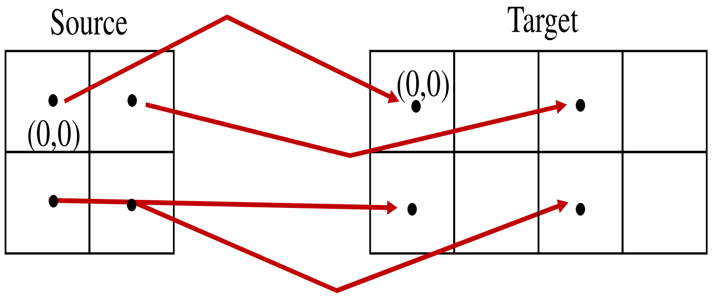
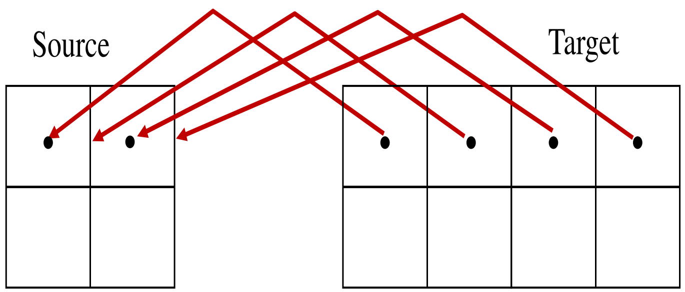
01.12.2016.

- Example: Scaling along X
  - Forward mapping: $x' = 2x\ ;\ y' = y$

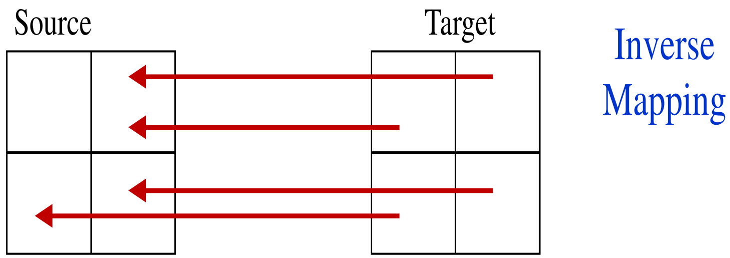


  - Inverse mapping: $x = x'/2\ ;\ y = y'$



Inverse Mapping

- Inverse mapping: $$x' \to f_x^{-1}(x', y') = x$$ $$y' \to f_y^{-1}(x', y') = y$$



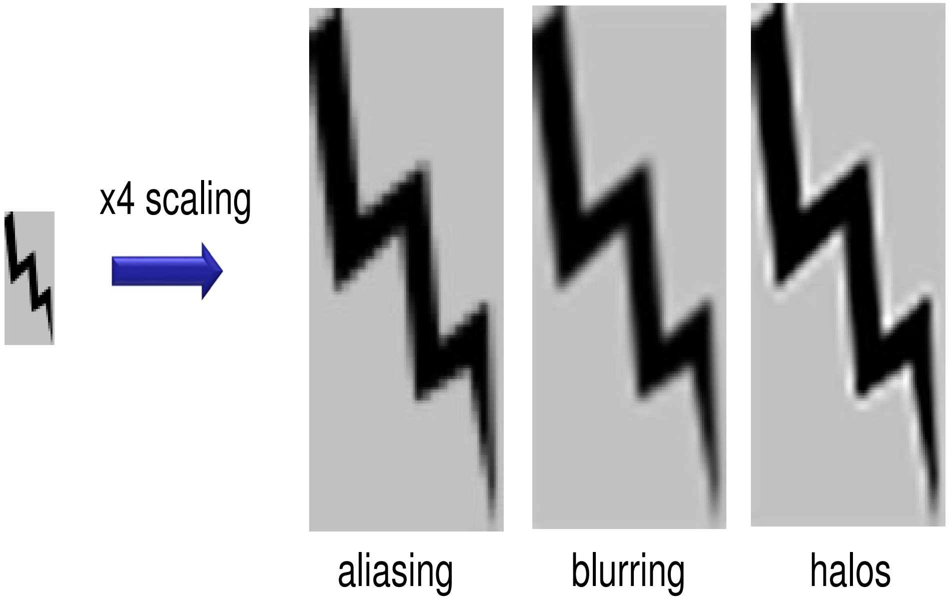
- Each target pixel assigned a single color.
- Color Interpolation is required.

⊛ מתחילים מתמונה ריקה, ועבור כל תמונה, אינדקס נחשב מאיפה הגיע
מאיפה במקור. ככה מבטיחים שאין חורים.

⊛ אם חישוב מפיל אותנו בין שני פיקסלים (2.5-3.5) נבצע
ממוצע ממושקל של הפיקסלים השכנים בתמונת המקור.

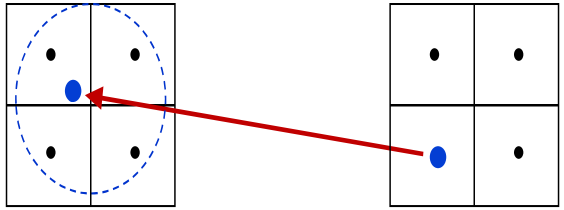
Interpolation

- Good interpolation techniques attempt to find an optimal balance between three undesirable artifacts: aliasing, blurring, and edge halos.



Interpolation

- What happens when a mapping function calculates a fractional pixel location?



- **Interpolation**: generates a new pixel by analyzing the surrounding pixels.

Original Image

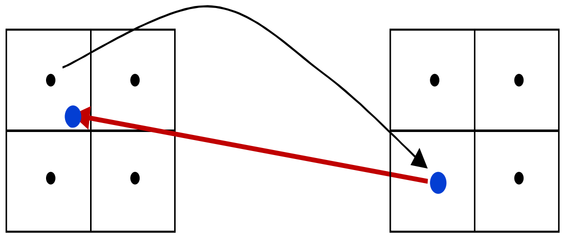
Nearest N. Interpolation

Nearest Neighbor Interpolation

- The assign value is taken from the pixel closest to the generated location:

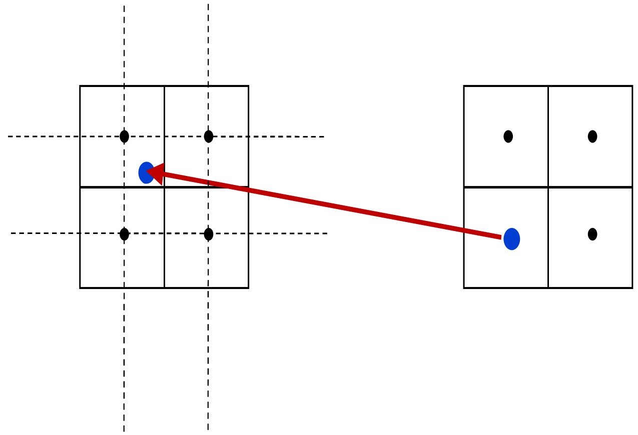
$$I'(x',y') = I\left(round\left\{f_x^{-1}(x',y')\right\}, round\left\{f_y^{-1}(x',y')\right\}\right)$$

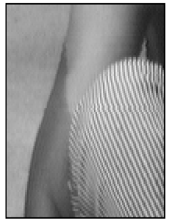
- Advantage:
  - Fast
- Disadvantage:
  - Jagged results
  - Aliasing near edges



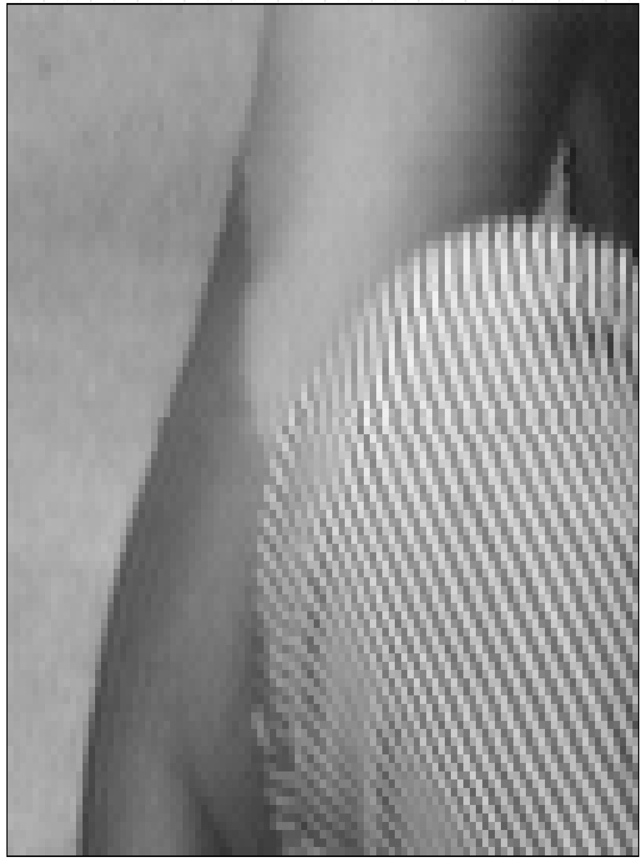
Bilinear Interpolation

- The assign value is a weighted sum of the four nearest pixels.
- Each weight is proportional to the distance from each existing pixel.



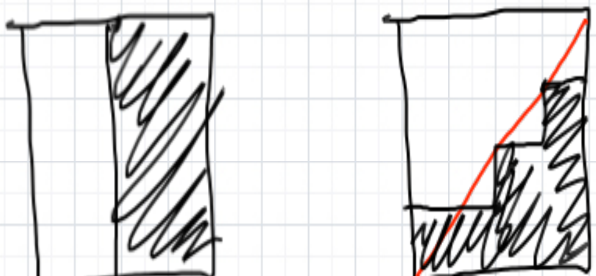


Original Image

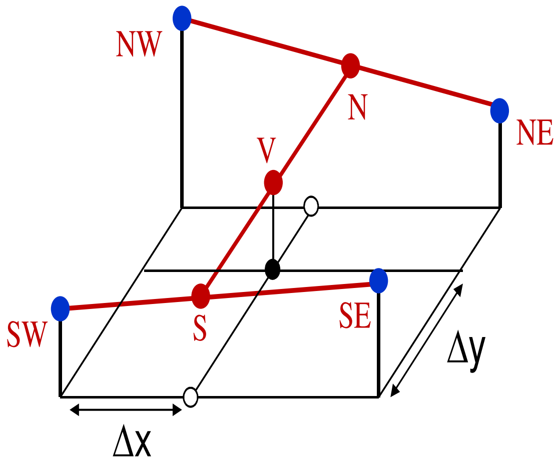


Nearest N. Interpolation

כאן רואים את הזיגזגיות - בזוויות יש בעיה, כי אין מעבר
חלק, אלא את הצבעים של השכן הקרוב ביותר.



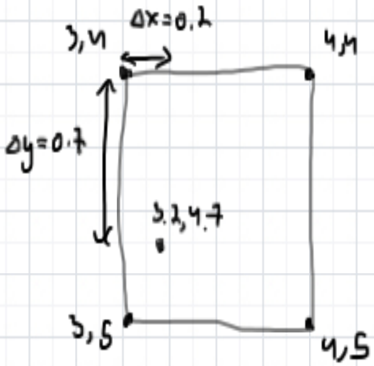
Bilinear Interpolation



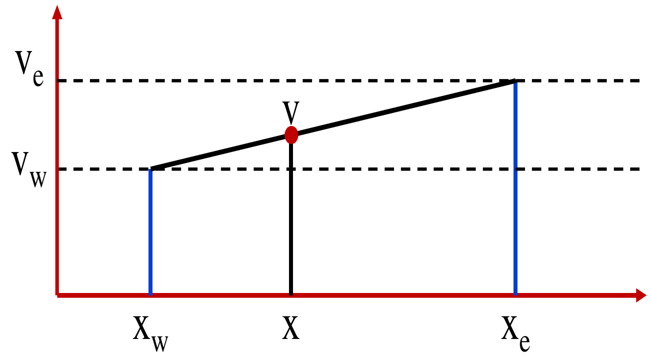
$$S = SE \cdot \Delta x + SW \cdot (1 - \Delta x)$$

$$N = NE \cdot \Delta x + NW \cdot (1 - \Delta x)$$

$$V = N \cdot \Delta y + S \cdot (1 - \Delta y)$$



Linear Interpolation



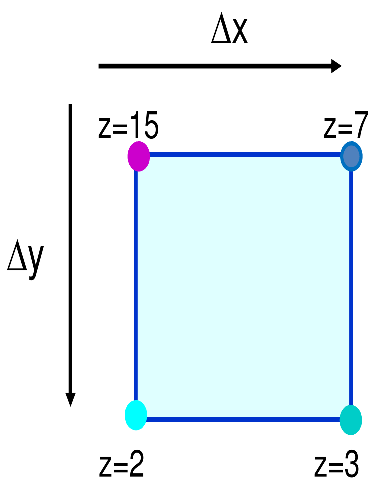
$$\frac{x - x_w}{x_e - x_w} = \frac{v - v_w}{v_e - v_w}$$

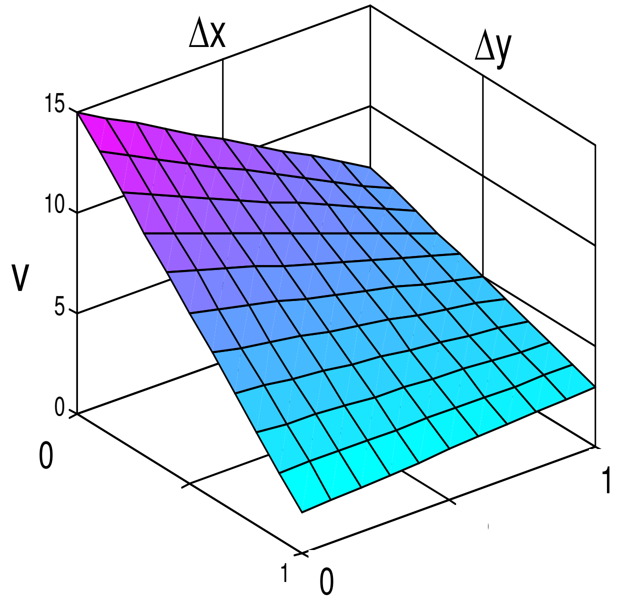
- Isolating v in the above equation:

$$\boxed{v = \alpha v_e + (1 - \alpha) v_w} \qquad \text{where } \alpha = \frac{x - x_w}{x_e - x_w}$$

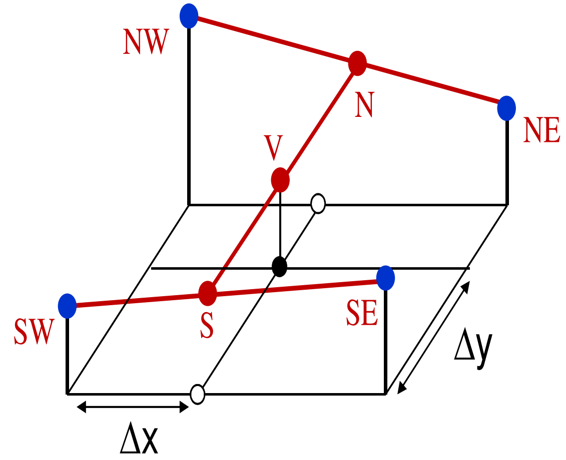
* הממוצע הממושקל מחושב לפי חלקיות הדיוק בין השכנים.
* בעצם זה לפי כמה רחוק או כמה קרוב נמצאים מהשכנים.

Bilinear example





Bilinear Interpolation



- The bilinear interpolation is the best fit low-degree polynomial of the form:

$$v(\Delta x, \Delta y) = \sum_{i,j=0}^{1} a_{ij} \Delta x^i \Delta y^j$$

- The pixel's boundaries are C_0 continuous (continuous values across boundaries).

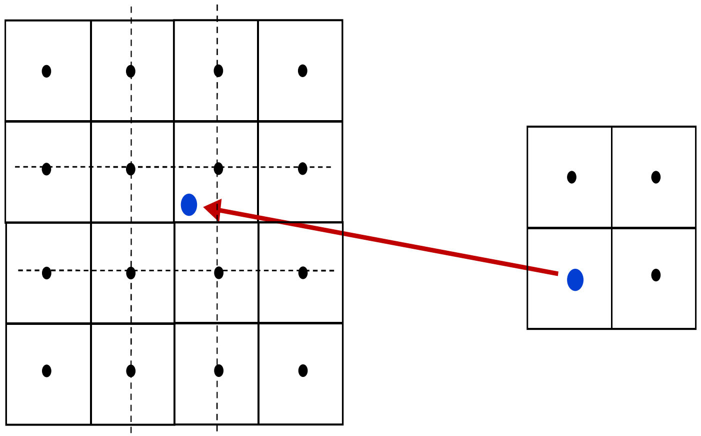
⊛ הסדר של הפולינום שלנו הוא 2.

⊛ a_{ij} הוא צירוף של כל הפינות המתאימות לנקודה.

Bicubic Interpolation

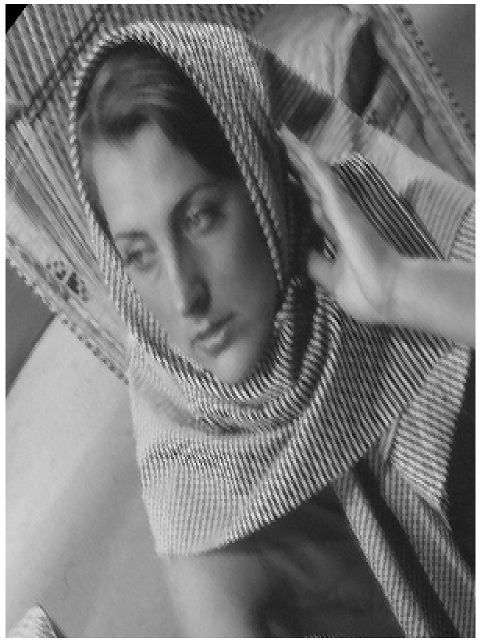
- The assign value is a weighted sum of the 4x4 nearest pixels:

$$v(\Delta x, \Delta y) = \sum_{i,j=0}^{3} a_{ij} \Delta x^i \Delta y^j$$



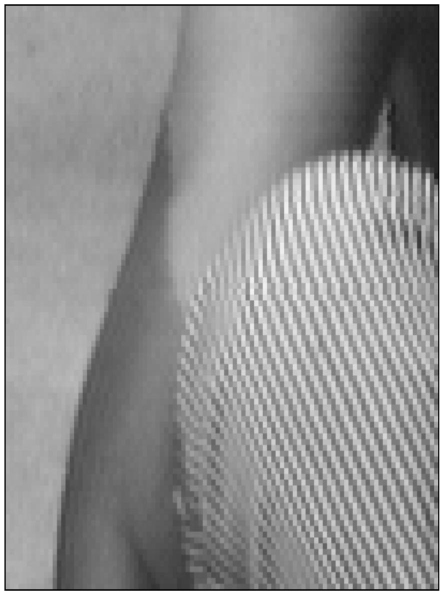
⊛ כאן, דרגת הפולינום היא 6.

⊛ יש 16 a נעלמים ו-16 משוואות.



Nearest N. Interpolation

Bilinear Interpolation



Nearest N. Interpolation

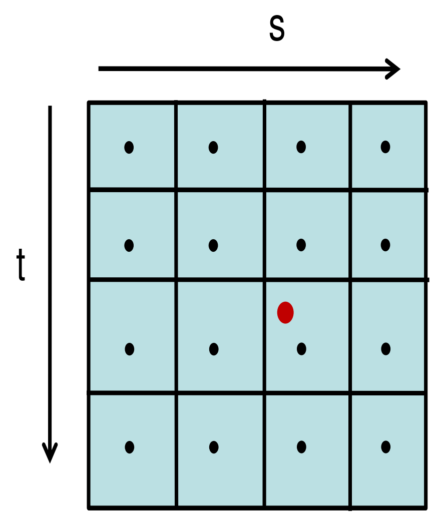
Bilinear Interpolation

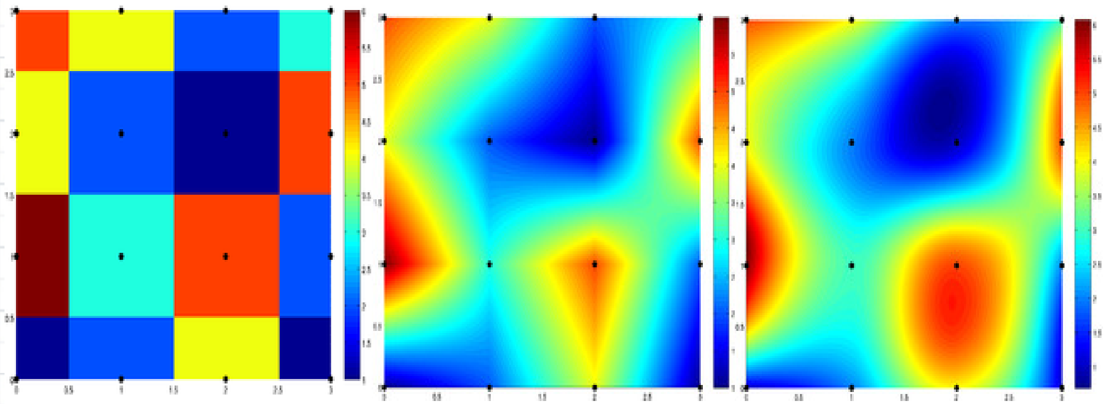
How can we find the right coefficients?

- Denote the pixel values V_{pq} $\{p,q=0..3\}$
- The unknown coefficients are a_{ij} $\{i,j=0..3\}$

$$V_{pq} = \sum_{i,j=0}^{3} a_{ij} \Delta x^i \Delta y^j \quad \text{for } p,q = \{0..3\}, \quad \Delta x, \Delta y \in [-1,2]$$

- We have a linear system of 16 equations with 16 coefficients.
- The pixel's boundaries are C_1 continuous (continuous derivatives across boundaries).



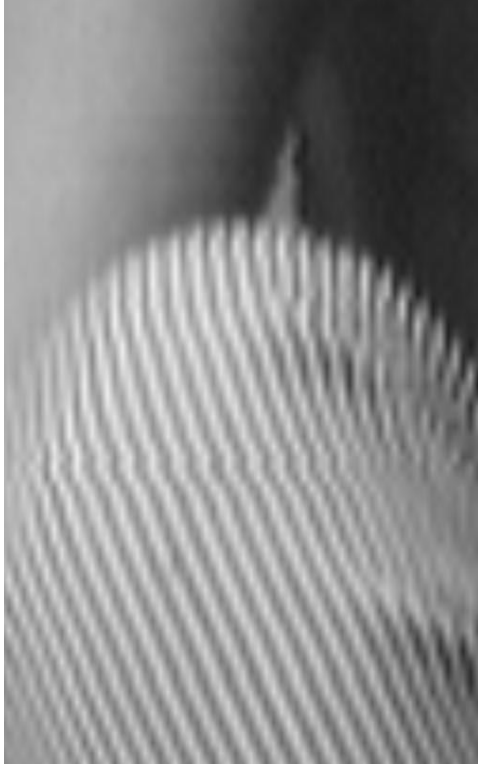


N.N Bilinear Bicubic

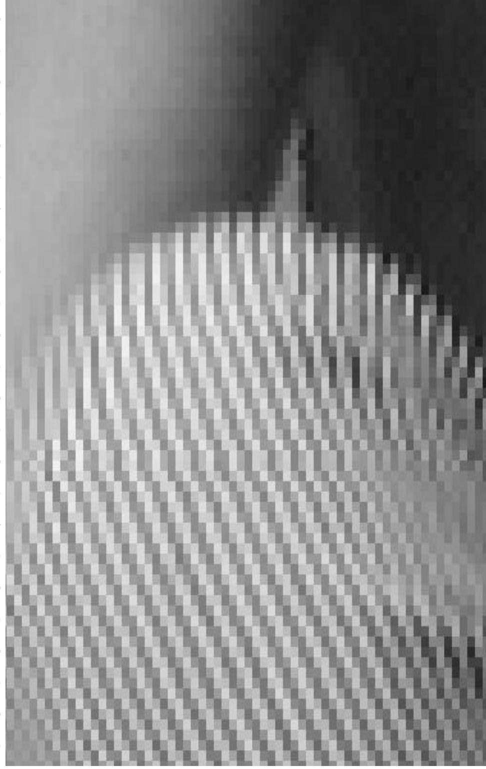
דוגמא על השקף הבא.

Applying the Transformation

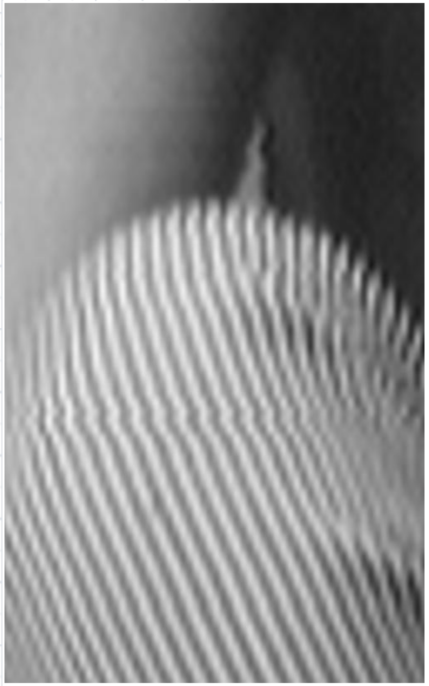
```
T = ......  % 2x2 transformation matrix
[r,c] = size(img)

% create array of destination x,y coordinates
[X,Y]=meshgrid(1:c,1:r);

% calculate source coordinates
sourceCoor = inv(T) * [X(:) Y(:) ]' ;

% calculate nearest neighbor interpolation
Xs = round(sourceCoor(1,:));
Ys = round(sourceCoor(2,:));

indx=find(Xs<1 | Xs>r);   %out of range pixels
Xs(indx)=1; Ys(indx)=1;

indy=find(Ys<1 | Ys>c);   %out of range pixels
Xs(indy)=1; Ys(indy)=1;

% calculate new image
newImage = img((Xs-1).*r+Ys);
newImage(indx)=0; newImage(indx)=0;
newImage = reshape(newImage,r,c);
```



Bilinear



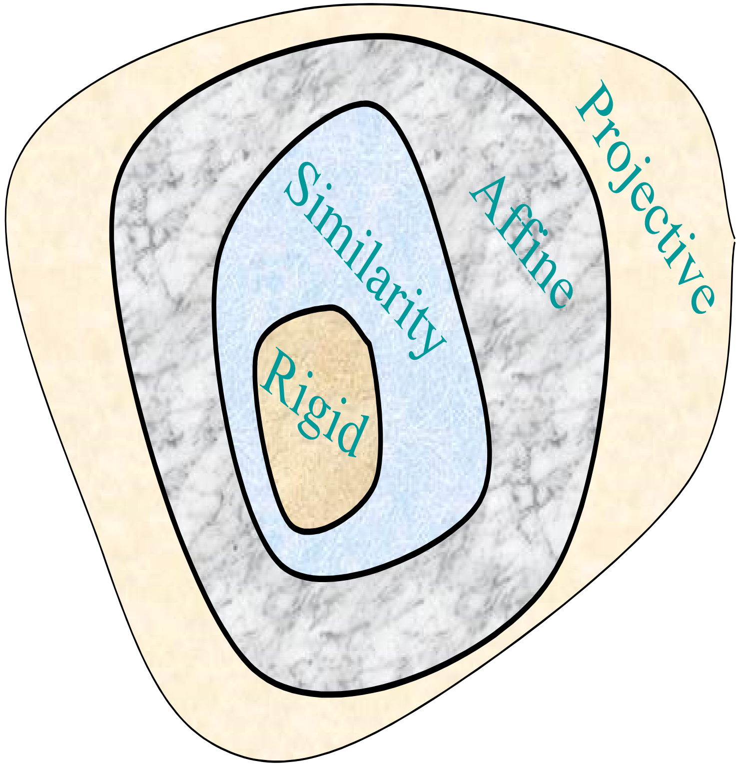
N.N



Bicubic

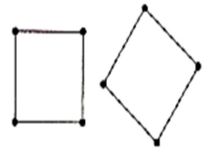
Types of linear 2D transformations

- All above transformations are groups where Rigid ⊂ Similarity ⊂ Affine ⊂ Projective

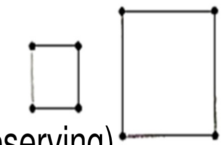


Types of linear 2D transformations

- Rigid (Euclidean) transformation: חפיפת משולשים
  - Translation + Rotation (distance preserving).



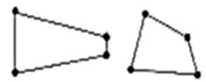
- Similarity transformation: דמיון משולשים
  - Translation + Rotation + Uniform Scale (angle preserving).



[illegible]
[illegible]

- Affine transformation: קווים מקבילים נשמרים, זוויות לא.
  - Translation + Rotation + Scale + Shear (parallelism preserving).

- Projective transformation הכל אפשרי
  - Cross-ratio preserving



All above transformations are groups where:
Rigid ⊂ Similarity ⊂ Affine ⊂ Projective

Matrix Notation - Scale

- Scale(a,b): (x,y) ⟶ (ax,by)

$$\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax \\ by \end{bmatrix}$$

- If *a* or *b* are negative, we get reflection.
- Inverse: $S^{-1}(a,b)=S(1/a,1/b)$

$$\begin{bmatrix} 1/a & 0 \\ 0 & 1/b \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Matrix Notation

- Every location (x,y) is treated as a column vector:

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

- Coordinate transformation is obtained by multiplying with a 2x2 matrix?

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

Matrix Notation - Rotation

- Rotate(θ):

  (x,y) ⟶ (*x*cosθ+*y*sinθ , -*x*sinθ + *y*cosθ)

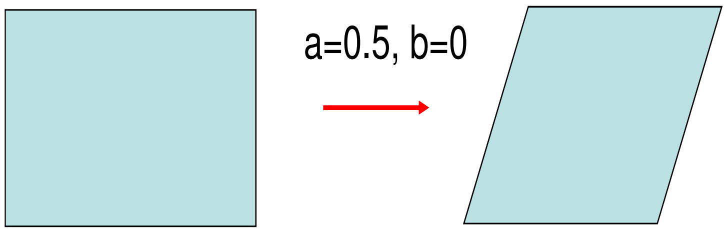
$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta + y\sin\theta \\ -x\sin\theta + y\cos\theta \end{bmatrix}$$

- Inverse: $R^{-1}(\theta)=R^{T}(\theta)=R(-\theta)$

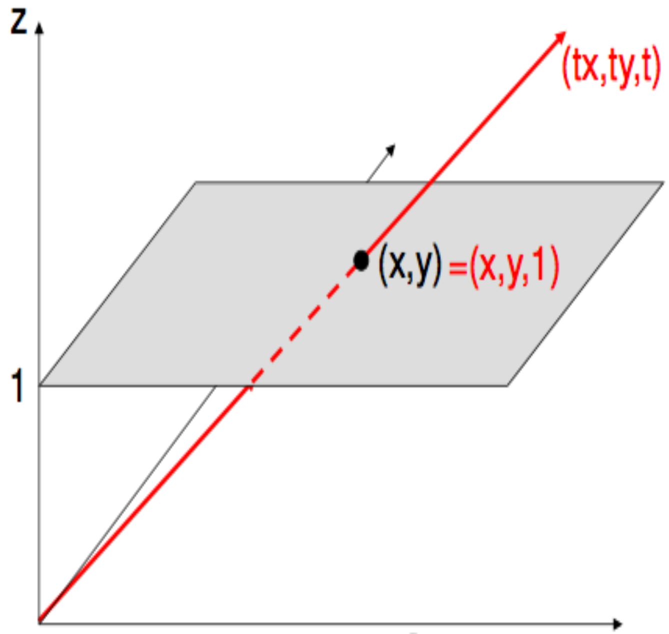
Matrix Notation - Shear

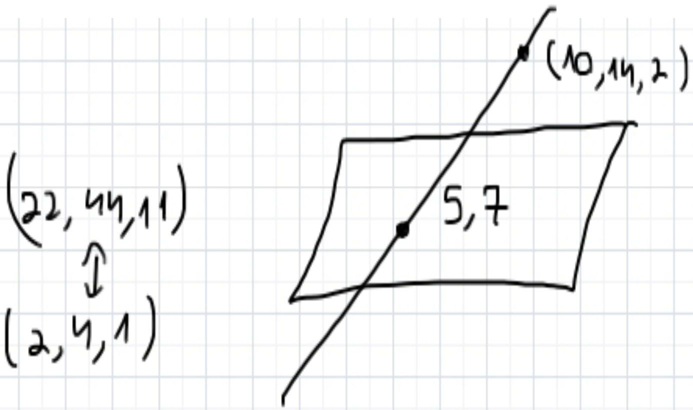
- Shear(a,b): (x,y) ⟶ (x+ay,y+bx)

$$\begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+ay \\ y+bx \end{bmatrix}$$



Homogeneous Coordinates





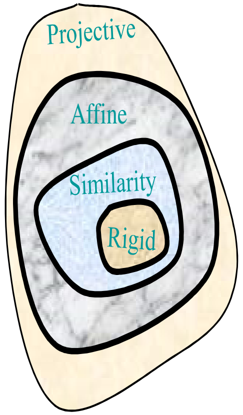
Matrix Notation - Translation

- Translation(a,b): $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x+a \\ y+b \end{bmatrix}$

- Cannot represent translation using 2x2 matrices.

- Inverse: $\begin{bmatrix} x' \\ y' \end{bmatrix} \rightarrow \begin{bmatrix} x'-a \\ y'-b \end{bmatrix}$

Some 2D Transformations



- Translation :

$$\begin{bmatrix} X' \\ Y' \\ W' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+t_x \\ y+t_y \\ 1 \end{bmatrix}$$

- Affine transformation:

$$\begin{bmatrix} X' \\ Y' \\ W' \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Projective transformation:

$$\begin{bmatrix} X' \\ Y' \\ W' \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ e & f & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- Homogeneous Coordinates is a mapping from R^n to R^{n+1}:

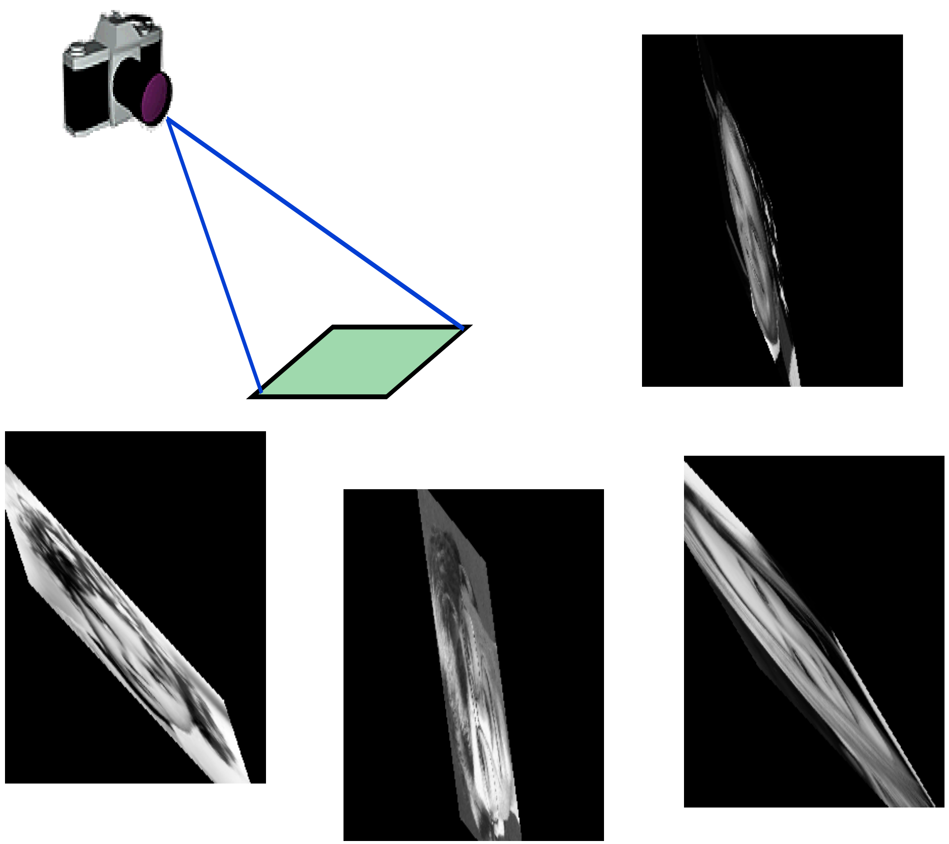
$$\boxed{(x, y) \rightarrow (X, Y, W) \equiv (tx, ty, t)}$$

- Note: (tx,ty,t) all correspond to the same non-homogeneous point (x,y). E.g. $(2,3,1) \equiv (6,9,3) \equiv (4,6,2)$.

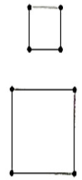
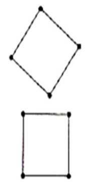
- Inverse mapping:

$$(X, Y, W) \rightarrow \left(\frac{X}{W}, \frac{Y}{W}\right) = (x, y)$$

Global Transformations – Image Rectification



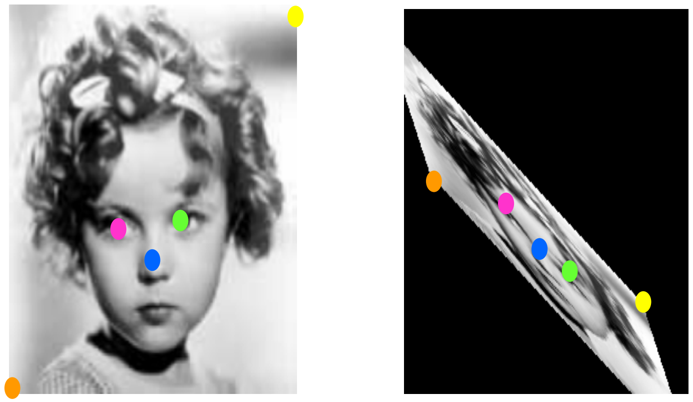
Hierarchy of Linear 2D Transformations

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, $\mathbf{I}, \mathbf{J}$ (see section 1.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

degrees of freedom

דרגות חופש - dof

Global Transformations – Global Warping



points $(x'_i, y'_i) \xleftarrow{\text{match}}$ points (x_i, y_i)

$$\begin{bmatrix} x'_1 & x'_2 & x'_3 & \cdots \\ y'_1 & y'_2 & y'_3 & \cdots \\ 1 & 1 & 1 & \cdots \end{bmatrix} = A \begin{bmatrix} x_1 & x_2 & x_3 & \cdots \\ y_1 & y_2 & y_3 & \cdots \\ 1 & 1 & 1 & \cdots \end{bmatrix}$$

Global Transformations – Global Warping

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Global Transformations – Global Warping

Affine Warping:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}$$

- How many points uniquely define the affine (projective) transformation?
- How can we find the affine transformation?
- What if we have more points?
- What can be done if points coordinates are inaccurate?

Global Transformations – Global Warping

Affine Warping:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rearrange:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}$$

Solving Linear Equation - LMS

$$\boxed{X' = XA}$$

Solve for A in terms of the least mean square. i.e. find A which minimizes:

$$\|X' - XA\|^2$$

Take derivative wrt A and equate to 0:

$$-2X^T(X' - XA)=0 \longrightarrow X^TX' = X^TXA$$

$$(X^TX)^{-1}X^TX' = (X^TX)^{-1}X^TXA$$

$$\underbrace{(X^TX)^{-1}X^T}_{pinv(X)}X' = A$$

$$\boxed{pinv(X)\cdot X' = A}$$

Global Transformations – Global Warping

Affine Warping:

Given 3 paired points:

$$\begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ x_2 & y_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_2 & y_2 & 0 & 1 \\ x_3 & y_3 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_3 & y_3 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}$$