Friday, February 22, 2019 8:02 AM

geometric transformations

- Images often need to be stretched, shrunk, shifted, rotated, magnified, or geometrically transformed in some other way.

Applications:

- correction of lens distortion
- correction for viewing angle
- correction of nonlinear field in MRI
- image registration (lining up for comparison)
- projection onto nonplanar surfaces (or reverse)
- correction of lens designed for hi-res middle for digital zoom

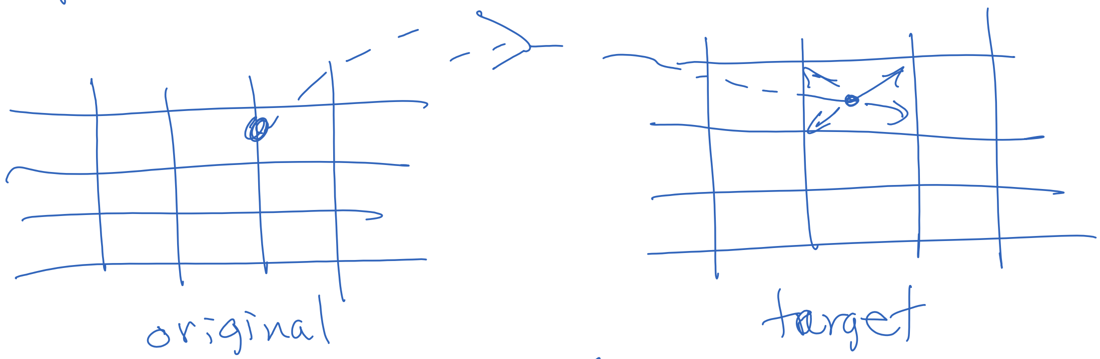
Two algorithms required:

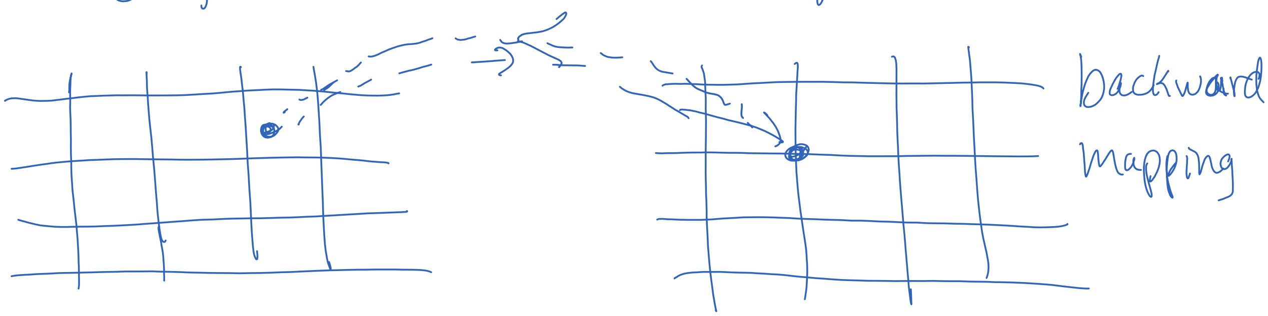
1) mapping that defines transformation from original target coordinates
2) method of interpolating one set of sample values to another set

Interpolation

- integer grid points may not map to integer grid points in transformed image

two options:





Backward mapping is preferable:

- each output pixel is addressed exactly once, in line-by-line fashion
- forward mapping is wasteful — many pixel values may map outside the target image

$\Rightarrow$ in practice, we must define the mapping that takes us from the <u>target</u> pixel locations back to <u>original</u> pixel locations

Since original image location is generally between samples, we must interpolate.

Read 5.1, 5.2, 5.5, 5.7
HW to be posted

Control point specification

A set of corresponding points between two images can be chosen to estimate a mapping from one image to another.
The transformation is chosen to map the points in one image to the corresponding points in the other.
These points can be used to estimate specific parameters of an affine map, polynomial warp, etc.

Let $x_i = a_1 + a_2 x_0 + a_3 y_0 + a_4 x_0 y_0$

$$Y_i = b_1 + b_2 x_0 + b_3 y_0 + b_4 x_0 y_0$$

$$\begin{bmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{in} \end{bmatrix} = \begin{bmatrix} 1 & X_{01} & Y_{01} & X_{01}Y_{01} \\ 1 & X_{02} & Y_{02} & X_{02}Y_{02} \\ & & \vdots & \\ 1 & X_{0n} & Y_{0n} & X_{0n}Y_{0n} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

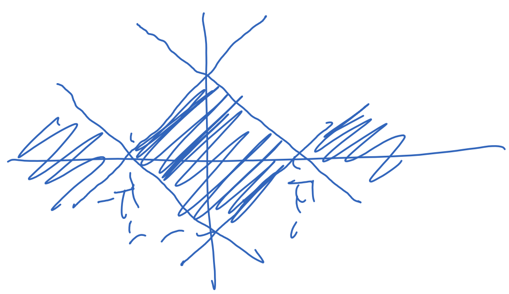
X_{0j} = output x " " " "

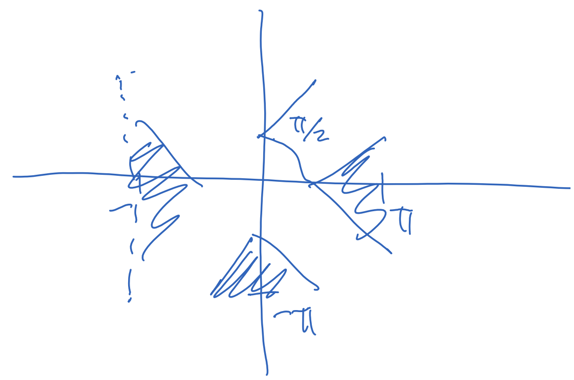
Y_{ij} =

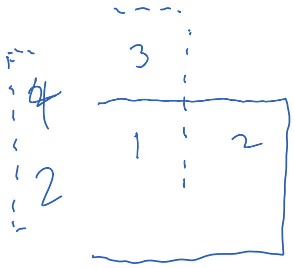
$a = \mathcal{U} \setminus x$

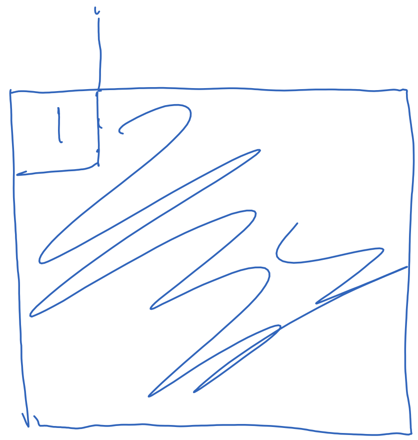
$m' = Lm$

$n' = Ln$






3
4
1
2
2


1