ARAVIND
2019102014

DIP Lecture 14
Geometric Operations

→ Modifies coordinates of image pixels

$$I(x,y) \to I'(x',y')$$

→ Common operations:
- → Scale
- → Rotate
- → Reflection
- → Translate
- → Shear
- → Affine transformation
  - ↳ General image contant linear geometric transfo

→ Problem: Pixel image can go out of bounds, so needs to be solved

→ Translation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} dx \\ dy \end{pmatrix}$$

→ Scaling:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

s_x, s_y are scale factors
>1, stretching
<1, contracting

→ Shearing: [Bivariate]

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & b_x \\ b_y & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

→ Rotation [Rotate by angle α] → usually counterclockwise.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} c\alpha & -s\alpha \\ s\alpha & c\alpha \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

For clockwise,

$$\begin{pmatrix} c\alpha & s\alpha \\ -s\alpha & c\alpha \end{pmatrix}$$

Homogenous Coordinate

→ Transforms point from euclidean plane to projective plane by adding dummy variable $(x, y, 1)$

→ Overall scaling cnt imp.

→ If $x' = a_0x + a_1y + a_2$
$y' = b_0x + b_1y + b_2$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Vector matrix multiplication can be done well with homogenous coordinate

☆ All linear transforms are affine but not vice versa

Affine (3-point) Mapping:

Examples:
1) Trans.
2) Scaling
3) Rotatio

$$H = RST = \begin{bmatrix} c\theta & s\theta & 0 \\ -s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_0 & 0 & 0 \\ 0 & s_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

→ Inverse of transform matrix is inverse mapping

To get new coordinate,

$$X = Hx$$

↑ Transformation matrix

Forward Mapping:

Problems
- → Dest. image may not accomodate all transformed pixel
- → Transformed coordinate may not be integers

⇓ Leads to

→ Expanded view increases overall image dimensions
→ As each output pixel may not be integers → Can result in holes if rounded off

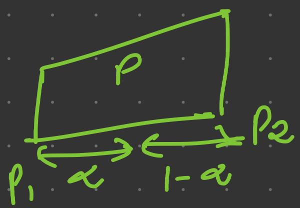
Solution → Backward / Inverse Mapping

Iterates over each pixel in output image and uses inverse transformation to determine position from which pixel intensity value must be sampled (interpolation)

↳ Won't result in holes in output image

Interpolation function

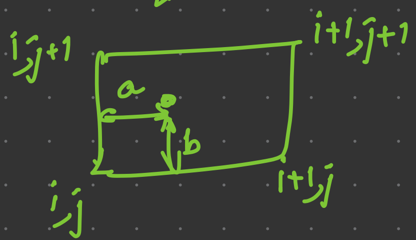
$$P = P_1(1-\alpha) + P_2\alpha$$



Bilinear: Weighted average of (2x2) 4 neighbouring pixels

Bicubic: Weighted average of (4x4) 16 neighboring pixels.

Closer pixels are weighted more → Smoother



$$f(x,y) = (1-a)(1-b)f[i,j] + a(1-b)f[i+1,j] + ab\,f[i+1,j+1] + (1-a)b\,f[i,j+1]$$

2 aspects of transform

→ Mapping [Type]

→ Interpolation [quality]

Geometric transforms and registration

Given I & O, we find transform T.

Point to point mapping:

find no of points $\{p_0, p_1 \ldots p_{n-1}\}$ in image A that matches $\{q_0, q_1 \ldots q_{n-1}\}$ in image B

$$q = Hp$$

$$H = QP^T (PP^T)^{-1} = QP^+$$ → pseudo inverse

Solution of H that provides minimum mean squared error

Face morph:

Combines face images from multiple identities to match constituents.

Uses:

- → Correct distortions introduced during imaging
- → Transformation: To create special effects
- → Registrations: Register two images taken of same scene at diff times/condition

Homography: $x' = Hx$ where H = homography matrix, $x = (u, v, 1)$, $x' = (u', v', 1)$

Two images are related only if:

- → Both are viewing same plane with diff angle
- → " " taken from same camera with diff angle [without translation]

→ Independent of scene structure → Holds regardless of what's seen

Doesn't depend on what cameras look at

→ New img is warped version of original img

Computing homography

Given R, K

$$x' = KRK^{-1}x$$

→ Used to remove perspective transform

→ Applications in birds eye view, mosaicing