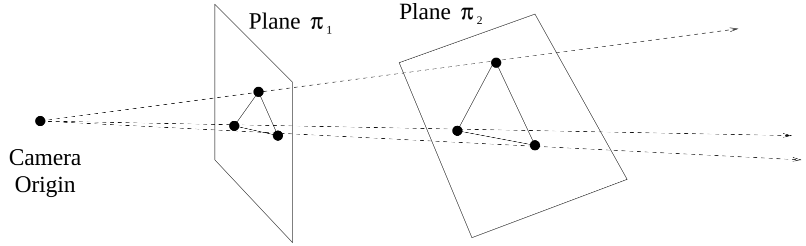
Homography and Transfer

Robert B. Fisher
School of Informatics
University of Edinburgh

 Slide credit: Bob Fisher & Vittorio Ferrari

Homography I

Projection: A non-singular (ie. invertible) linear transformation **P** that maps points from one plane to another



Plane π_2 shapes observed in a 3D position project onto image plane π_1 (2D $\rightarrow$ 2D)
$\mathbf{P}$: $\pi_2 \rightarrow \pi_1$

 Slide credit: Bob Fisher & Vittorio Ferrari

Homography II

Accounts for 8 degrees of freedom (2 more than affine transformations), including rotation, translation, scale, shear.

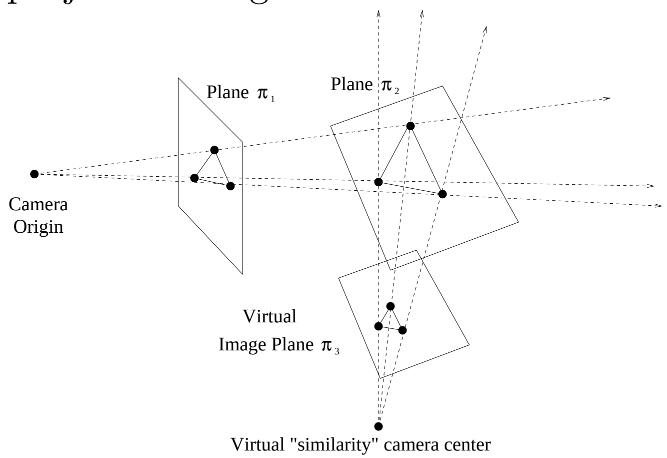
To do properly, use homogeneous coordinates
Augment point positions $(x, y)'$ to $(x, y, 1)'$

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

 Slide credit: Bob Fisher & Vittorio Ferrari

Projective Transfer

Can use projective transfer to map observed projective image to another view



 Slide credit: Bob Fisher & Vittorio Ferrari

Use homogeneous coordinates
$\mathbf{P}_{21}$: $\pi_2 \rightarrow \pi_1$ (ie. copy scene plane into image plane)
$\mathbf{P}_{23}$: $\pi_2 \rightarrow \pi_3$ (ie. copy scene plane into new plane)
Therefore $\mathbf{P}_{13} = \mathbf{P}_{23} \, (\mathbf{P}_{21})^{-1}$: $\pi_1 \rightarrow \pi_3$

Slide credit: Bob Fisher & Vittorio Ferrari

Projection matrix P

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

Slide credit: Bob Fisher & Vittorio Ferrari

Estimating P:$(u, v) \rightarrow (x, y)$

Direct Linear Transform Method

Given: $N \geq 4$ matched points: $\{((x_i, y_i), (u_i, v_i))\}_{i=1}^{N}$

Least square estimate of **P**

Let $\vec{p}' = (p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{23}, p_{31}, p_{32}, p_{33})'$
Let $\mathbf{A}_i =$

$$\begin{bmatrix} 0 & 0 & 0 & -u_i & -v_i & -1 & y_i u_i & y_i v_i & y_i \\ u_i & v_i & 1 & 0 & 0 & 0 & -x_i u_i & -x_i v_i & -x_i \end{bmatrix}$$

Slide credit: Bob Fisher & Vittorio Ferrari

Estimating P cont.

Construct $\mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \\ \dots \\ A_N \end{bmatrix}$

Compute SVD($\mathbf{A}$) = $\mathbf{UDV}'$
$\vec{p}$ is last column of $\mathbf{V}$ (eigenvector of smallest eigenvalue of $\mathbf{A}$)
Repack $\vec{p}$ back into matrix $\mathbf{P}$

See `esthomog.m` and Sec. 4.1 of Harley & Zisserman

Slide credit: Bob Fisher & Vittorio Ferrari

Transfer/Remapping Algorithm

Input image $I(x, y)$
Remapped image $R(u, v)$

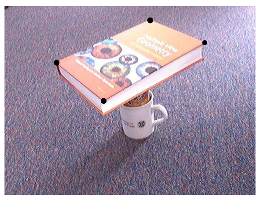
If homography $\mathbf{P}$ between planes known, then can map (u, v) onto (x, y) using:

$$\begin{pmatrix} \lambda x \\ \lambda y \\ \lambda \end{pmatrix} = \mathbf{P} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

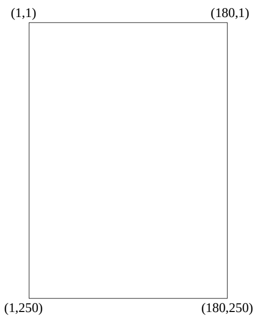
for each (u, v)
  get (x, y) from projection divided by λ
  $R(u, v) = I(x, y)$

See `remap.m`

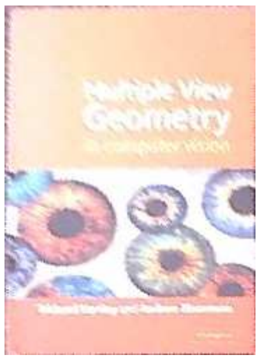
Projective Transfer Example



ORIGINAL



TARGET



REMAPPED

Lecture Overview

- Homography - for mapping between planes
- How to estimate a homography
- Using a homography for projective mapping