اصول پردازش تصویر

Principles of Image Processing

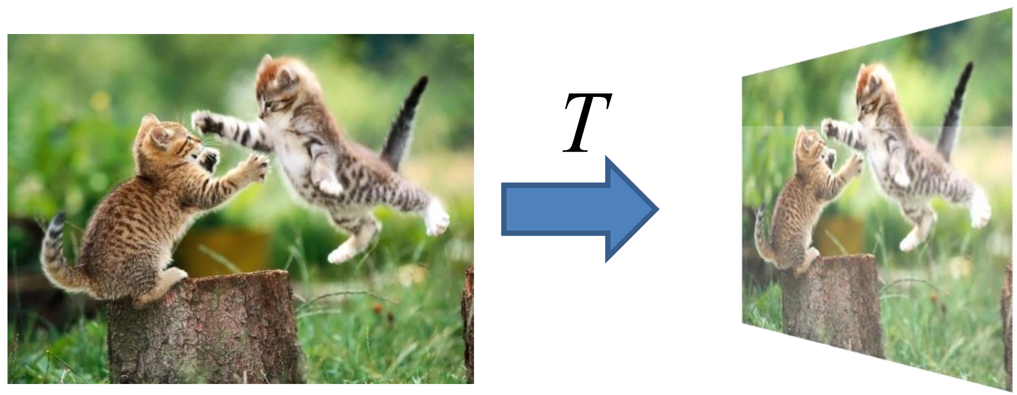
مصطفي كمالى تبريزى

۳ آبان ۱۳۹۹

جلسه دهم

Geometric Transformations and Image Warping

Geometric Transformations



y

x

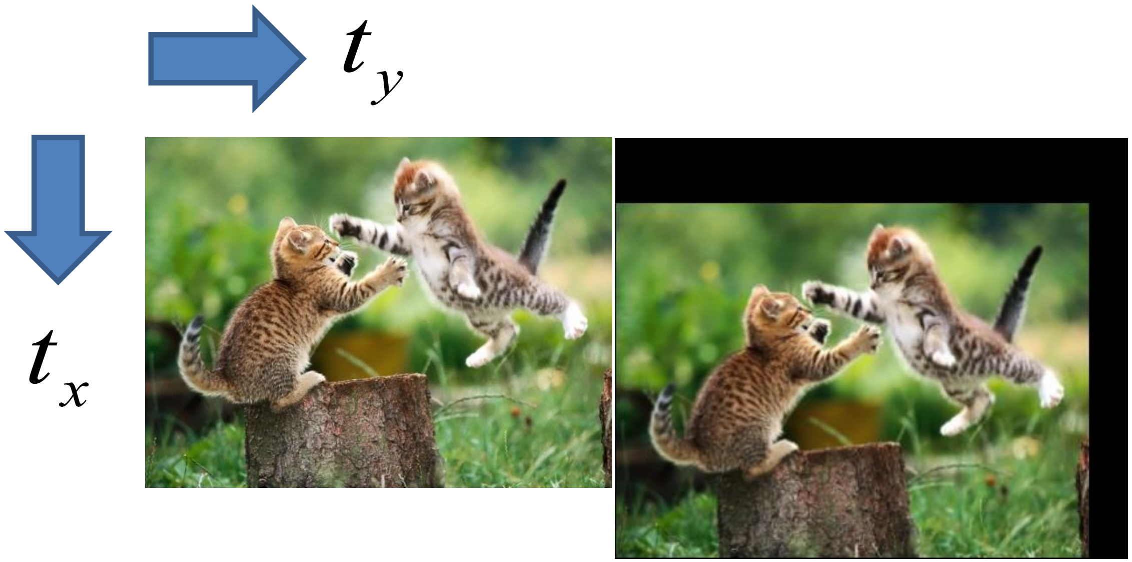
(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)	(4,8)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)
(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)
(7,1)	(7,2)	(7,3)	(7,4)	(7,5)	(7,6)	(7,7)	(7,8)

$$T\begin{bmatrix}4\\5\end{bmatrix}=\begin{bmatrix}5.23\\7.81\end{bmatrix}$$

1. *Translation*

DoF = 2

What does it preserve?	
Orientation	✓
Scale	✓
Angle	✓
Parallelism	✓
Lines	✓



$$x' = x + t_x$$
$$y' = y + t_y$$

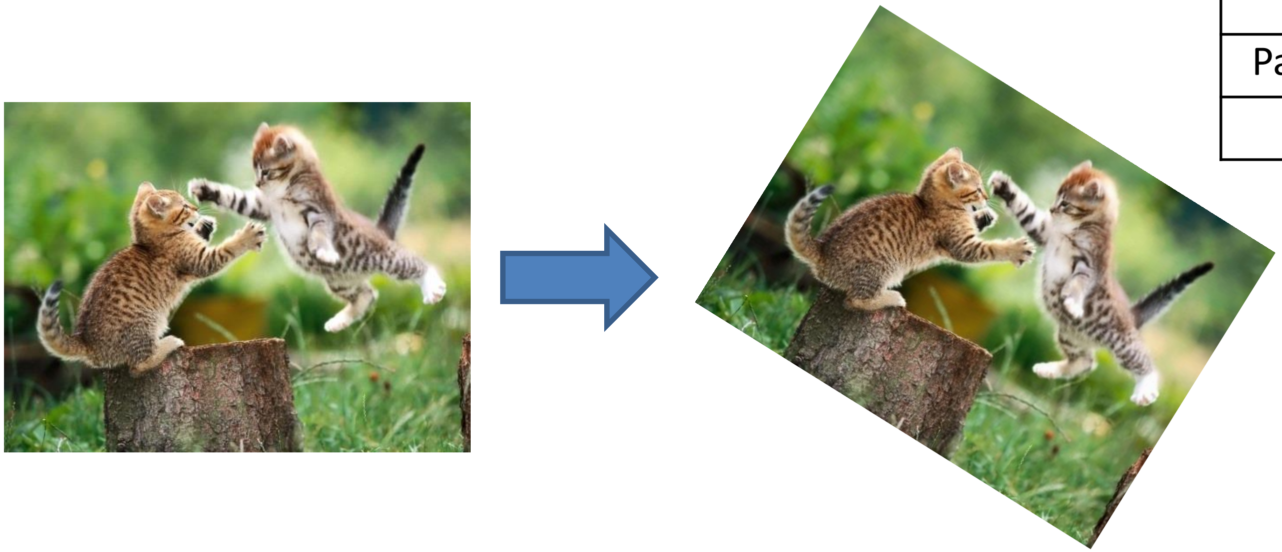
$$\boxed{\mathbf{x}' = \mathbf{x} + \mathbf{t}}$$

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

2. *Rotation*

DoF = 1

What does it preserve?	
Orientation	✖
Scale	✓
Angle	✓
Parallelism	✓
Lines	✓



$$\boxed{\mathbf{x}' = \mathbf{R}\,\mathbf{x}}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3. *Scale*

DoF = 1

What does it preserve?	
Orientation	✓
Scale	✗
Angle	✓
Parallelism	✓
Lines	✓

$$\boxed{\mathbf{x}' = a\,\mathbf{x}}$$

$$\begin{bmatrix} ax \\ ay \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

4. *Shear*

DoF = 2

$$x' = x + a\,y$$
$$y' = y + b\,x$$

Horizontal

Vertical

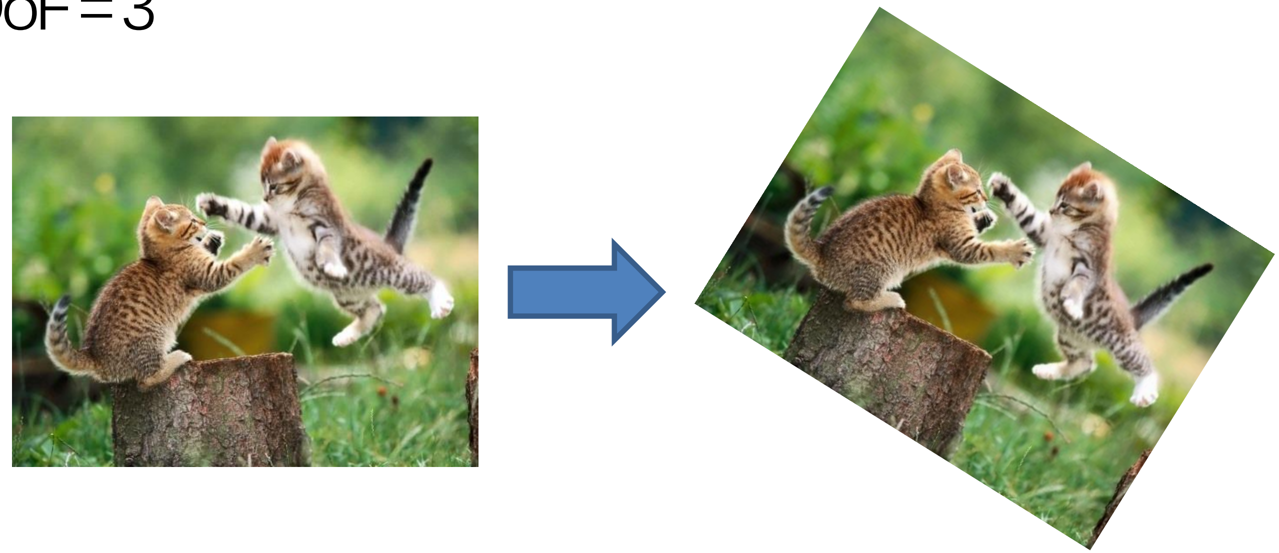
$$\begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

5. *Euclidean*

Rotation + Translation

DoF = 3

What does it preserve?	
Orientation	✖
Scale	✓
Angle	✓
Parallelism	✓
Lines	✓



$$\begin{bmatrix} \cos(\theta) & -sin(\theta) & t_x \\ sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -sin(\theta) & t_x \\ sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

6. *Similarity*

Rotation + Scale + Translation

DoF = 4

What does it preserve?	
Orientation	✗
Scale	✗
Angle	✓
Parallelism	✓
Lines	✓

$$\begin{bmatrix} a\cos(\theta) & -a sin(\theta) & t_x \\ a sin(\theta) & a\cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a\cos(\theta) & -a\sin(\theta) & t_x \\ a\sin(\theta) & a\cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

7. *Affine*

DoF = 6

What does it preserve?	
Orientation	✗
Scale	✗
Angle	✗
Parallelism	✓
Lines	✓

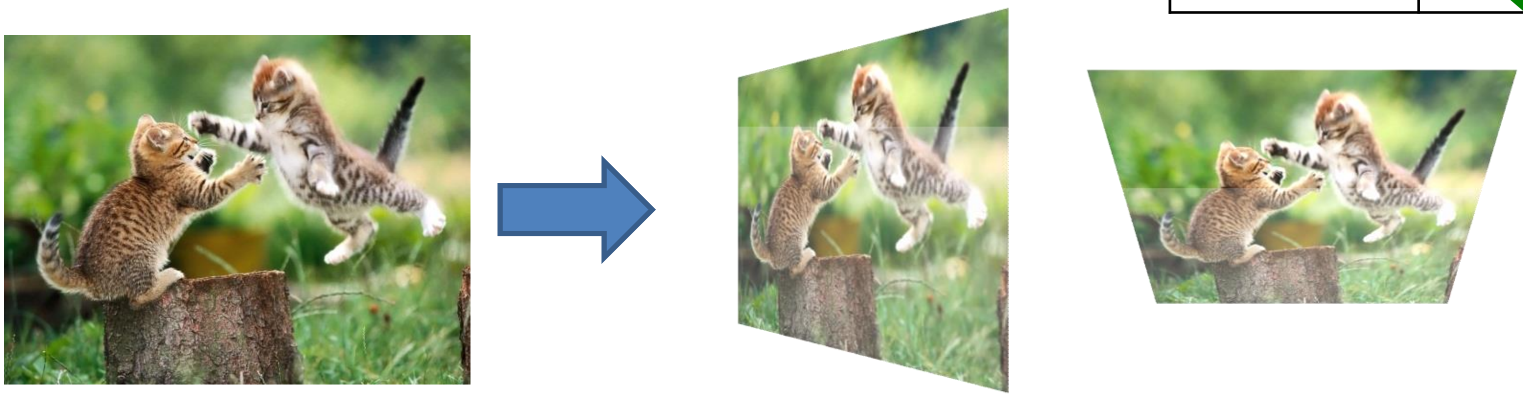
$$\boxed{\mathrm{x}' = \mathbf{A}\,\mathbf{x} + \mathbf{t}}$$

$$\begin{bmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

8. *Projective (Homographies)*

DoF = 8

What does it preserve?	
Orientation	✖
Scale	✖
Angle	✖
Parallelism	✖
Lines	✔

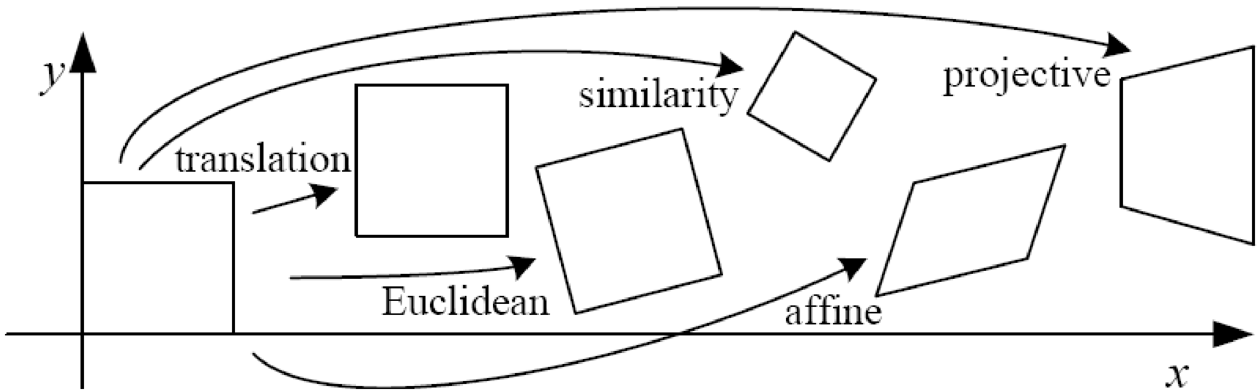


$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$x' = \frac{u}{w}$$

$$y' = \frac{v}{w}$$

2D Image Transformations



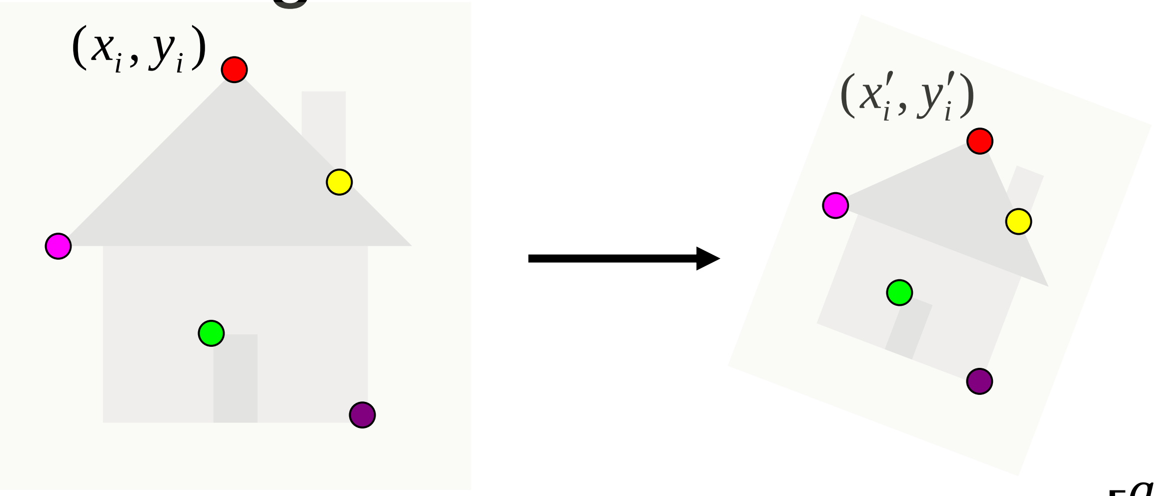
Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles $+\cdots$	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

These transformations are a nested set of groups

- Closed under composition and inverse is a member

Fitting an Affine Transformation

- Assume we know the correspondences, how do we get the transformation?



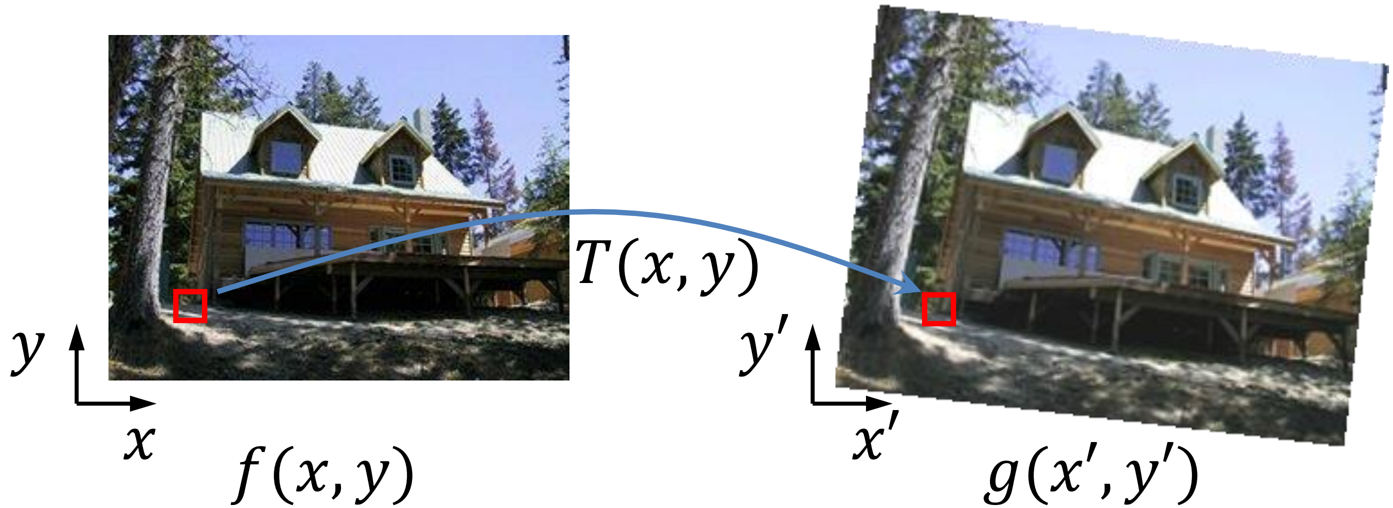
$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} & & \cdots & \cdots & & \\ x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \\ & & \cdots & \cdots & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \cdots \\ x'_i \\ y'_i \\ \cdots \end{bmatrix}$$

Fitting an Affine Transformation

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

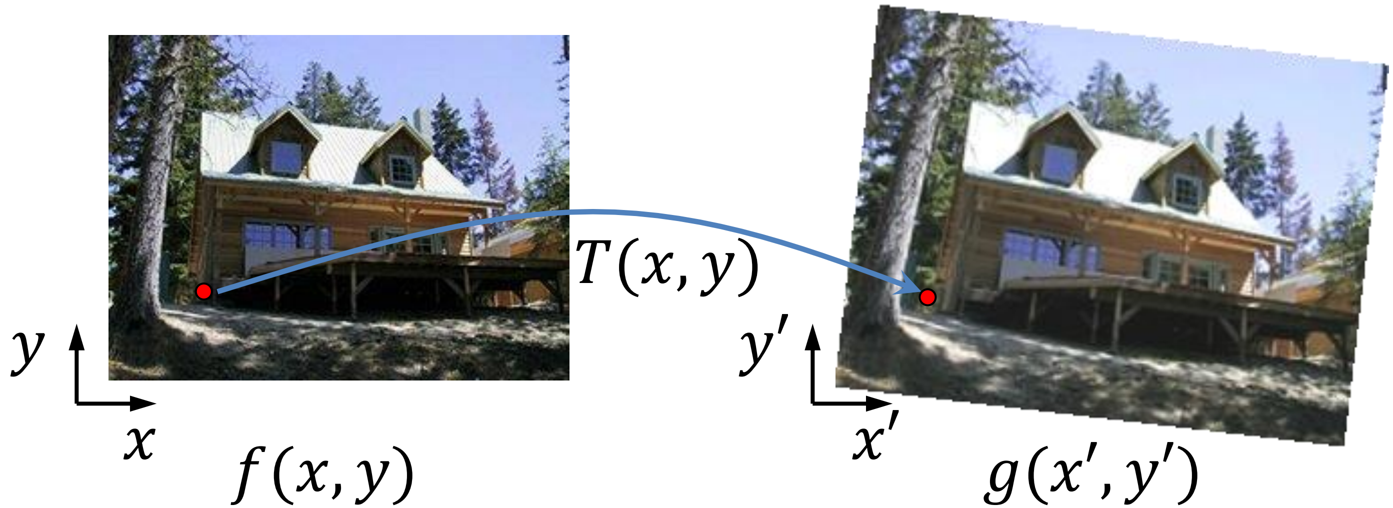
- Linear system with six unknowns
- Each match gives us two linearly independent equations: need at least three to solve for the transformation parameters

Image warping



- Given a coordinate transform $(x', y') = T(x, y)$ and a source image $f(x, y)$, how do we compute a transformed image $g(x', y') = f(T(x, y))$?

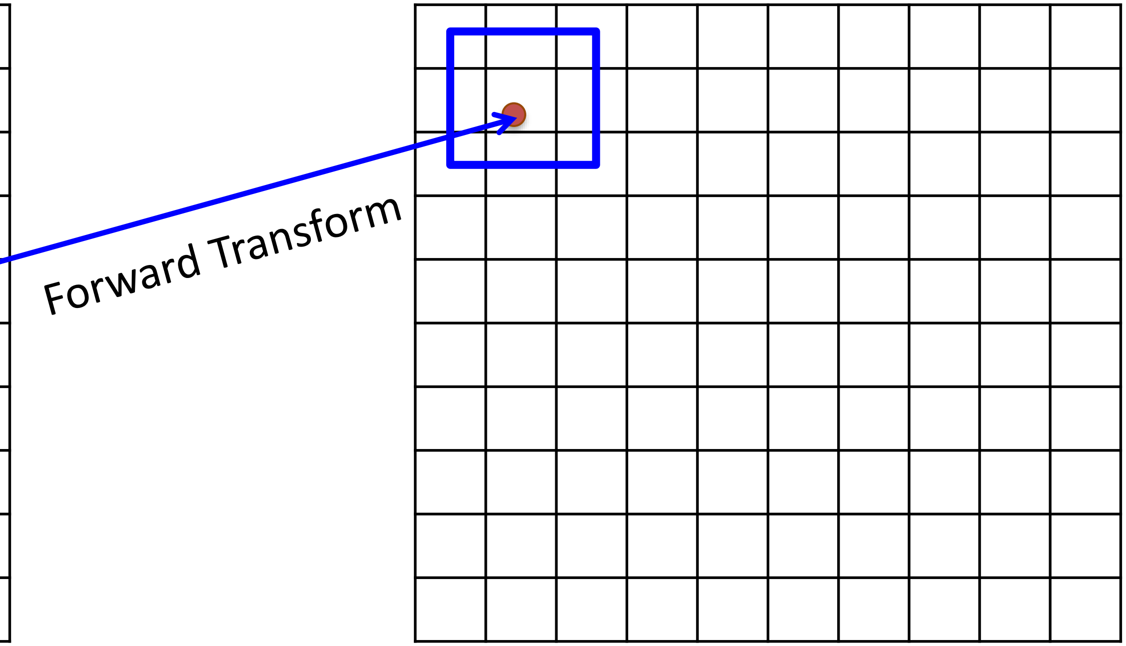
Forward warping



- Send each pixel $f(x,y)$ to its corresponding location
- $(x',y') = T(x,y)$ in the second image

Image Warping

180	184	181	187	185	188	199	189	192	199
186	184	191	195	169	49	15	10	10	11
187	186	191	37	13	21	12	11	11	11
186	187	100	13	18	18	15	12	17	12
189	192	148	15	10	9	9	9	11	12
185	194	14	10	10	8	8	8	10	15
187	88	11	10	10	9	9	10	12	10
190	17	9	8	9	9	9	9	8	11
201	11	9	10	10	9	9	9	11	13
196	11	10	10	10	9	9	8	10	10



$$\begin{bmatrix} 1 & 0.8 & 2 \\ 1.2 & 0.5 & 1.1 \\ 1 & 0.3 & 0.1 \end{bmatrix} \begin{bmatrix} 5 \\ 9 \\ 1 \end{bmatrix} = \begin{bmatrix} 14.2 \\ 11.6 \\ 7.8 \end{bmatrix} \rightarrow \begin{bmatrix} 1.82 \\ 1.49 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1.82 \\ 1.49 \end{bmatrix}$$

Some pixels might remain empty!

Image Warping

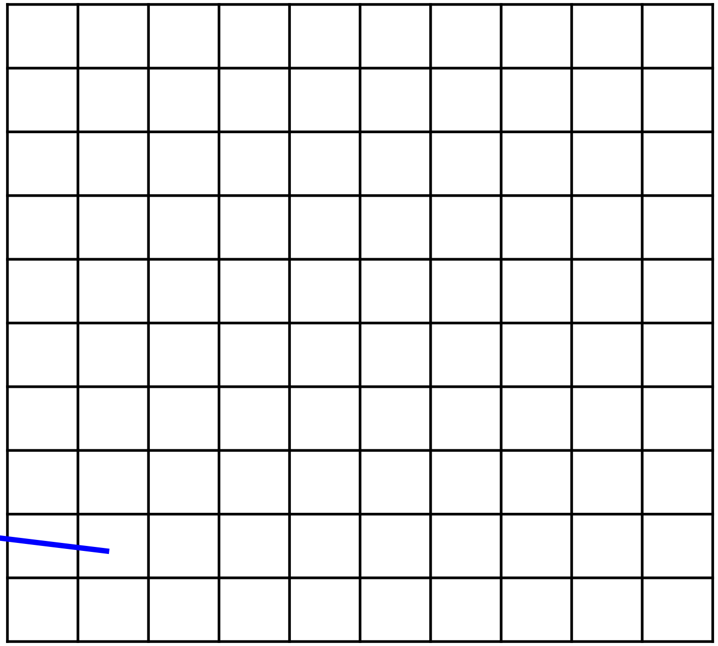
Original Frame

After Rotation

Image Warping

180	184	181	187	185	188	199	189	192	199
186	184	191	195	169	49	15	10	10	11
187	186	191	37	13	21	12	11	11	11
186	187	100	13	18	18	15	12	17	12
189	192	148	15	10	9	9	9	11	12
185	194	14	10	10	8	8	8	10	15
187	88	11	10	10	9	9	10	12	10
190	17	9	8	9	9	9	9	8	11
201	11	9	10	10	9	9	9	11	13
196	11	10	10	10	9	9	8	10	10

Backward Transform

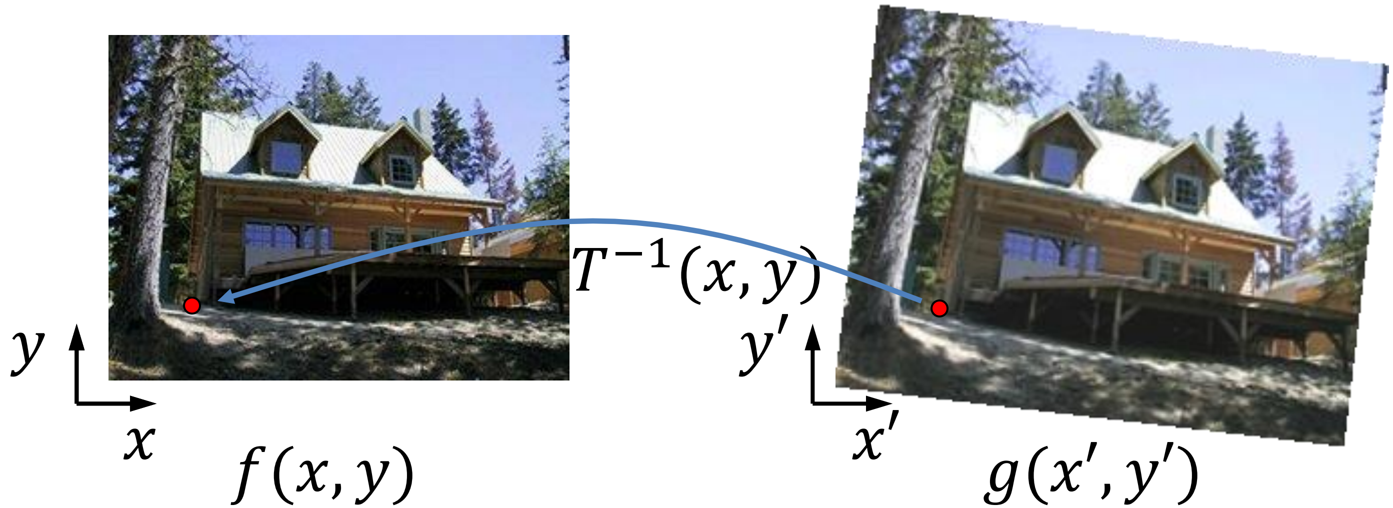


$$\begin{bmatrix} 1 & 0.8 & 2 \\ 1.2 & 0.5 & 1.1 \\ 1 & 0.3 & 0.1 \end{bmatrix} \begin{bmatrix} 5 \\ 9 \\ 1 \end{bmatrix} = \begin{bmatrix} 14.2 \\ 11.6 \\ 7.8 \end{bmatrix} \rightarrow \begin{bmatrix} 1.82 \\ 1.49 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1.82 \\ 1.49 \end{bmatrix}$$

Some pixels might remain empty!

Solution: Backward Transformation

Inverse warping



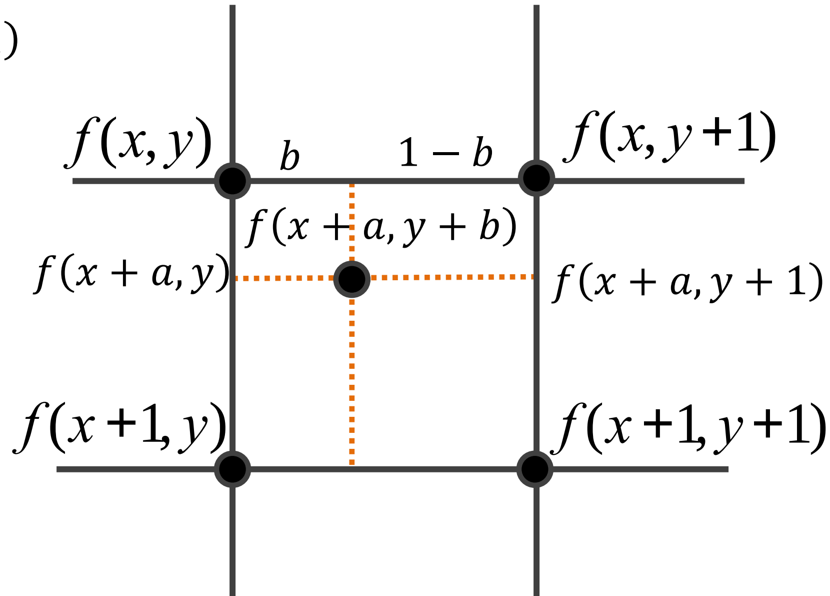
Q: what if pixel comes from “between” two pixels?

Computing values of pixels at fractional positions

Bilinear interpolation:

$$f(x+a, y+b) = (1-b)\, f(x+a, y) + b\, f(x+a, y+1)$$

$$f(x+a, y) = (1-a)\, f(x, y) + a\, f(x+1, y)$$
$$f(x+a, y+1) = (1-a)\, f(x, y+1) + a\, f(x+1, y+1)$$



$$f(x+a, y+b) =$$
$$(1-b)(1-a)\, f(x, y) +$$
$$a(1-b) f(x+1, y) +$$
$$b(1-a)\, f(x, y+1) +$$
$$ab\, f(x+1, y+1)$$

$$f(x+a, y+b)$$
$$= \begin{bmatrix} 1-a & a \end{bmatrix} \begin{bmatrix} f(x,y) & f(x,y+1) \\ f(x+1,y) & f(x+1,y+1) \end{bmatrix} \begin{bmatrix} 1-b \\ b \end{bmatrix}$$

Bicubic interpolation fits a higher order function using a larger area of support.

References

- Geometric Transformations and Image Warping
  Gonzalez: Section 6.2
  Szeliski: Section 2.1 and 3.6