Example - affine

• Recall that the image of a planar surface undergoing a small rotation can be approximated by an affine transformation

• Find the parameters of the affine transformation via least squares fitting
  – We’ll need some corresponding points

• Then use it to transform image 1 to map it to image 2
Example - affine

• Write the system of linear equations $Ax=b$ for least squares fitting of an affine transformation; i.e., give the elements of the matrices $A$, $x$, and $b$.

• Solution:
  – An affine transformation has the form

$$\begin{bmatrix} x_B \\ y_B \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ 1 \end{bmatrix}$$

  or

$$x_B = a_{11}x_A + a_{12}y_A + t_x \quad \text{or} \quad x_B = a_{11}x_A + a_{12}y_A + t_x$$

$$y_B = a_{21}x_A + a_{22}y_A + t_y$$

  – There are 6 unknowns so we need at least 3 corresponding points.
  – The system of linear equations is $Ax=b$ where

$$A = \begin{bmatrix} x_A^{(1)} & y_A^{(1)} & 0 & 0 & 1 & 0 \\ 0 & 0 & x_A^{(1)} & y_A^{(1)} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & x_A^{(N)} & y_A^{(N)} & 0 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ t_x \\ t_y \end{bmatrix}, \quad b = \begin{bmatrix} x_B^{(1)} \\ y_B^{(1)} \\ \vdots \\ x_B^{(N)} \\ y_B^{(N)} \end{bmatrix}$$
Example - affine

• Find corresponding points between and estimate the affine transformation from image 1 to image 2 by solving the system of linear equations $Ax=b$

• Solution:
  – First, use imtool or imshow to find corresponding points by hand
Example - affine

- Put into pA, pB (2xN matrices)

```matlab
clear all
close all

% Here are corresponding points (x;y)
pA = [
    120 422 106 557;
    28 6 472 400];
pB = [
    201 501 81 536;
    6 28 420 462];

I1 = imread('book1.jpg');
I2 = imread('book2.jpg');

N = size(pA,2);
imshow(I1,[]);
for i=1:N
    rectangle('Position', [pA(1,i)-4 pA(2,i)-4 8 8], 'FaceColor', 'r');
end
figure, imshow(I2,[]);
for i=1:N
    rectangle('Position', [pB(1,i)-4 pB(2,i)-4 8 8], 'FaceColor', 'r');
end
```
Example - affine

- Create matrices A and b
- Solve for x
- Unload values out of x and put into a 3x3 transformation matrix

```matlab
% Calculate the transformation T from I1 to I2; ie p2 = T p1.
A = zeros(2*N,6);
for i=1:N
    A( 2*(i-1)+1, :) = [ pA(1,i)  pA(2,i)  0       0       1  0];
    A( 2*(i-1)+2, :) = [ 0        0        pA(1,i) pA(2,i) 0  1];
end
b = reshape(pB, [], 1);

x = A\b;

T = [ x(1)  x(2)    x(5);
     x(3)  x(4)    x(6);
     0     0       1];
```
Example - affine

• Transform image 1 to map it to image 2, and display the resulting image (it should closely match image 2)
  – As discussed in the lecture, we generate the transformed image by scanning through the output image point by point, and calculate where in the source image the value should come from
  – So we actually use the inverse transformation, $T^{-1}$
Example - affine

- Matlab code:

```matlab
% Get the inverse transformation from image2 to image1.
Tinv = inv(T);  % p1 = Tinv * p2

% Now warp I1 using the transformation above. It should match I2.
I3 = zeros(size(I2));
for x2=1:size(I2,2)
    for y2=1:size(I2,1)
        p2 = [x2; y2; 1];
        p1 = Tinv * p2;
        p1 = p1/p1(3);

        % We'll just pick the nearest point to p1 (better way is to
        % interpolate).
        x1 = round(p1(1));
        y1 = round(p1(2));

        if x1>0 && x1<=size(I1,2) && y1>0 && y1<=size(I1,1)
            I3(y2,x2) = I1(y1,x1);
        end
    end
end
```