EE 569 Discussion

Zohreh Azizi

02/19/2021

Announcements

- HW2

  Due: 02/23/21 11:59PM

- HW3

  Issue: 02/14/21

  Due: 03/11/21 11:59PM

➢ *Start early*!

Announcements

Preview of Discussion Sessions:

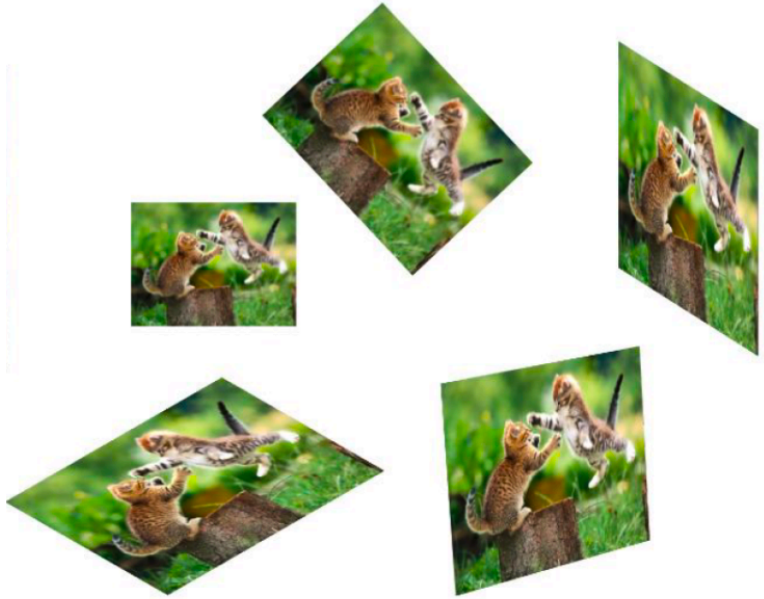
- …
- Week 6 (Feb. 19): Geometric Image Modification (HW3, P1&2)
- Week 7 (Feb. 26): Midterm 1 Review
- Week 8 (Mar. 5): Morphological Processing (HW3, P3)
- …

Overview

- Geometrics Modifications
- Image Warping (Problem 1)
- Homographic Transformation and Image Stitching (Problem 2)

Geometric Modification

- Translation
- Rotation
- Scale

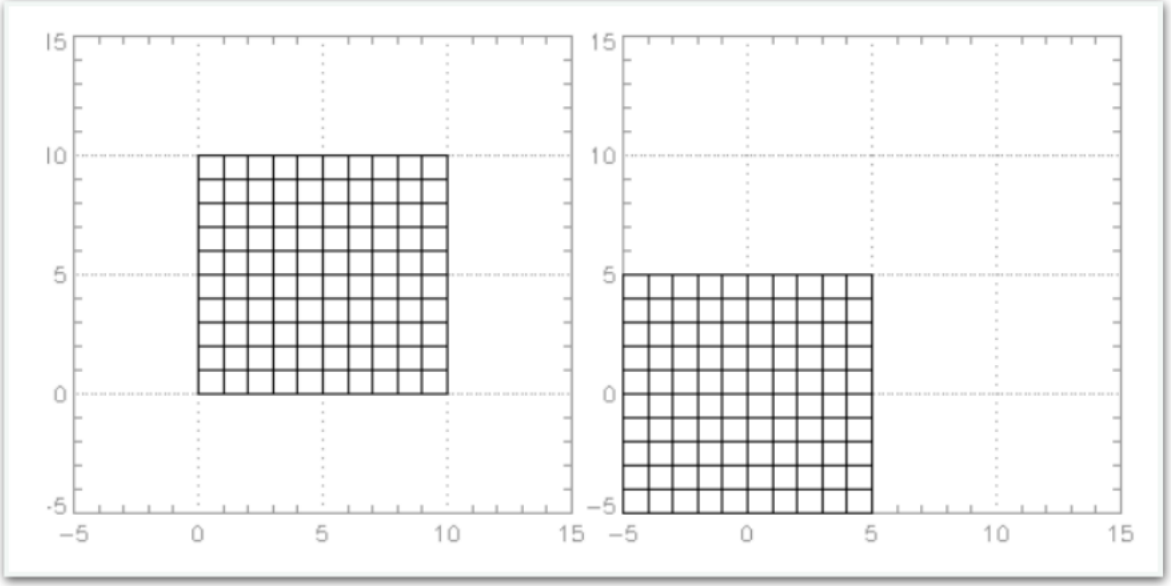


Geometric Modification

Translation

$$T = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + x_0 \\ y + y_0 \\ 1 \end{bmatrix}$$

- Preserve the orientation
- Pivot is cartesian origin

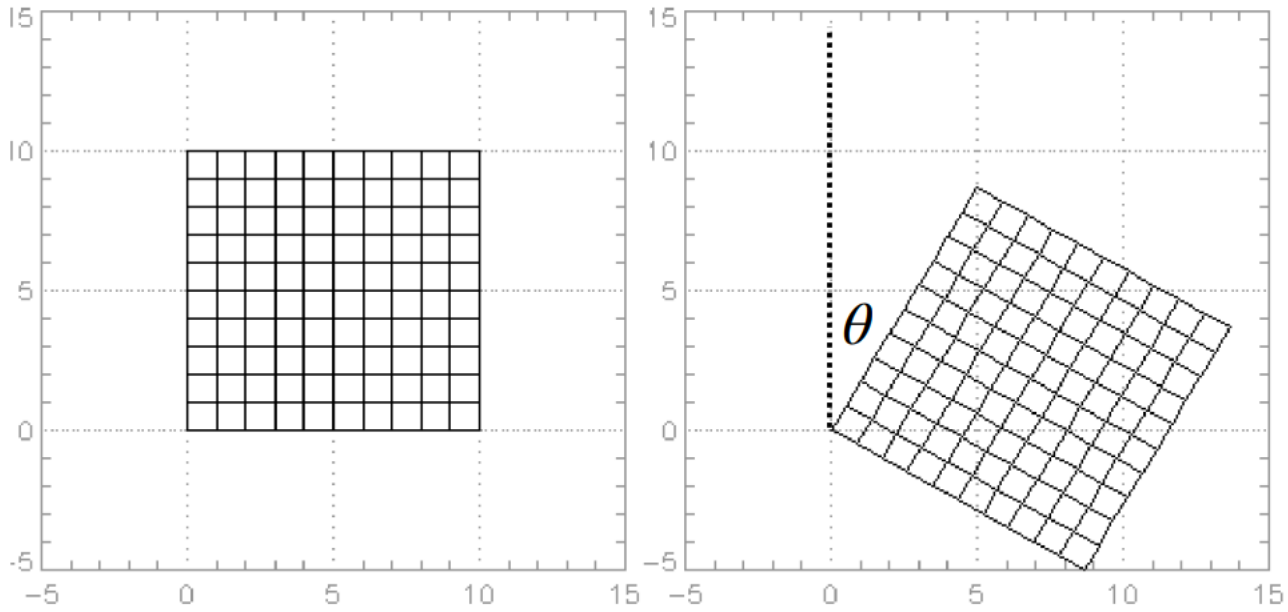


Geometric Modification

Rotation

$$T = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

- Change the orientation
- Pivot is cartesian origin

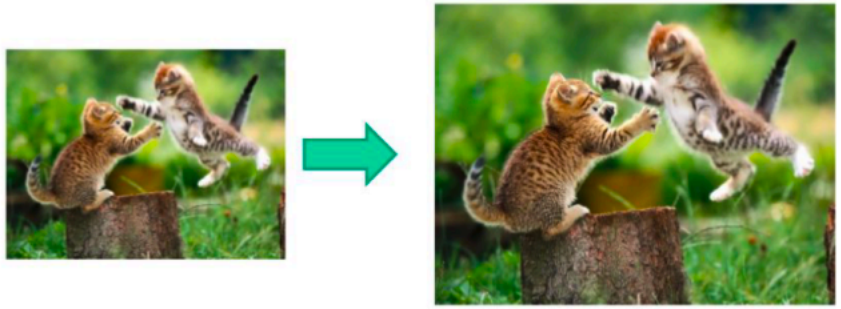


Geometric Modification

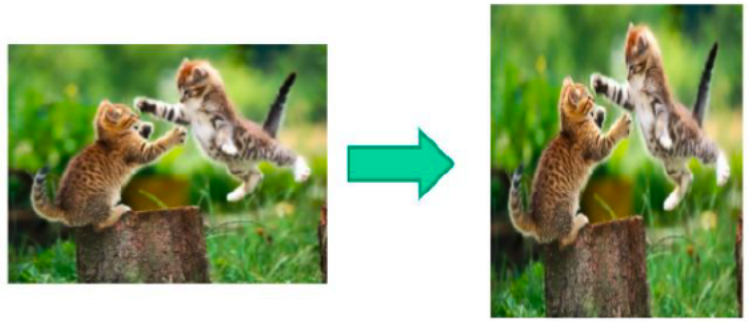
Scaling

$$T = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_1 x \\ s_2 y \\ 1 \end{bmatrix}$$

- Change aspect ratio
- Introduce distortion



Same scaling parameter



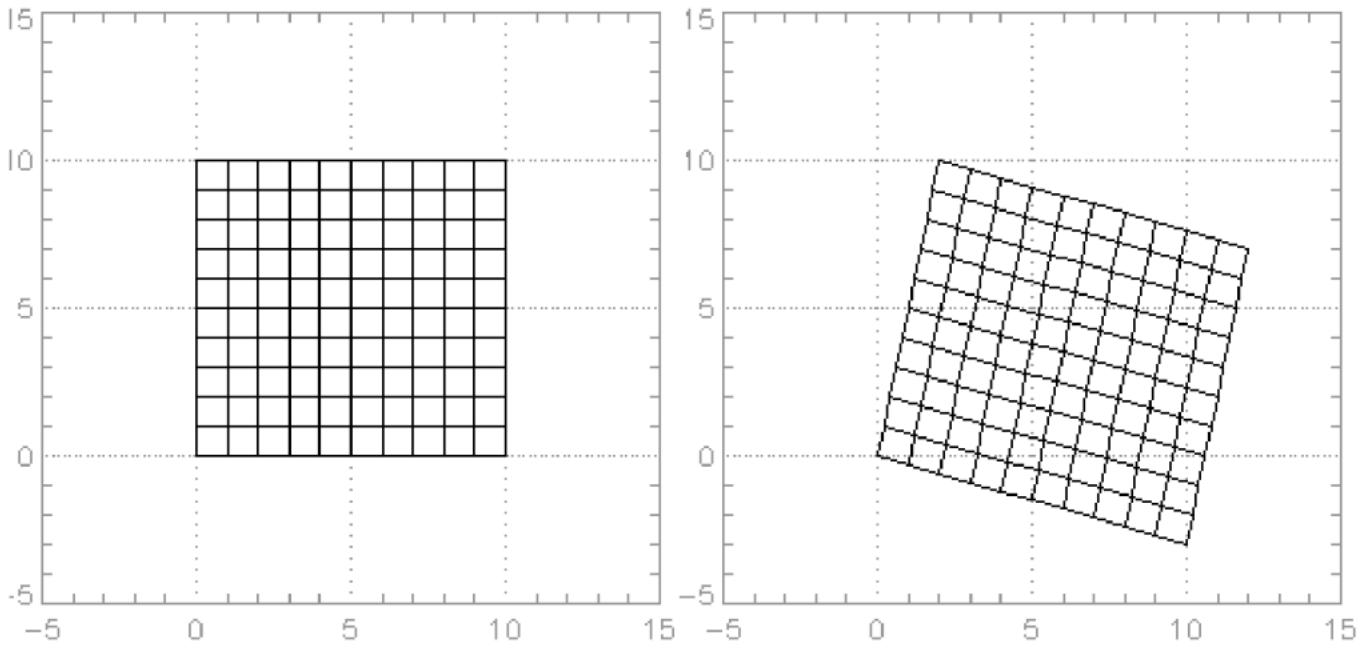
different scaling parameter

Geometric Modification

Shear

$$T = \begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

- Different from rotation
- Change all angles



Affine transformation

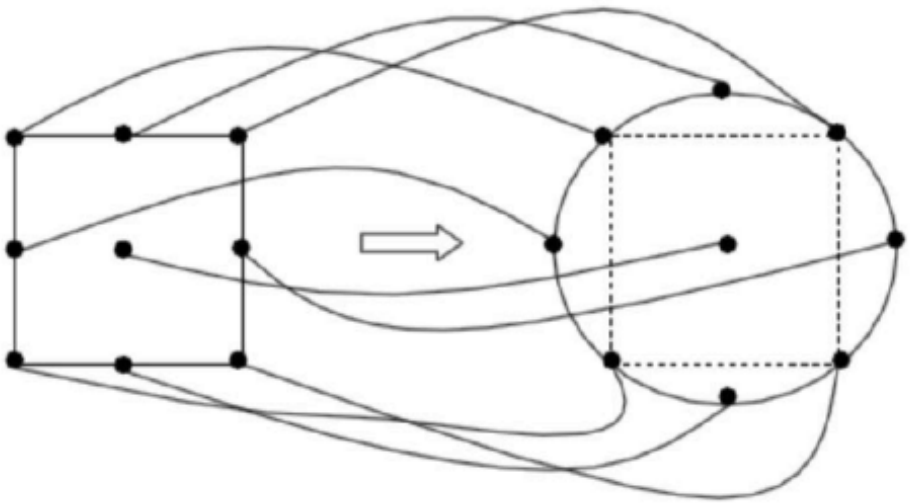
- Combination of transformations
- Translation + Scale + rotation + Shear
- Achieved by building blocks:
  - Forward Transform: $T = T_1 * T_2 * T_3$
  - Inverse Transform: $T^{-1} = T_3^{-1} * T_2^{-1} * T_1^{-1}$

$$T_1 = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}, T_2 = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, T_3 = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$, 6 Degrees of freedom

Image Warping

Warped image should satisfy 3 requirements:

1. Boundary → Boundary
2. Center → Center
3. Mapping should be reversible.



- It doesn't have just one answer. See this example (different from your problem):

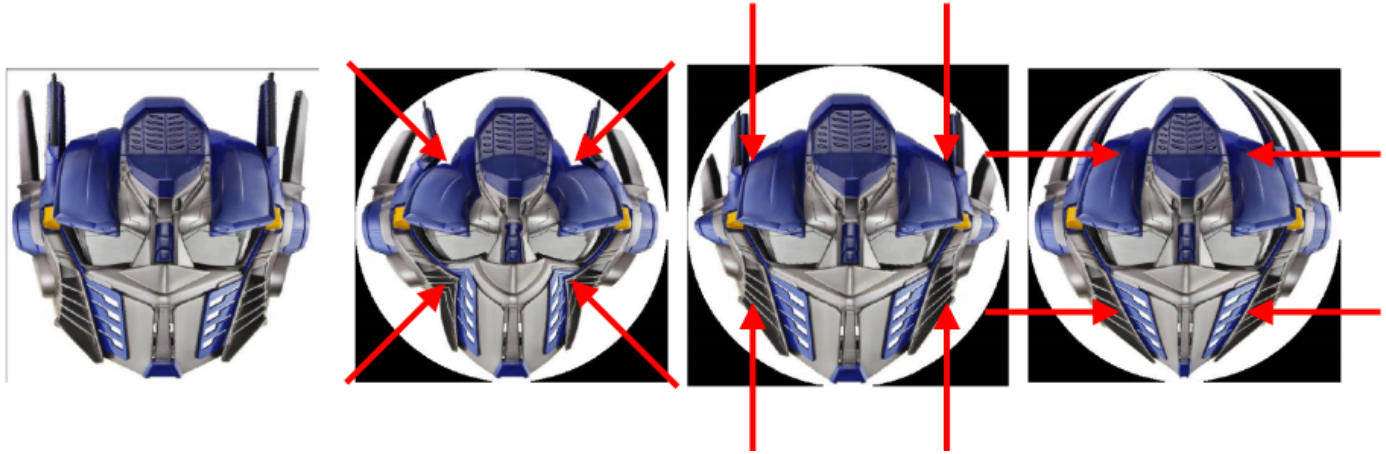
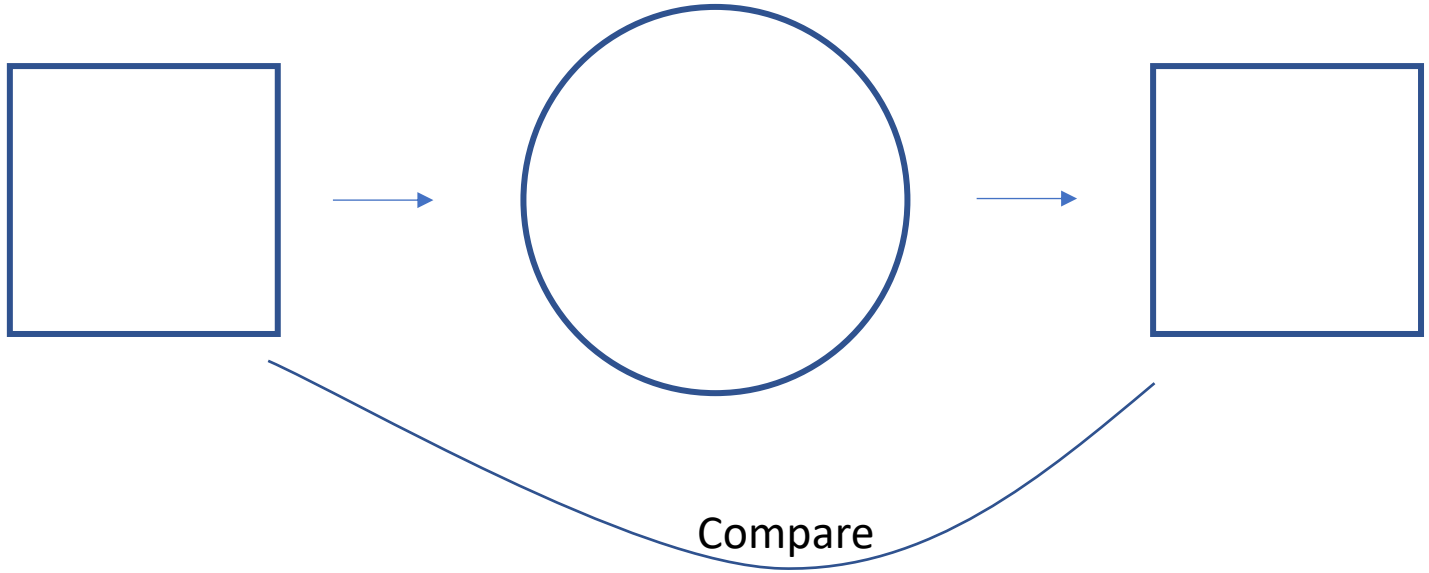


Image Warping



For geometric modification problems:

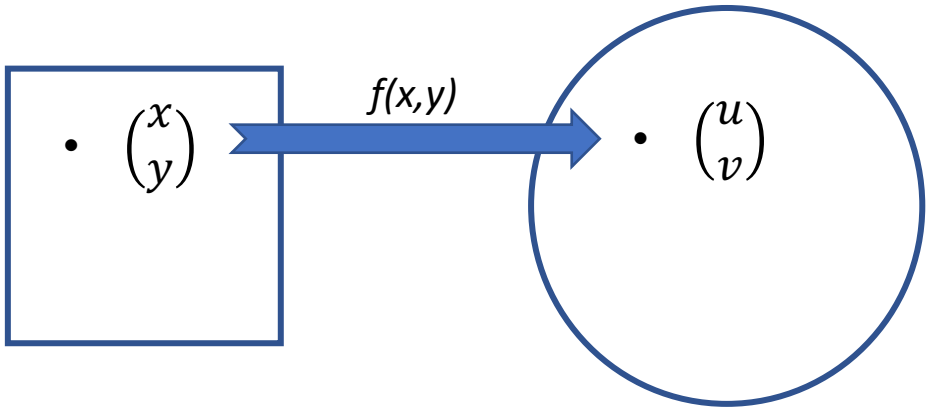
- Conversion between image and cartesian coordinates:

$$\underset{\text{Image coordinates}}{\begin{pmatrix} i \\ j \end{pmatrix}} \rightarrow \underset{\text{Cartesian coordinates}}{\begin{pmatrix} x \\ y \end{pmatrix}} \xRightarrow{warping} \underset{\text{Cartesian coordinates}}{\begin{pmatrix} u \\ v \end{pmatrix}} \rightarrow \underset{\text{Image coordinates}}{\begin{pmatrix} p \\ q \end{pmatrix}}$$

Image Warping

- Forward address mapping:

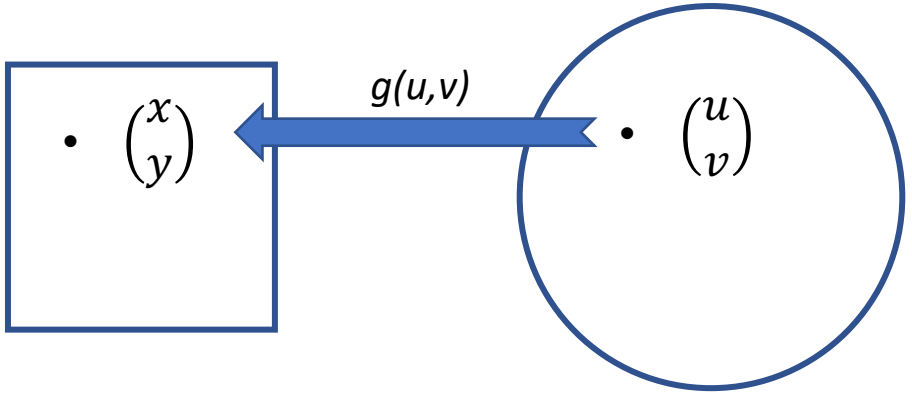
$$\begin{pmatrix} u \\ v \end{pmatrix} = f(x, y)$$



- Black dots.
- Some pixels in target image may be missed.

- Inverse address mapping:

$$\begin{pmatrix} x \\ y \end{pmatrix} = g(u, v)$$



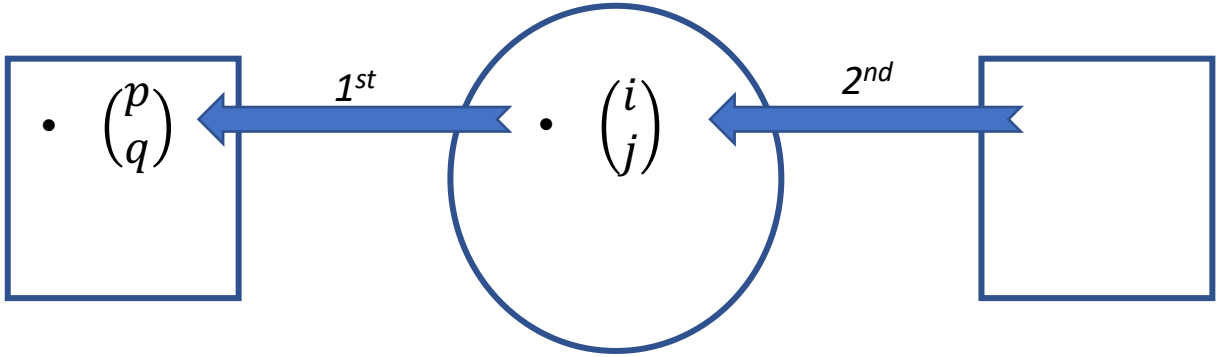
- No black dots.
- After finding *(x,y)* with *g(u,v)* function, use the value in *(x,y)* to fill in *(u,v)* in the target image.

Image Warping

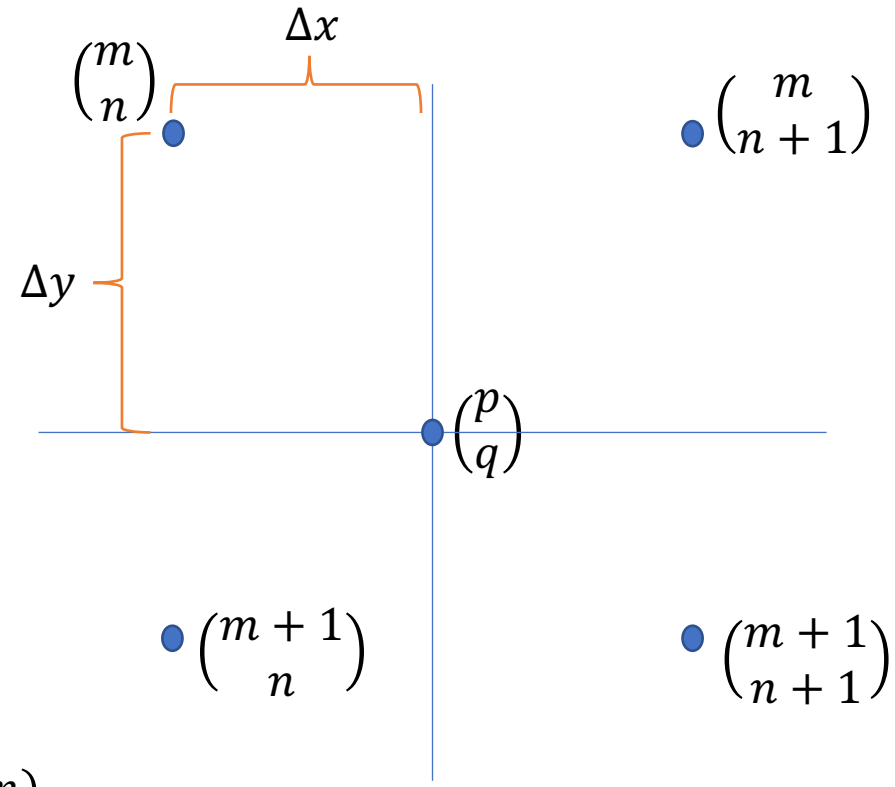
- Bilinear interpolation:

$$\begin{pmatrix} i \\ j \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \end{pmatrix} \xRightarrow{warping} \begin{pmatrix} u \\ v \end{pmatrix} \rightarrow \begin{pmatrix} p \\ q \end{pmatrix}$$

Image coordinates — Cartesian coordinates — Cartesian coordinates — Image coordinates



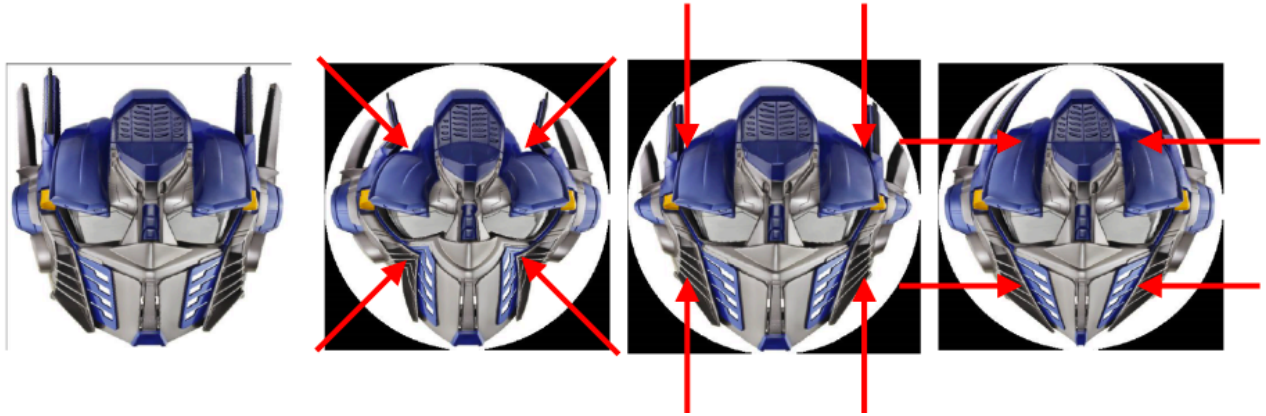
$$color(i,j) = color(p,q)$$

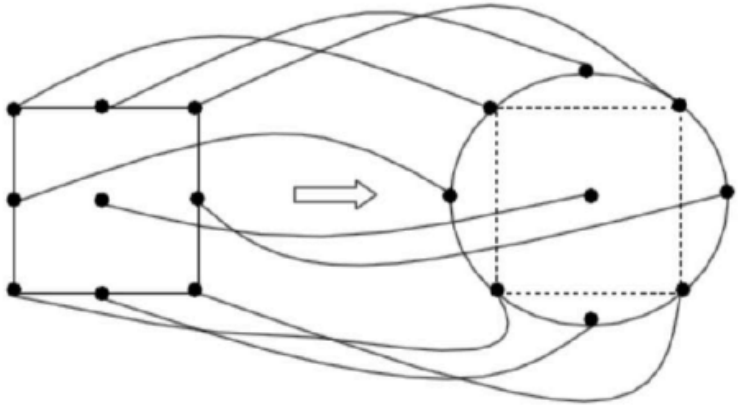


- *(p,q) may be fractional → Bilinear interpolation.*
- $\mathrm{m} = floor(p), n = floor(q)$
- $color(p,q) = \Delta x \Delta y color(m+1, n+1) + \Delta x(1-\Delta y) color(m, n+1) +$
  $(1-\Delta x)\Delta y color(m+1, n) + (1-\Delta x)(1-\Delta y) color(m, n)$

Image Warping

1. Decide the triangles you want to use for warping. Based on this, you may have more points or lines that should not change, beside the center and boundary.
2. Create a large enough canvas for the warped image.
3. Convert image coordinate to cartesian coordinates.
4. For each pixel in target, do inverse mapping to find corresponding pixel in original image.
5. Convert cartesian coordinate to image coordinate.
6. Do bilinear interpolation if required and fill in the the target image.
7. With the same method, warp the image back to the original shape.





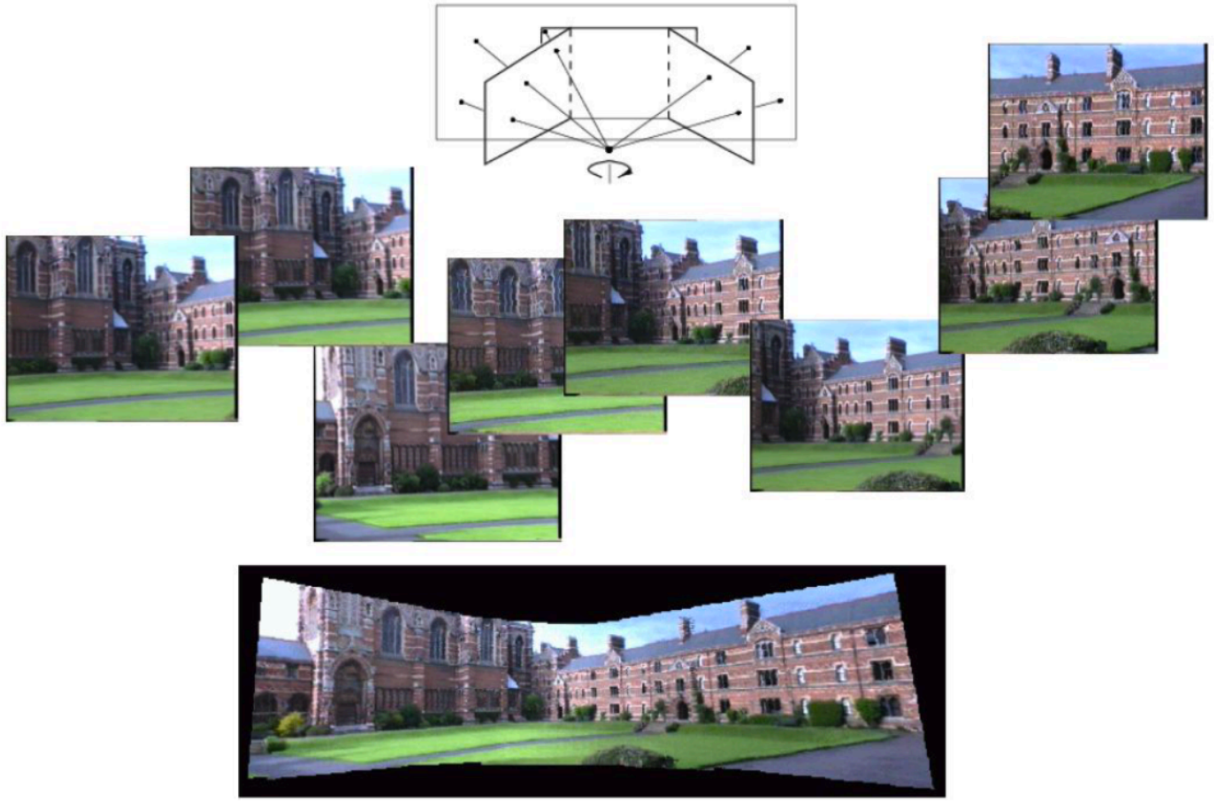
Homography Transformation & Image Stitching

➢ Homography

- Plane projective transformation
- Two views of a same planar surface
- Two camera share the same center

➢ Panorama

- Homography image stitching



Fitting a Homography

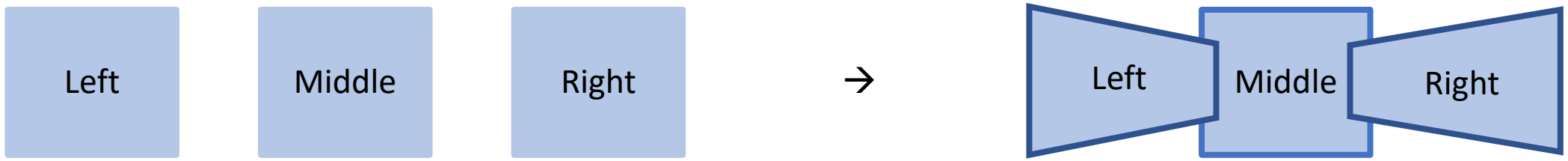
- Like affine transformations
- 8 degree of freedom (scale is arbitrary)

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}, h_{33} = 1$$

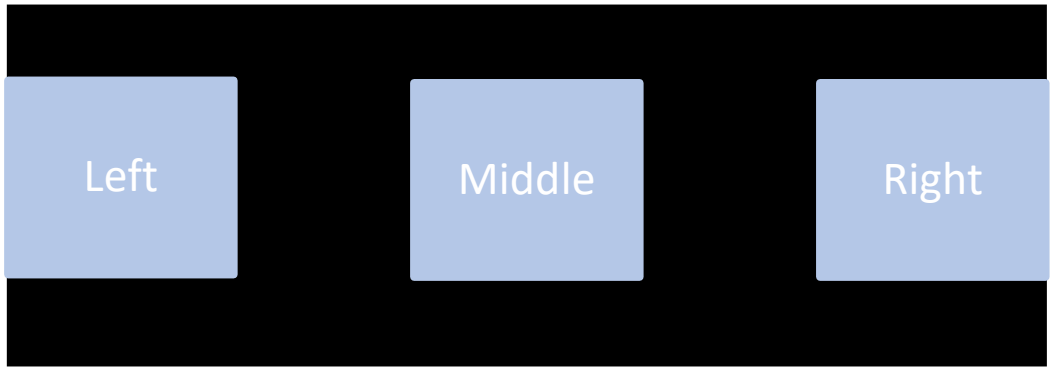
- Linear system with 8 unknowns.
- Each match provides 2 equations.
- Need at least 4 matches.

Homography Transformation & Image Stitching

- Image stitching

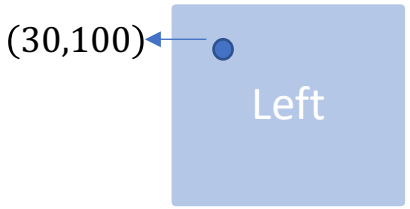


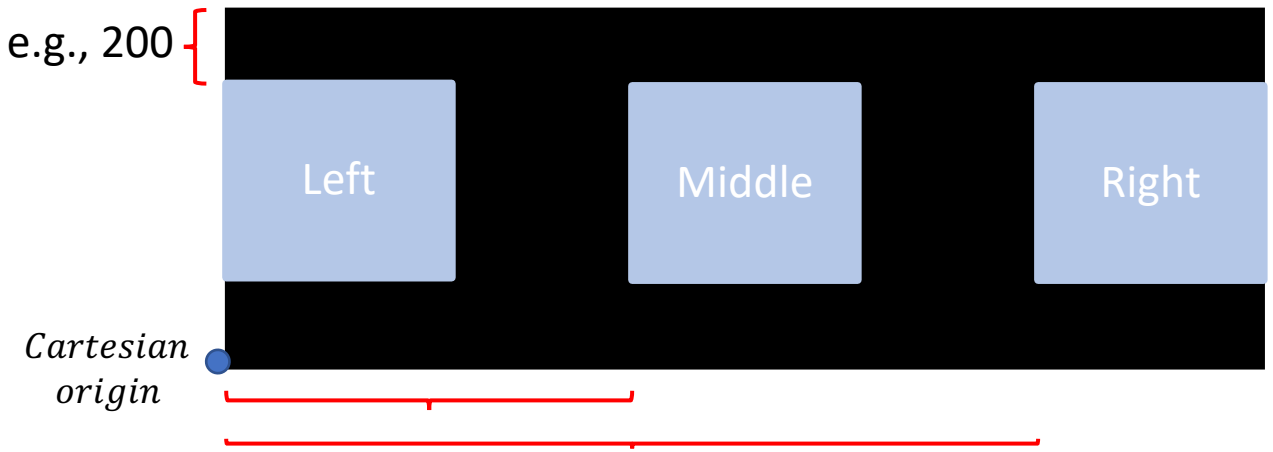
1. SURF/SIFT online source feature detection.
2. Find matching point pairs.
3. Create a large enough canvas.
4. Put left, middle, and right images on canvas. (e.g., 2000*2000)



Homography Transformation & Image Stitching

- Location conversion





❖ Take care of offsets:

location on individual image (i, j)	Location on canvas (i', j')	Cartesian location on canvas (x, y)
Left: (30, 100)	(230, 100)	(100-0.5, 2000-230+0.5)
Left: 3 more points matching middle		(x1,y1)
Middle: 4 points matching left		(x2,y2)
Middle: 4 points matching right		(x2,y2)
Right: 4 points matching middle		(x3,y3)

Homography Transformation & Image Stitching

Homography

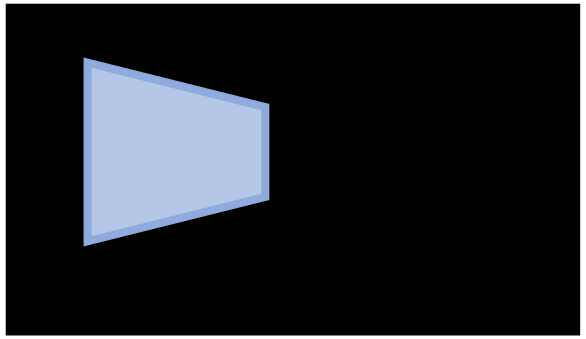
$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} x_2{}' \\ y_2{}' \\ w_2{}' \end{pmatrix}, \qquad \begin{cases} x_2 = \dfrac{x_2{}'}{w_2{}'} \\ y_2 = \dfrac{y_2{}'}{w_2{}'} \end{cases}$$

$$\rightarrow \begin{cases} h_{11}x_1 + h_{12}y_1 + h_{13} = x_2(h_{31}x_1 + h_{32}y_1 + h_{33}) \\ h_{21}x_1 + h_{22}y_1 + h_{23} = y_2(h_{31}x_1 + h_{32}y_1 + h_{33}) \end{cases}, \qquad h_{33}=1$$

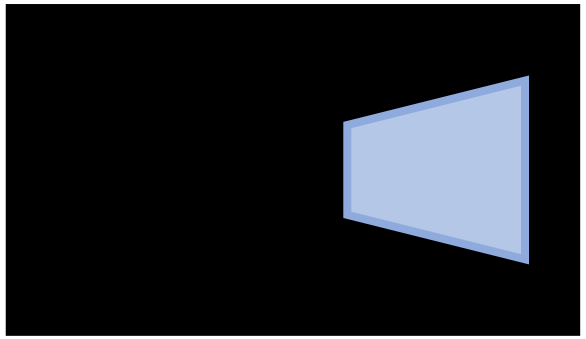
$4\ pairs \rightarrow 4\ equations \rightarrow solve\ h_{11}\text{-}\ h_{32}$

Homography Transformation & Image Stitching

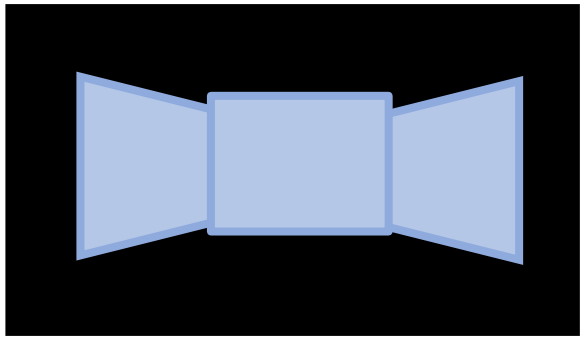
H-left2middle
→



H-right2middle
→



→



Overlap areas: take average

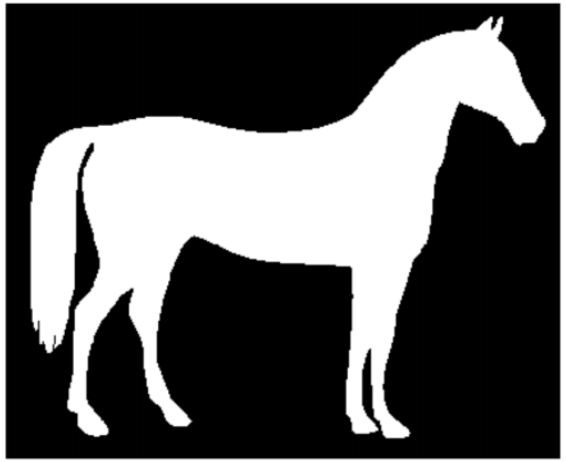
Homography Transformation & Image Stitching

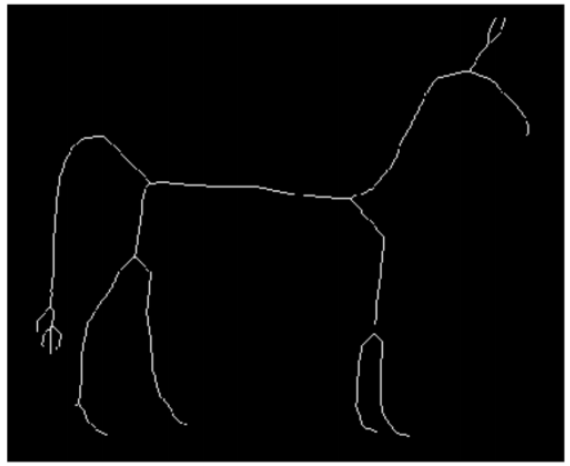
Image stitching Summary

1. Find matching point pairs, select good ones. (online source)
2. Create a large enough canvas.
3. Put left, middle, and right images on canvas.
4. Convert the location of matching points on individual images to the cartesian location on canvas.
5. Find the transformation matrix H, using the coordination of 4 matching points.
6. Warp the left and right images, to stitch to the middle image.
7. Use the average in overlap areas.
8. Use interpolation to find the value for left out pixels (black dots).

Morphological Processing

- Shrinking: central pixel
- Thinning: central line
- Skeletonizing: skeleton containing edges





Morphological Processing

- Shrinking:

Objects without holes shrink to a point, and objects with holes shrink to a connected ring halfway between each hole and the outer boundary.

- Thinning:

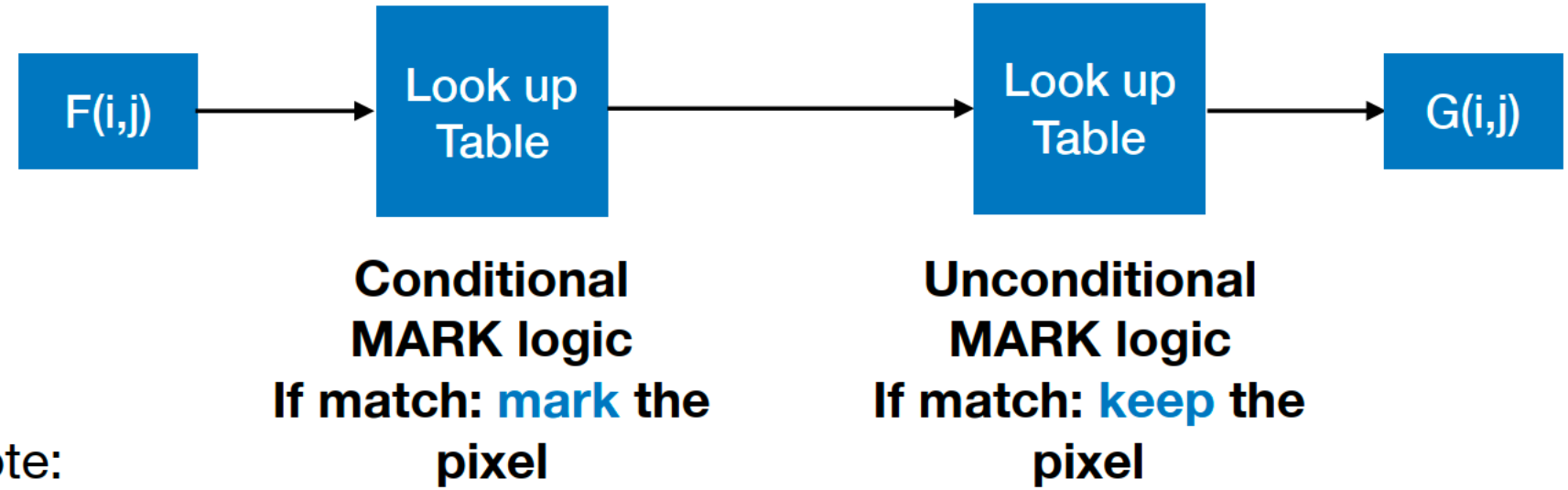
Object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a connected ring halfway between each hole and the outer boundary.

- Skeletonizing:

removes pixels on the boundaries of objects but does not allow objects to break apart. The pixels remaining make up the image skeleton.

Morphological Processing

- Implement: Look up Table
- 2-pass filter



- Note:
  - Table 14.3-1 Shrink /Thin /Skeleton conditional LUT
  - Table 14.3-2 Shrink/ Thin unconditional LUT
  - Table 14.3-3 Skeleton unconditional LUT

References

- Yao Zhu, EE569 Spring 2020 discussion 6 & 7

Q
A