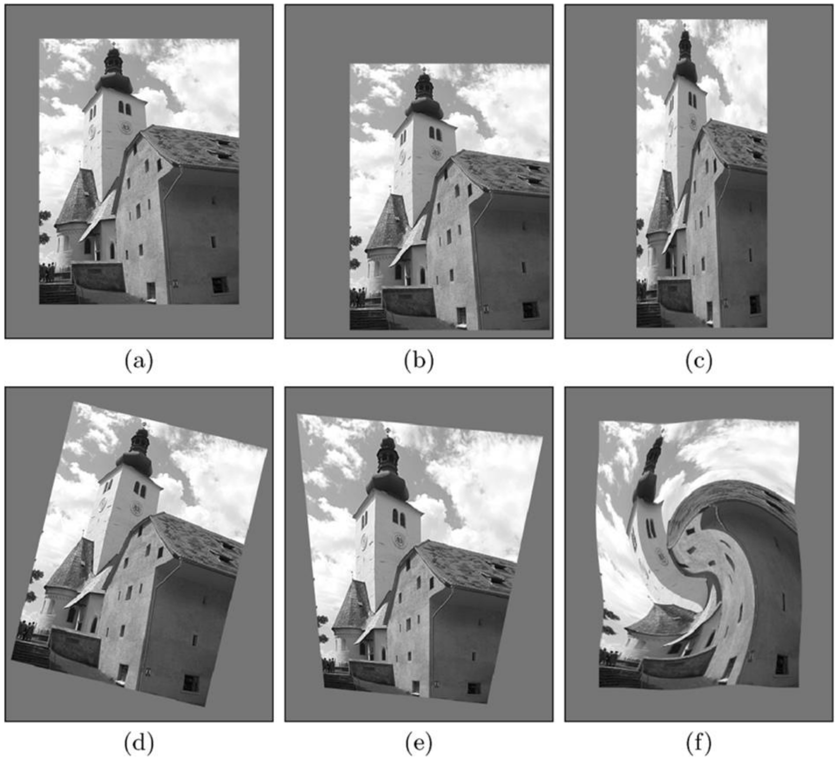
Geometric Operations

- Filters, point operations change intensity
- Pixel position (and geometry) unchanged
- Geometric operations: change image geometry
- Examples: translating, rotating, scaling an image

Examples of Geometric operations



Geometric Operations

- **Example applications of geometric operations:**
  - **Zooming images, windows to arbitrary size**
  - **Computer graphics: deform textures and map to arbitrary surfaces**
- **Definition: Geometric operation transforms image *I* to new image *I'* by modifying coordinates of image pixels**

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

- **Intensity value originally at (x,y) moved to new position (x',y')**

Example: Translation geometric operation moves value at (x,y) to $(x + d_x, y + d_y)$

- **Geometric Operations**
  - **Since image coordinates can only be discrete values, some transformations may yield (x',y') that's not discrete**
  - **Solution: interpolate nearby values**
  - **Translation: (shift) by a vector (d_x, d_y)**

$$T_x : x' = x + d_x \qquad \text{or} \qquad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$
$$T_y : y' = y + d_y$$

  - **Scaling: (contracting or stretching) along x or y axis by afactor s_x or s_y**

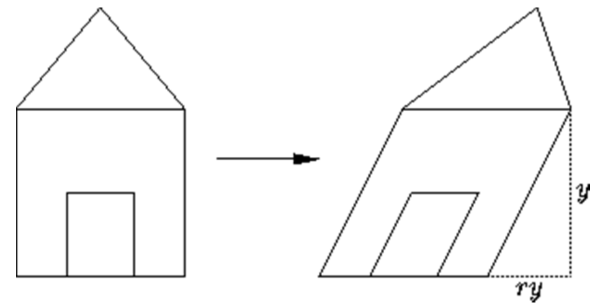
$$T_x : x' = s_x \cdot x \qquad \text{or} \qquad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$
$$T_y : y' = s_y \cdot y$$

- **Shearing: along x and y axis by factor b_x and b_y**

$$T_x : x' = x + b_x \cdot y$$
$$T_y : y' = y + b_y \cdot x$$

or

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



- **Rotation: the image by an angle α**

$$T_x : x' = x \cdot \cos\alpha - y \cdot \sin\alpha$$
$$T_y : y' = x \cdot \sin\alpha + y \cdot \cos\alpha$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Image Flipping & Rotation by 90 degrees

- **We can achieve 90,180 degree rotation easily**
- **Basic idea: Look up a transformed pixel address instead of the current one**
- **To flip an image upside down:**
  - **At pixel location *xy*, look up the color at location *x (1 – y)***

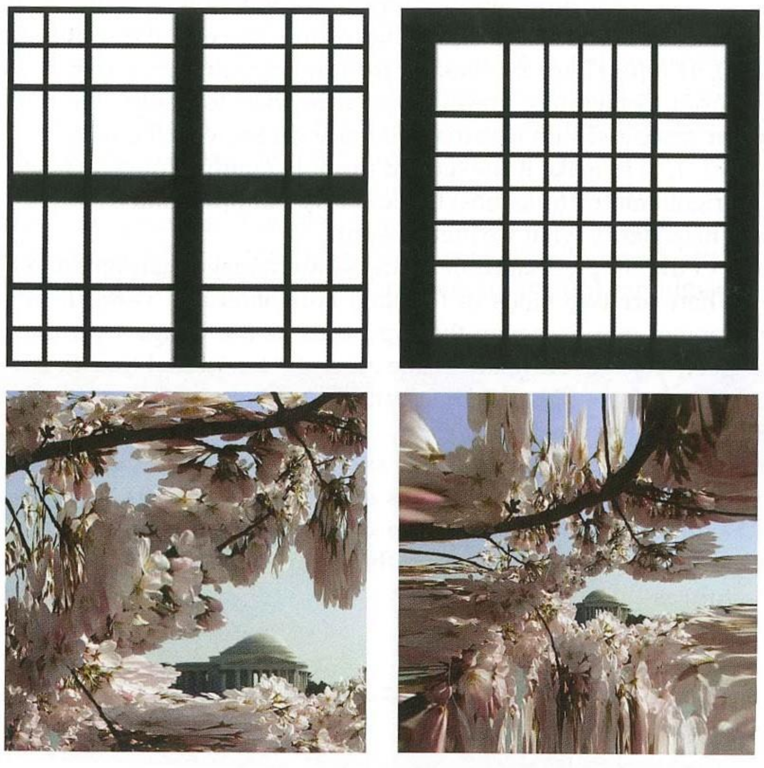
- **For horizontal flip:**
  - **At pixel location *xy*, look up *(1 – x) y***

- **Rotating an image 90 degrees counterclockwise:**
  - **At pixel location *xy*, look up *(y, 1 – x)***

Image Flipping, Rotation and Warping

- **Image warping: we can use a function to select which pixel somewhere else in the image to look up**
- **For example: apply function on both texel coordinates (x, y)**

$$x = x + y * \sin(\pi * x)$$



Homogeneous Coordinates

- **Notation useful for converting scaling, translation, rotating into point-matrix multiplication**
- **To convert ordinary coordinates into homogeneous coordinates**

$$\boldsymbol{x} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{converts to} \quad \hat{\boldsymbol{x}} = \begin{pmatrix} \hat{x} \\ \hat{y} \\ h \end{pmatrix} = \begin{pmatrix} h\,x \\ h\,y \\ h \end{pmatrix}$$

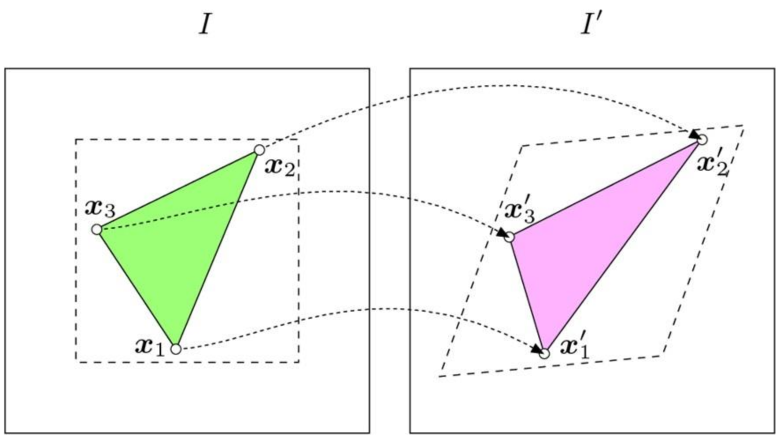
Affine (3-Point) Mapping

- **Can use homogeneous coordinates to rewrite translation,**

  rotation, scaling, etc as vector-matrix multiplication

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- **Affine mapping: Can then derive values of matrix that achieve desired transformation (or combination of transformations)**



- **Inverse of transform matrix is inverse mapping**