Geometric Image Manipulation

Lecture #1

January 25, 2013

Image Manipulation: Context

- To start with the obvious, an image is a 2D array of pixels
  - The pixel locations represent points on the image plane
  - The pixel values represent measurements of light at those locations
    - Color images : energies by frequency ranges (RGB: three overlapping ranges)
    - Intensity images : average energy across the visible range
  - Your CS410 ray tracers should have taught you about image formation

- To directly compare two images, they should be *registered*
  - Geometrically : to preserve the spatial pattern, which pixel in image 1 "lines up with" each pixel in image 2?
  - Photometrically : if "100" measures a certain amount of energy in image 1, it should imply the same amount of energy in image 2

Geometric Registration

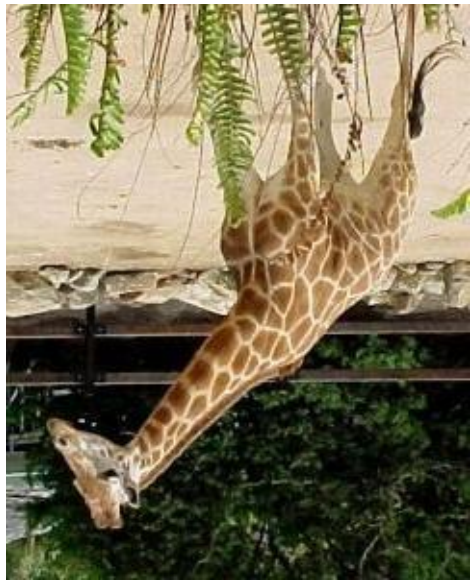
- It's not enough for two matching images to have the same set of pixel values
- They have to be in the same relative positions

Image from CalTech256 data set

Otherwise, these two images match!

Geometric Registration (II)

- Geometric registration finds a mapping that maps one image onto the other
  - We will limit ourselves to linear transformation



We should be able to register these…

Registration formalism

- We denote an image as a 2D function:

$$I(x, y)$$

- Or, in homogeneous coordinates:

$$I(x, y, w)$$

- The goal is to find the transformation matrix G such that:

$$I_i(u, v, w) = I_j\left(G \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\right)$$

Interpolation (foreshadow…)

- Seldom get integer-to-integer mapping.
- Geometry part computes real-valued positions of pixel centers.
- We will worry about how to interpolate values later.

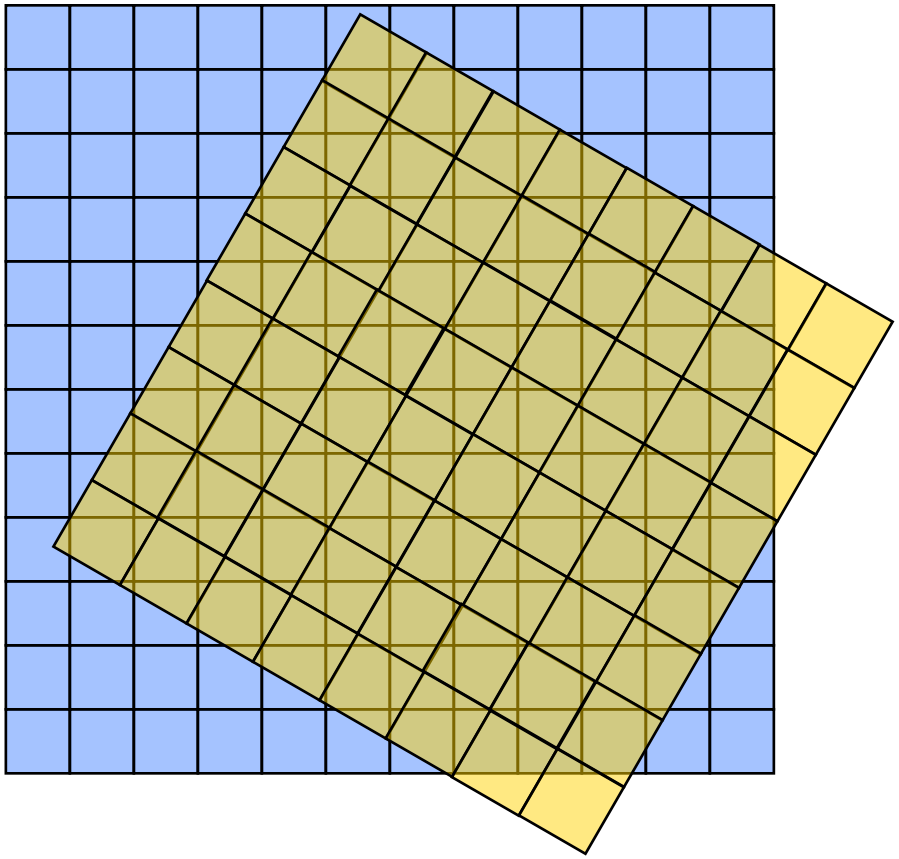


Image Transformations

- The simplest set of transformations are translation, rotation, and scale
  - Together these are called the *similarity* transform.
  - Similarity transforms have 4 degrees of freedom.
- In matrix form these are:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} s1 & 0 \\ 0 & s1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

scale

and...

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

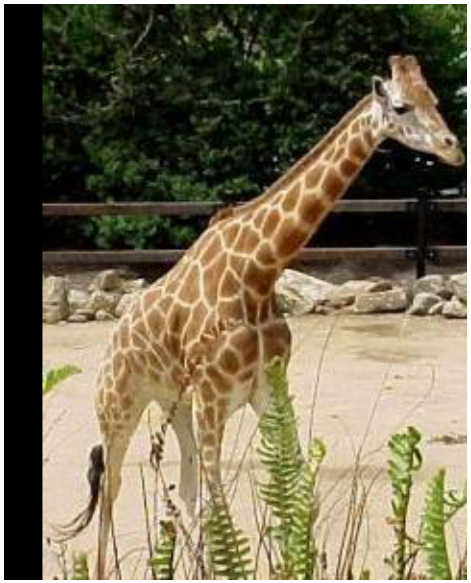
rotation

Image Transformations : Translation

Translation
(note the 2D homogeneous coordinates)

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

Translation Applied to Images



Translate 20 in x

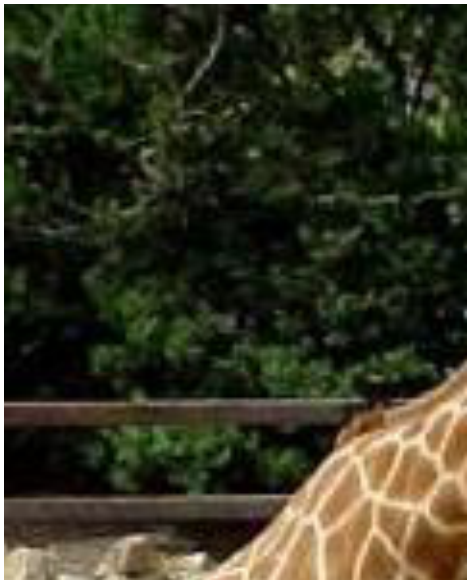
$$\begin{bmatrix} 1 & 0 & 20 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

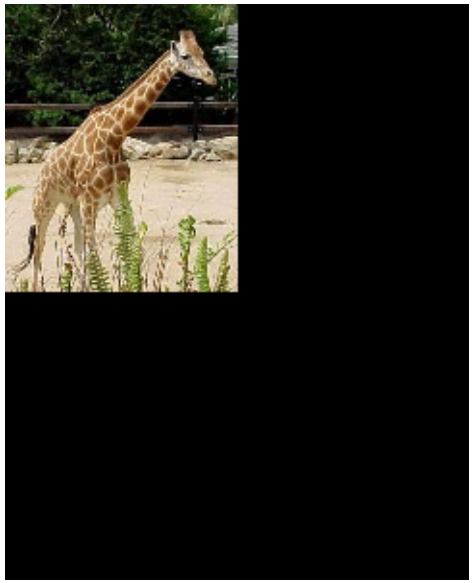
Translate -20 in x

$$\begin{bmatrix} 1 & 0 & -20 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scale Applied to Images

Note the origin





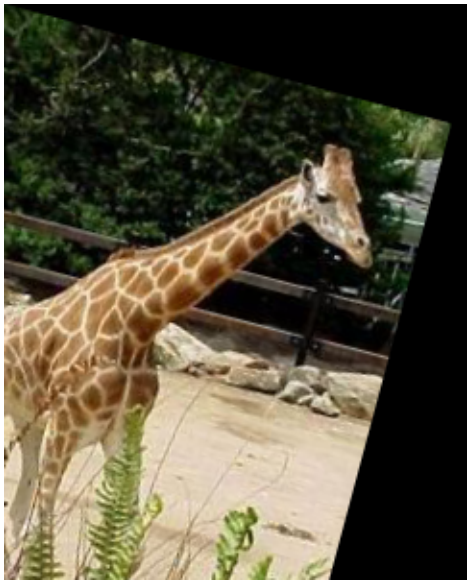
Scale Uniformly by 2

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

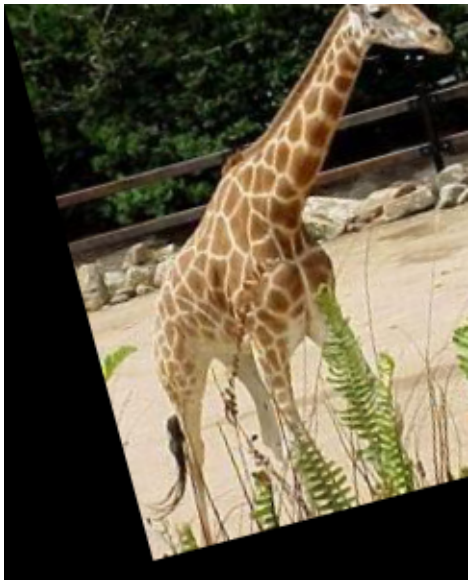
Scale Uniformly by 0.5

$$\begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation Applied to Images



Rotate by 15°



Rotate by -15°

Note that a positive rotation rotates the positive X axis toward the positive Y axis

Remember Composition of Matrices

To rotate by θ around a point (x,y):

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

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

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

Similarity & Affine Transformations

- All the similarity transforms can be combined into one generic matrix:

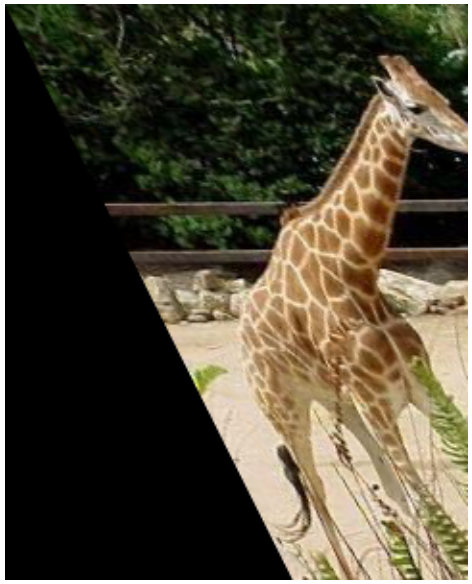
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Hint: diagonal terms are not equal, and b ≠ -d.

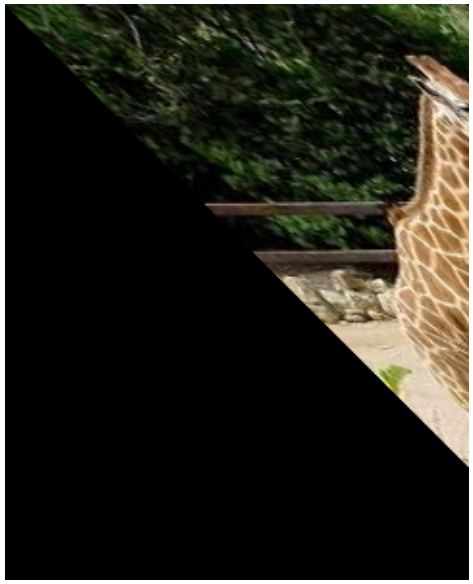
- **But!** This matrix does more. What?
  - hint: two more transformation types included.
  - hint: 6 degrees of freedom (DOF)
- How can you specify this matrix?

This is equivalent to adding two shear parameters (or unequal scaling & one shear).

Affine Examples



$$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Specifying Affine Transformations

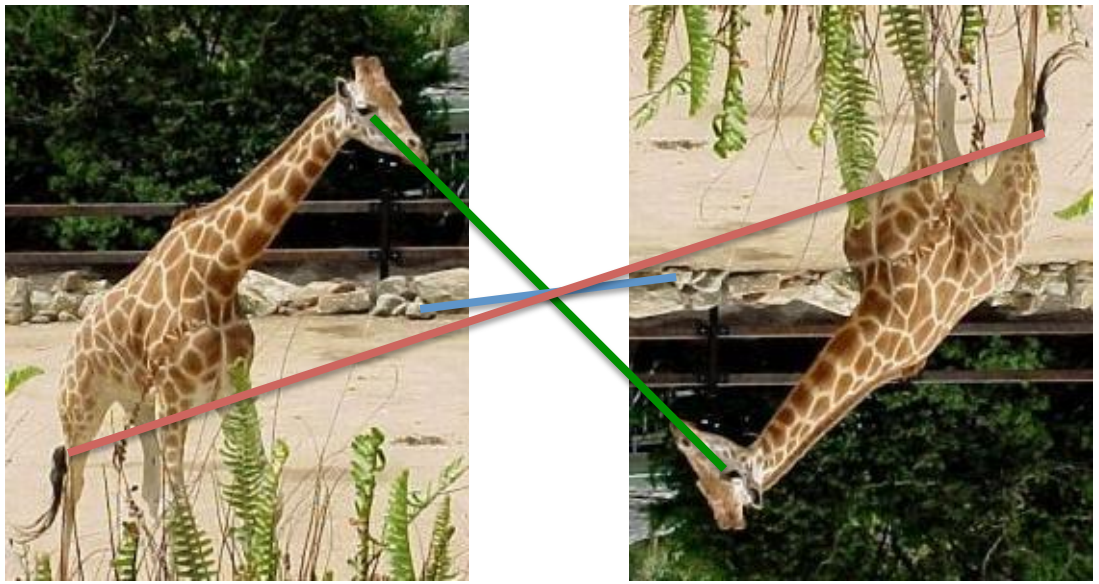
- There are six unknowns in the matrix (*a* through *f*)
- If you specify one point in the source image and a corresponding point in the target image, that yields two equations:

$$u_i = ax_i + by_i + c$$

$$v_i = dx_i + ey_i + f$$

- So providing three point-to-point correspondences specifies an affine matrix

Affine Specification: Example



There is one affine transformation that will map the green point on the right to the green point on the left, and align the red and blue points too.

Solving Affine Transformations

These linear equations can be easily solved:

- WLOG, assume $x_1=y_1=0$
- then $u_1=$ c and $v_1 =$ f
- so:

Calculation of *a, b* & *c* is independent of calculation of *e, f* & *g*.

$$u_2 = ax_2 + by_2 + u_1$$

$$u_3 = ax_3 + by_3 + u_1$$

$$a = \frac{u_2 - u_1 - by_2}{x_2}$$

$$\frac{x_3(u_2 - u_1 - by_2)}{x_2} = u_3 - u_1 - by_3$$

$$\left(\frac{-x_3y_2}{x_2} - y_3\right)b = u_3 - u_1 - \frac{x_3}{x_2}(u_2 - u_1)$$

$$b = \frac{u_3 - u_1 - \frac{x_3}{x_2}(u_2 - u_1)}{\frac{-x_3y_2}{x_2} - y_3} = \frac{x_2(u_3 - u_2) - x_3(u_2 - u_1)}{-x_3y_2 - y_3x_2}$$

Solving Affine (cont.)

- This can be substituted in to solve for a
- The same process with y's solves for d,e,f
- About the WLOG:
  - It was true because you can translate the original coordinate system by $(-x_1, -y_1)$
  - So what do you do to compensate?
- Alternatively, set up a system of linear equations and solve...
  - Will show this for a harder case shortly….

Perspective Transformations

- We can go beyond just affine transformations.
- We can do any perspective transformation of a plane to a plane.
- Therefore, we can model an image as a plane in space, and project it onto any other image.
  - How does this differ from the perspective projection pipeline in CS410?

Perspective Matrix

$$\begin{bmatrix} u' \\ v' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$u = u'/w \;,\; v = v'/w$$

- Why does element [3,3] = 1?
- How many points are needed to specify this matrix?

Solving for Perspective

- Four corresponding points produce eight equations, eight unknowns --- but we can't observe w

$$u_i = {u'_i}/{w_i} = \frac{ax_i + by_i + c}{gx_i + hy_i + 1}$$

$$v_i = {v'_i}/{w_i} = \frac{dx_i + ey_i + f}{gx_i + hy_i + 1}$$

Solving (cont.)

- Multiply to get rid of the fraction…

$$u_i(gx_i + hy_i + 1) = ax_i + by_i + c$$
$$v_i(gx_i + hy_i + 1) = dx_i + ey_i + f$$

- Now, remember that the u's, v's, x's & y's are known; group the unknown terms

$$u_i = ax_i + by_i + c - gx_iu_i - hy_iu_i$$
$$v_i = dx_i + ey_i + f - gx_iv_i - hy_iv_i$$

Solving (III)

- And express the result as a system of linear equations

$$\begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -y_1v_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -y_2u_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2v_2 & -y_2v_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3u_3 & -y_3u_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3v_3 & -y_3v_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4u_4 & -y_4u_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4v_4 & -y_4v_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$

Solving (IV)

- Finally, invert the constant matrix and solve

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -y_1v_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -y_2u_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2v_2 & -y_2v_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3u_3 & -y_3u_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3v_3 & -y_3v_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4u_4 & -y_4u_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4v_4 & -y_4v_4 \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$

Solving (V) : Questions

- Is there always a solution?
- Is the solution always unique?
- Under what conditions is the matrix invertible?

Perspective Image Transforms (Intuition)

- What does the following matrix do?

$$\begin{bmatrix} \sqrt{2} & -\sqrt{2} & 0 \\ \sqrt{2} & \sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix Decomposition

$$\begin{bmatrix} \sqrt{2} & -\sqrt{2} & 0 \\ \sqrt{2} & \sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

original — Scale by 2 — Rotation by 45

- Note that such decompositions are:
  - not unique (why?)
  - difficult to intuit

More Intuitions

- What will the following matrix do?

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & 1 \end{bmatrix}$$

- More specifically, what will it do to the giraffe image?

Check Your Intuitions

- What's going on here?

More Intuition Checking

- Part of what you are seeing is a scale effect
  - positive terms in the bottom row create larger w values, and therefore smaller u,v values
- Something much weirder is also going on:
  - What happens when y = x+1?
  - How do you interpret this geometrically?
  - Isn't the perspective transform linear?
- So how do you select transform matrices?

Perspective Transform of 2D Planes Continued.

- Recall the basic equation for the perspective transform

$$\begin{bmatrix} u' \\ v' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$u = u'/w \quad , \quad v = v'/w$$

- The only practical way to specify an image transform is by providing four point correspondences

Computing Transformations

- Remember how to build a transformation from four point correspondences….

$$\begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -y_1v_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -y_2u_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2v_2 & -y_2v_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3u_3 & -y_3u_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3v_3 & -y_3v_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4u_4 & -y_4u_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4v_4 & -y_4v_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$

Computing...

- So if we want the following mapping:

$(0,0) \rightarrow (0,0)$, $(0,144) \rightarrow (0,144)$,

$(152,0) \rightarrow (152,50)$, $(152,144) \rightarrow (152,94)$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 144 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 144 & 1 & 0 & -20736 \\ 152 & 0 & 1 & 0 & 0 & 0 & -23104 & 0 \\ 0 & 0 & 0 & 152 & 0 & 1 & -7600 & 0 \\ 152 & 144 & 1 & 0 & 0 & 0 & -23104 & -21888 \\ 0 & 0 & 0 & 152 & 144 & 1 & -14288 & -13536 \end{bmatrix}$$

…More Computing...

What does This say about x?

$$\begin{bmatrix} -.014 & -.023 & .007 & .023 & .014 & .022 & -.007 & -.022 \\ -.007 & 0 & .007 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -.002 & -.014 & .002 & .007 & .002 & .014 & -.002 & -.007 \\ -.007 & -.007 & .007 & .007 & .007 & 0 & -.007 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ .000 & -.000 & .000 & .000 & .000 & .000 & -.000 & .000 \\ -.000 & 0 & .000 & 0 & .000 & 0 & -.000 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 144 \\ 152 \\ 50 \\ 152 \\ 94 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} 3.274 \\ 0 \\ 0 \\ 1.077 \\ 1 \\ 0 \\ .01497 \\ 0 \end{bmatrix}$$

Remember the earlier WLOG? c = u_1, …

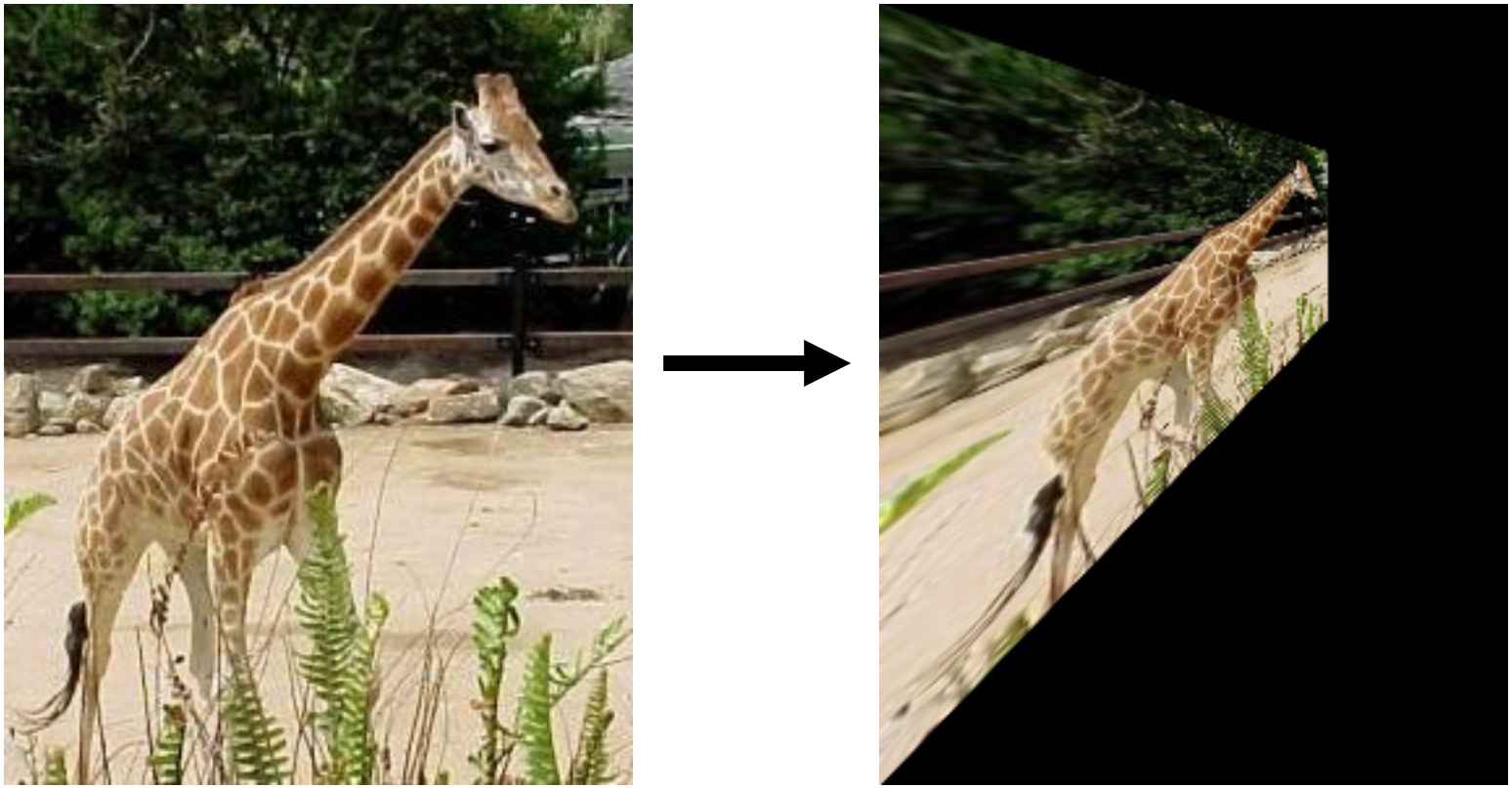
M^{-1}

u&v vector

How does This alter it?

…yields

$$\begin{bmatrix} 3.274 & 0 & 0 \\ 1.077 & 1 & 0 \\ .01497 & 0 & 1 \end{bmatrix}$$



Geometric Transformation : Review

- Goal: $I_i(u,v,w) = I_j(G \bullet [x,y,1]^T)$
- Transformation classes:
  - Similarity (4 DoF : translation, rotation, scale)
  - Affine (6 DoF)
  - Perspective (8 DoF)
- Specified via point correspondences

Still To Do….

- Photometric registration
- Interpolation
  - Transformations are not integer to integer

- Then we can tackle matching under various transformation classes