching

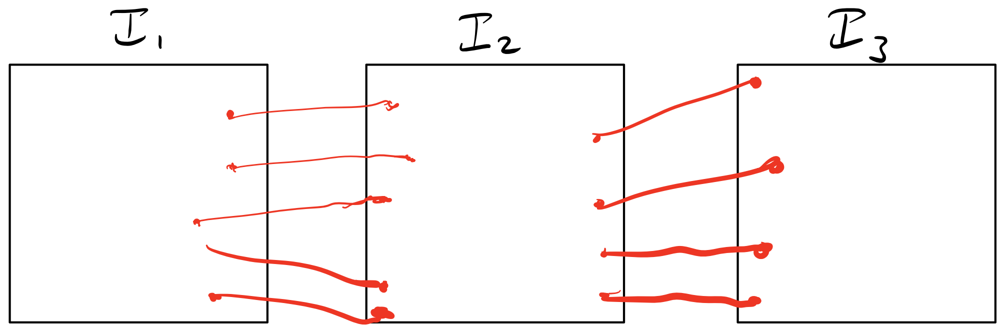
We have most of the pieces now!

CAPTURE

1. Capture images with some overlap.

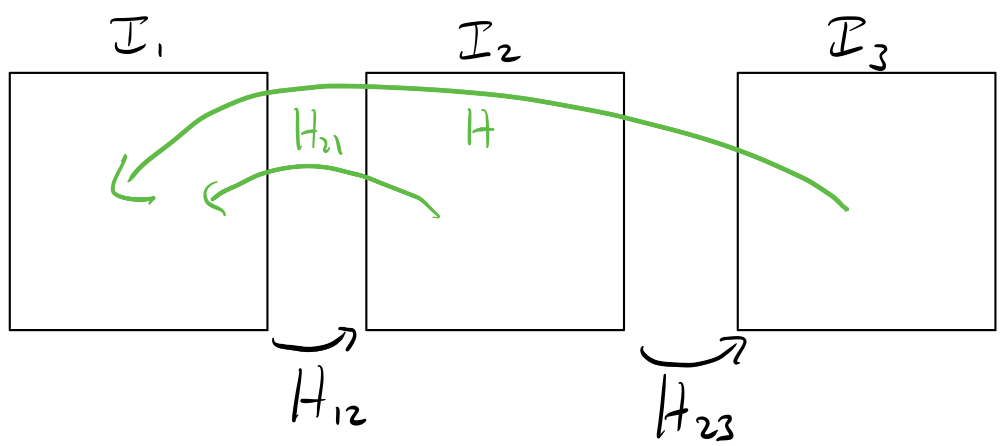
2. Detect, describe, and match features, yielding a set of feature correspondences between neighboring pairs.

MATCH



3. Fit a transformation to align neighboring pairs of images.

ALIGN PAIRWISE



4. Compute each image's transform to the coordinates of some reference image (here: I_1)

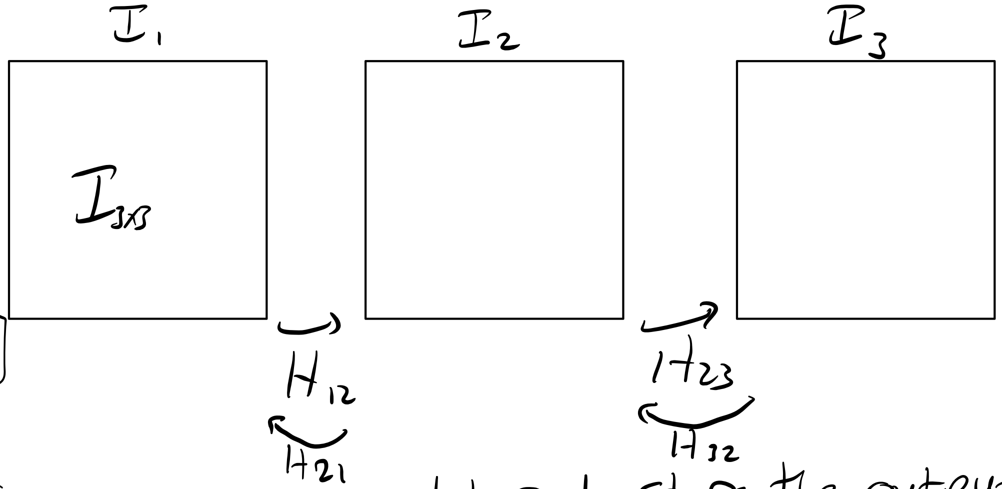
ALIGN GLOBAL

HW #2:

Given H_{12} and H_{23}, find $\underline{H_{31}}$?

$H_{21} H_{32} \begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix}$

$H_{12}^{-1} H_{23}^{-1}$



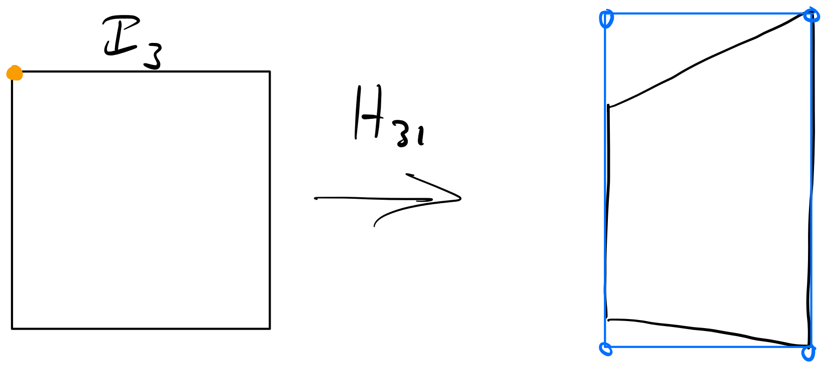
5. Create an accumulator to store the output panorama: Needs to fit <u>all</u> warped images.

SET UP OUTPUT IMAGE

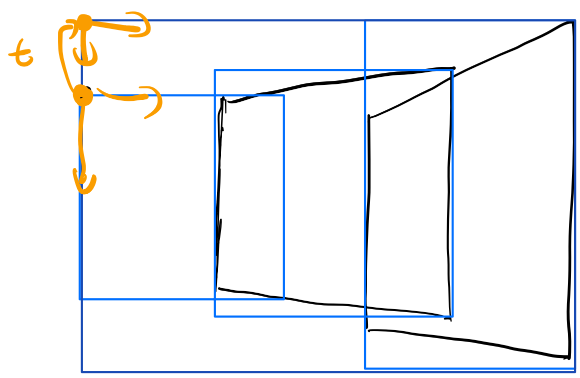
5a. Compute the bounding box of each warped image.

HW #3:

What are the corners of the bounding box of I_3 after it is warped with H_{31}?



5b. Bounding box of all the bounding boxes.



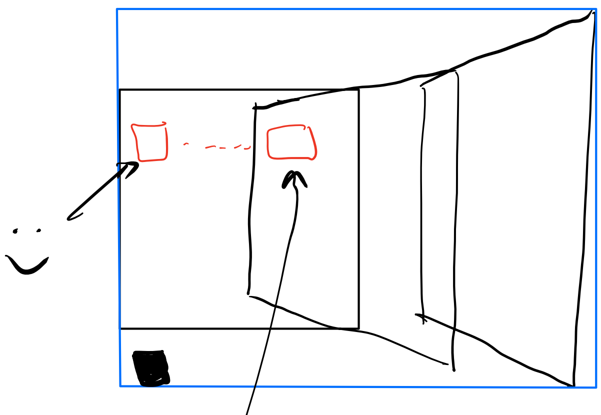
5c. Adjust transformations to put the accumulator origin in the top left.

WARP

use inverse warping

6. Warp each image and add it into acc

```
for each img
    for (y',x') in acc:
        acc[y',x'] = img[y,x]
```



2 images!

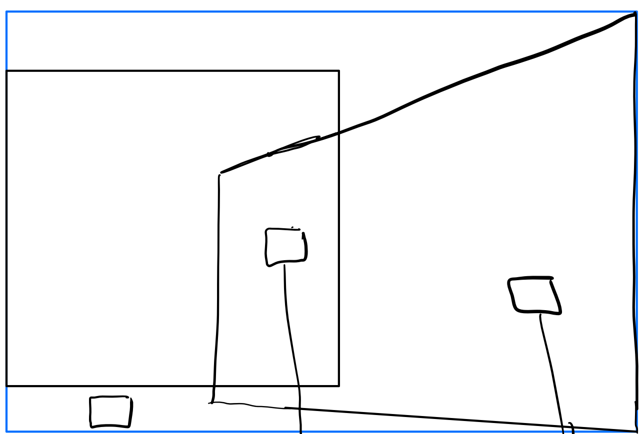
BLEND

6a. Accumulate weighted pixel contributions.

Store weight in 4th channel of input and acc.

Input Pixel:

r	g	b	1



Same as above, but:

$acc(y,x,:) \boxed{+=} interp(I,ix,iy)$

Output pixels: $\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$ $\begin{bmatrix}r_2+r_1\\g_2+g_1\\b_2+b_1\\2\end{bmatrix}$ $\begin{bmatrix}r_3\\g_3\\b_3\\1\end{bmatrix}$

6b. Normalize: divide each pixel by its total weight.

$a[a>0] += 10$

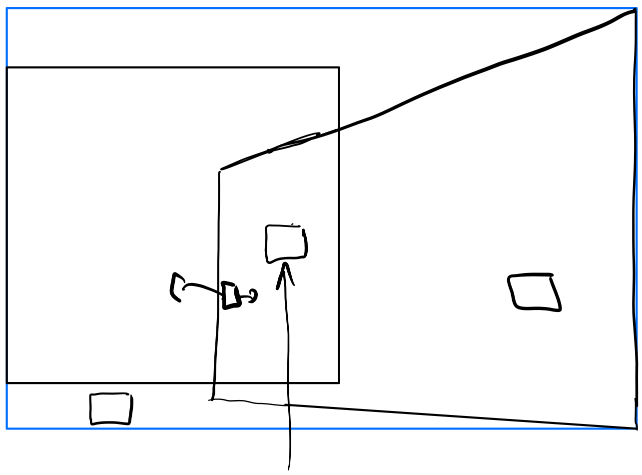
NP Magic!

$acc /= acc[:,:,3]$

be careful!

6' (Refinement): Hide the seams!

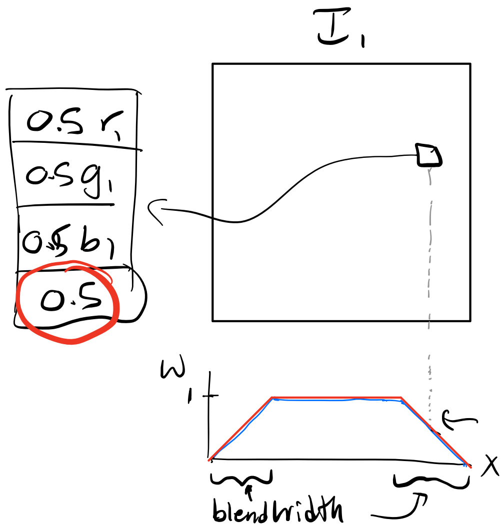
BLEND w/ FEATHERING



Trick: weight images less near the edges.

How? Use the same 4th channel!

Inputs:



Output: