IMAGE-BASED MODELING AND RENDERING

6. TRANSFORMATION AND CALIBRATION

I-Chen Lin, Dept. of CS, National Chiao Tung University

Objective

- Image transformation.
- Plane-based camera calibration from homography.

Plenty of slides are modified from the reference lectures:

• Image-based Modeling and Rendering, SIGGRAPH'99 course notes.

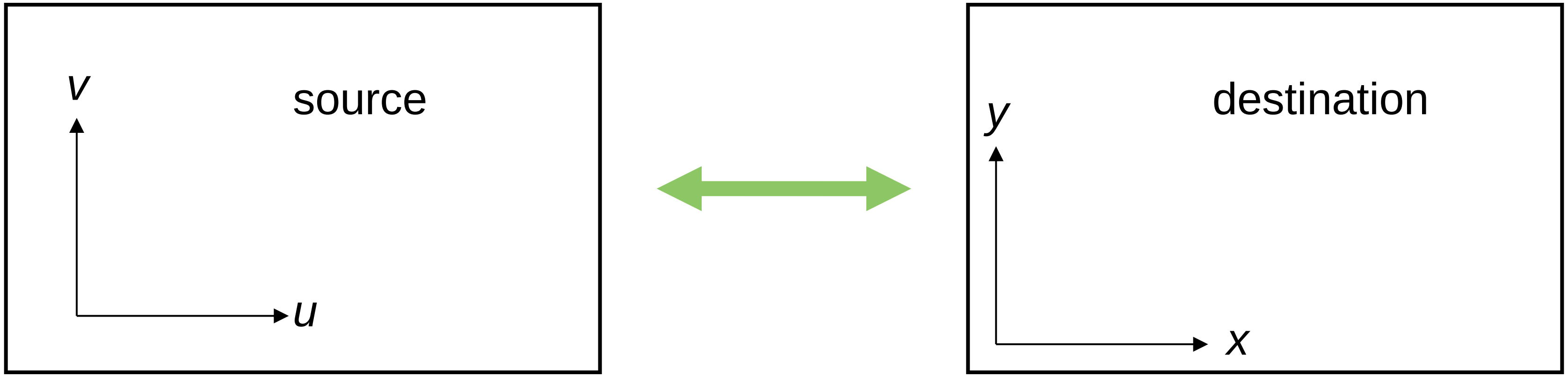
• R. Szeliski, "Image Alignment and Stitching: A Tutorial," Foundations and Trends in Computer Graphics and Vision, vol.2, no.1., pp.1-104.

• P. F. Sturm and S. J. Maybank, "On Plane-based Camera Calibration: A General Algorithm, Singularities and Applications", Proc. CVPR'09.

Image Warping and Transformation

- Rearranging pixels of a picture.
  - It’s useful for both image processing and for computer graphics (namely, for texture mapping).
- Finding corresponding points in the source and destination images.
- This function is called the “mapping” or “transformation”.

Image Warping (cont.)



Forward mapping : $(x, y) = f\,(u, v)$

Inverse mapping : $(u, v) = f'\,(x, y)$

Mapping types

- How to create the mapping systematically?

- Simple mappings
  - affine mapping

  - projective mapping

  - bilinear mapping

  - ………

Affine Mappings

- $\boldsymbol{u} = a\boldsymbol{x}+b\boldsymbol{y}+c$
- $\boldsymbol{v} = d\boldsymbol{x}+e\boldsymbol{y}+f$

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

- A combination of 2-D scale, rotation, and translation transformations.
- Allows a square to be distorted into any parallelogram.
- 6 degrees of freedom.
- Inverse is of same form (is also affine).

Projective Mappings

- $\boldsymbol{u} = (a\boldsymbol{x}+b\boldsymbol{y}+c)/(g\boldsymbol{x}+h\boldsymbol{y}+i)$
- $\boldsymbol{v} = (d\boldsymbol{x}+e\boldsymbol{y}+f)/(g\boldsymbol{x}+h\boldsymbol{y}+i)$

$$\begin{bmatrix} uq \\ vq \\ q \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\boldsymbol{u = uq/q \ , \ v = vq/q}$$

- Linear numerator and denominator.
- If g=h=0 then you get affine as a special case.

- Allows a square to be distorted into any quadrilateral.
- 8 degrees of freedom.
- Inverse is of same form (is also projective)

Projective Mappings

- Every point correspondence can have two equations:
  - $x(XA_{31} + YA_{32} + A_{33}) = (XA_{11} + YA_{12} + A_{13})$
  - $y(XA_{31} + YA_{32} + A_{33}) = (XA_{21} + YA_{22} + A_{23})$

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

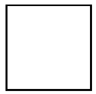
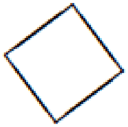
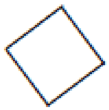
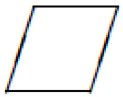
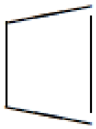
- After rearrangement, a pair of quadrilateral:

$$\begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2 \\ X_3 & Y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -x_3Y_3 & -x_3 \\ 0 & 0 & 0 & X_3 & Y_3 & 1 & -y_3X_3 & -y_3Y_3 & -y_3 \\ X_4 & Y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -x_4Y_4 & -x_4 \\ 0 & 0 & 0 & X_4 & Y_4 & 1 & -y_4X_4 & -y_4Y_4 & -y_4 \end{bmatrix} * \begin{bmatrix} A_{11} \\ A_{12} \\ A_{13} \\ A_{21} \\ A_{22} \\ A_{23} \\ A_{31} \\ A_{32} \\ A_{33} \end{bmatrix} = 0$$

min E = Ux, subject to $|x|=1$

The x that minimize E is eigenvector e_1 of U^TU

Properties of Image Transformation

Name	Matrix	Number of 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	

Is a projective mapping related to the rotation and projection of a plane?

Performing an Image Warp

- Source scanning:

```
for v = vmin to vmax
   for u = umin to umax
         x = x(u,v)
         y = y(u,v)
         copy pixel at source[u,v] to dest[x,y]
```

- Destination scanning:

```
for y = ymin to ymax
   for x = xmin to xmax
         u = u(x,y)
         v = v(x,y)
         copy pixel at source[u,v] to dest[x,y]
```

Vanishing Points and Parallel Lines

- The intersection of projection of a set of parallel lines in space onto the picture plane.

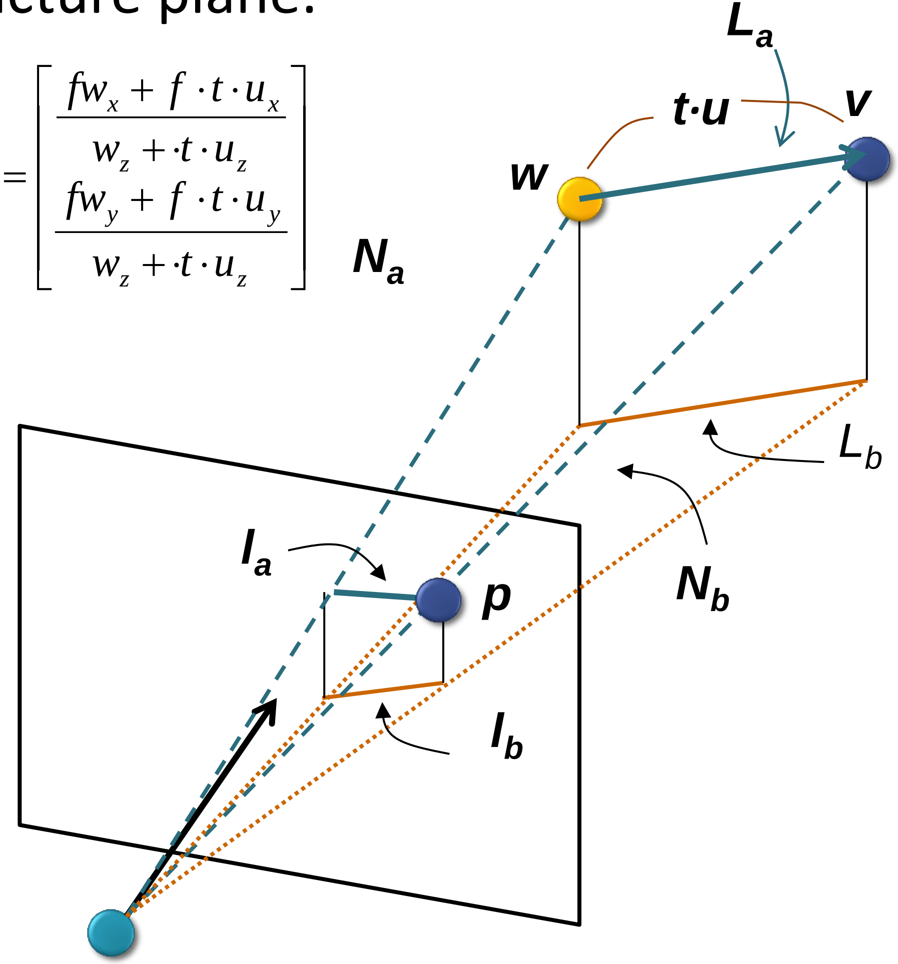
$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} + t \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

$$\begin{bmatrix} p_x(t) \\ p_y(t) \end{bmatrix} = \begin{bmatrix} \frac{fv_x(t)}{v_z(t)} \\ \frac{fv_y(t)}{v_z(t)} \end{bmatrix} = \begin{bmatrix} \frac{fw_x + f \cdot t \cdot u_x}{w_z + \cdot t \cdot u_z} \\ \frac{fw_y + f \cdot t \cdot u_y}{w_z + \cdot t \cdot u_z} \end{bmatrix}$$

$$t \to \infty, \quad \begin{bmatrix} p_x(\infty) \\ p_y(\infty) \end{bmatrix} = \begin{bmatrix} \frac{f \cdot u_x}{u_z} \\ \frac{f \cdot u_y}{u_z} \end{bmatrix} \quad \frac{p_x(\infty)}{p_y(\infty)} = \frac{u_x}{u_y}$$

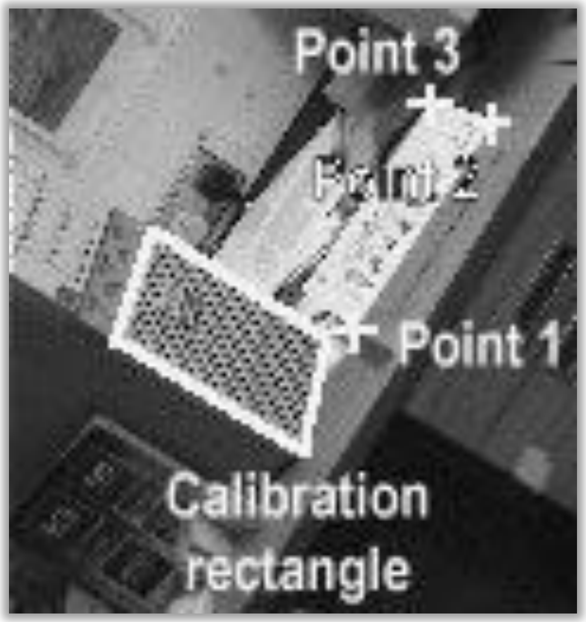
$$\|u\| = 1 \qquad u_x = \frac{p_x}{p_y} u_y \qquad u_z = \frac{f \cdot u_y}{p_y}$$

$$u_y^2 = \frac{1}{\frac{p_x^2}{p_y^2} + 1 + \frac{f^2}{p_y^2}}$$



Plane-based Camera Calibration

- P. F. Sturm and S. J. Maybank, “On Plane-based Camera Calibration: A General Algorithm, Singularities and Applications”, Proc. CVPR’99.



Plane-based Camera Calibration (cont.)

- Planar calibration patterns are cheap and easy to produce.
  - E.g. using a laser printer output.

- This work calibrates a camera with possibly
  - variable intrinsic parameters and position
  - Arbitrary number of calibration planes and camera views.

- The procedure can be solved in two steps (linear equations).
  - Estimate the 2D-to-2D projections (transformation).
  - Extract the intrinsic parameters.

Plane-based Camera Calibration (cont.)

- Projection matrix P incorporates K, R, t :

$$P \sim KR\left(I_3 \quad | \quad -t\right)$$

~ : equality up to a non zero scale factor
I_3 : 3×3 identity matrix
K : 3 ×3 calibration matrix

$$\mathrm{K} = \begin{pmatrix} \tau f & s & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

f : the focal length
τ : the aspect ratio
s : the skew factor is ignored
(u_0, v_0) : the principal point

- Suppose that the calibration plane is the plane $Z = 0$, and drop the 3rd column.

$$H \sim KR\begin{pmatrix} 1 & 0 & \\ 0 & 1 & -t \\ 0 & 0 & \end{pmatrix}$$

H : the 3×3 **homography** can be estimated from 4 or more point or line correspondences.

Plane-based Camera Calibration (cont.)

- The objective is to extract K from H through the IAC matrix ω.

$$\omega \sim (KK^T)^{-1} = K^{-T}K^{-1} \sim \begin{pmatrix} 1 & 0 & -u_0 \\ 0 & \tau^2 & -\tau^2 v_0 \\ -u_0 & -\tau^2 v_0 & \tau^2 f^2 + u_0^2 + \tau^2 v_0^2 \end{pmatrix}$$

the skew s is ignored

Since

$$K^{-1}H \sim R\begin{pmatrix} 1 & 0 & \\ 0 & 1 & -t \\ 0 & 0 & \end{pmatrix} = \begin{pmatrix} R_1 & R_2 & (-t_1R_1 - t_2R_2 - t_3R_3) \end{pmatrix}$$

R_i : the i^{th} column of R
t_j : the j^{th} element of t

$$H^T\omega H \sim H^TK^{-T}K^{-1}H \sim \begin{pmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ -t_1 & -t_2 & t^Tt \end{pmatrix}$$

$$H^T\omega H = \begin{pmatrix} h_1^T \\ h_2^T \\ h_3^T \end{pmatrix}\omega\begin{pmatrix} h_1 & h_2 & h_3 \end{pmatrix}$$

h_i : the i^{th} column of H

Plane-based Camera Calibration (cont.)

- We can therefore find two constraints among h_1, h_2, ω (invariant to t).

$$h_1^T \omega\, h_1 - h_2^T \omega\, h_2 = 0\,, \quad h_1^T \omega\, h_2 = 0$$

- The variables in ω are x = (ω_{11} , ω_{22} , ω_{13} , ω_{23} , ω_{33}).

$$\omega \sim \begin{pmatrix} 1 & 0 & -u_0 \\ 0 & \tau^2 & -\tau^2 v_0 \\ -u_0 & -\tau^2 v_0 & \tau^2 f^2 + u_0^2 + \tau^2 v_0^2 \end{pmatrix}$$

- After rearrangement, we can solve the variables of ω with a given H, in the form of $Ax=0$.

$$\tau^2 = \frac{\omega_{22}}{\omega_{11}} \qquad u_0 = \frac{-\omega_{13}}{\omega_{11}} \qquad v_0 = \frac{-\omega_{23}}{\omega_{22}} \qquad f^2 = \frac{\omega_{11}\omega_{22}\omega_{33} - \omega_{22}\omega_{13}^2 - \omega_{11}\omega_{23}^2}{\omega_{11}\omega_{22}^2}$$

Plane-based Camera Calibration (cont.)

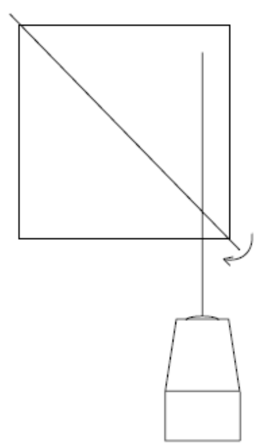
- Let a_i be the i^{th} column of A.

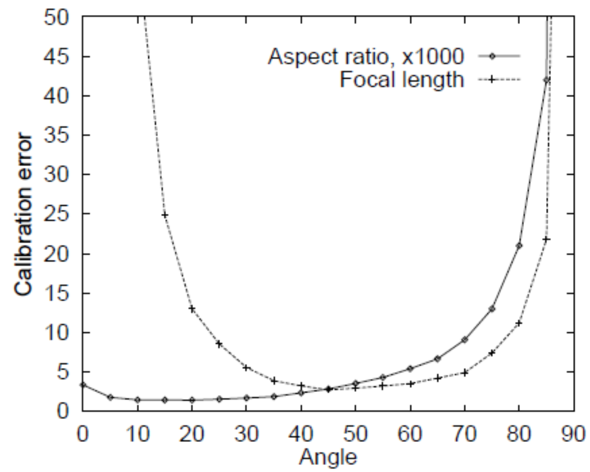
$$\omega_{11}a_1 + \omega_{22}a_2 + \omega_{13}a_{13} + \omega_{23}a_4 + \omega_{33}a_5 = 0$$

- Given prior knowledge, such as τ, the reduced linear system becomes

$$\omega_{11}(a_1 + \tau^2 a_2) + \omega_{13}a_{13} + \omega_{23}a_4 + \omega_{33}a_5 = 0$$

- The simulation error (with gaussian noise) (w.r.t angles)





K, *H* and Extrinsic Parameters

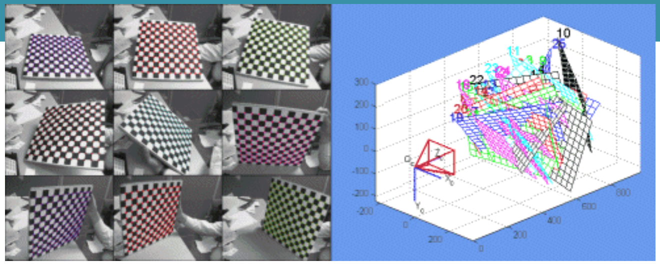
- With estimated *H*, *K*,

$$H \sim KR\begin{pmatrix} 1 & 0 & \\ 0 & 1 & -t \\ 0 & 0 & \end{pmatrix}$$

$$K^{-1}H \sim R\begin{pmatrix} 1 & 0 & \\ 0 & 1 & -t \\ 0 & 0 & \end{pmatrix} = \begin{pmatrix} R_1 & R_2 & (-t_1R_1 - t_2R_2 - t_3R_3) \end{pmatrix}$$

R_i : the i^{th} column of R
t_j : the j^{th} element of t

Camera Calibration



- Public camera calibration tools
  - Camera calibration toolbox for OpevnCV.
  - Camera calibration toolbox for Matlab.
    - http://www.vision.caltech.edu/bouguetj/calib_doc/
  - A flexible new technique for camera calibration.
    - http://research.microsoft.com/~zhang/calib/
    - Z. Zhang. A flexible new technique for camera calibration. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.
  - Tsai’s camera model.
    - http://www.cs.cmu.edu/~rgw/TsaiDesc.html
    - "A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", Roger Y. Tsai, IEEE J. Robotics and Automation, Vol. RA-3, No. 4, 1987, pages 323-344.