Human Computer Interaction

6. Fundamental of Vision-based Interfaces (B)

National Chiao Tung Univ, Taiwan

By: I-Chen Lin, Assistant Professor

Goals

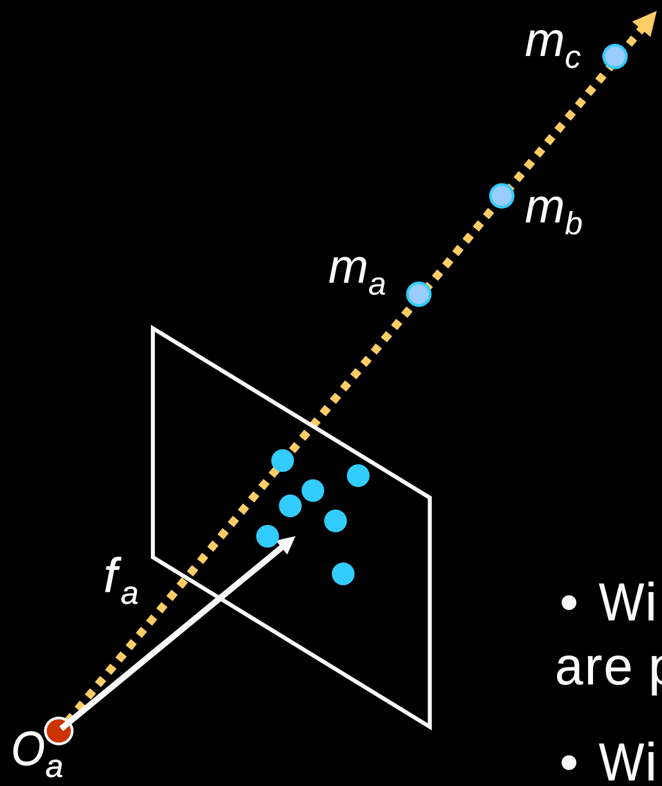
- From 2D to 3D, learn the basic 3D reconstruction.
- Efficient (and simple ?) approaches for real-time user interfaces.

Outline

- Feature extraction
  - Color matching
  - Grouping or clustering
  - Silhouette & foreground
- Motion tracking
  - Filtering and prediction
- 3D position estimation
  - Two views
  - mirrored views

Estimating 3D positions

- Given a projected point set, what are the 3D structure?

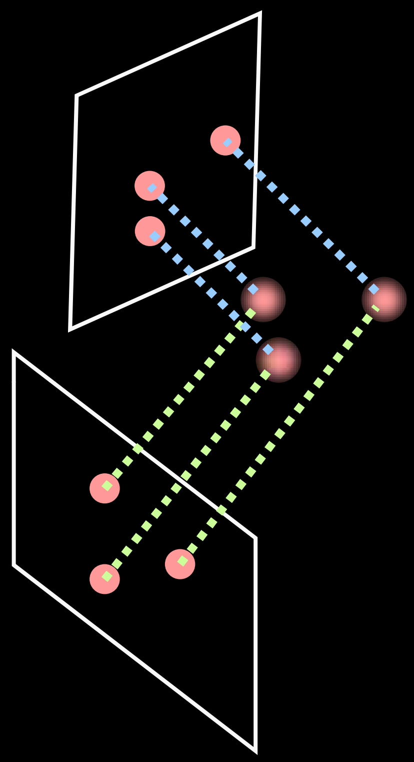


Which one is correct?

- Without other information, all these points are possible!!!
- With prior constraints or at least two views !

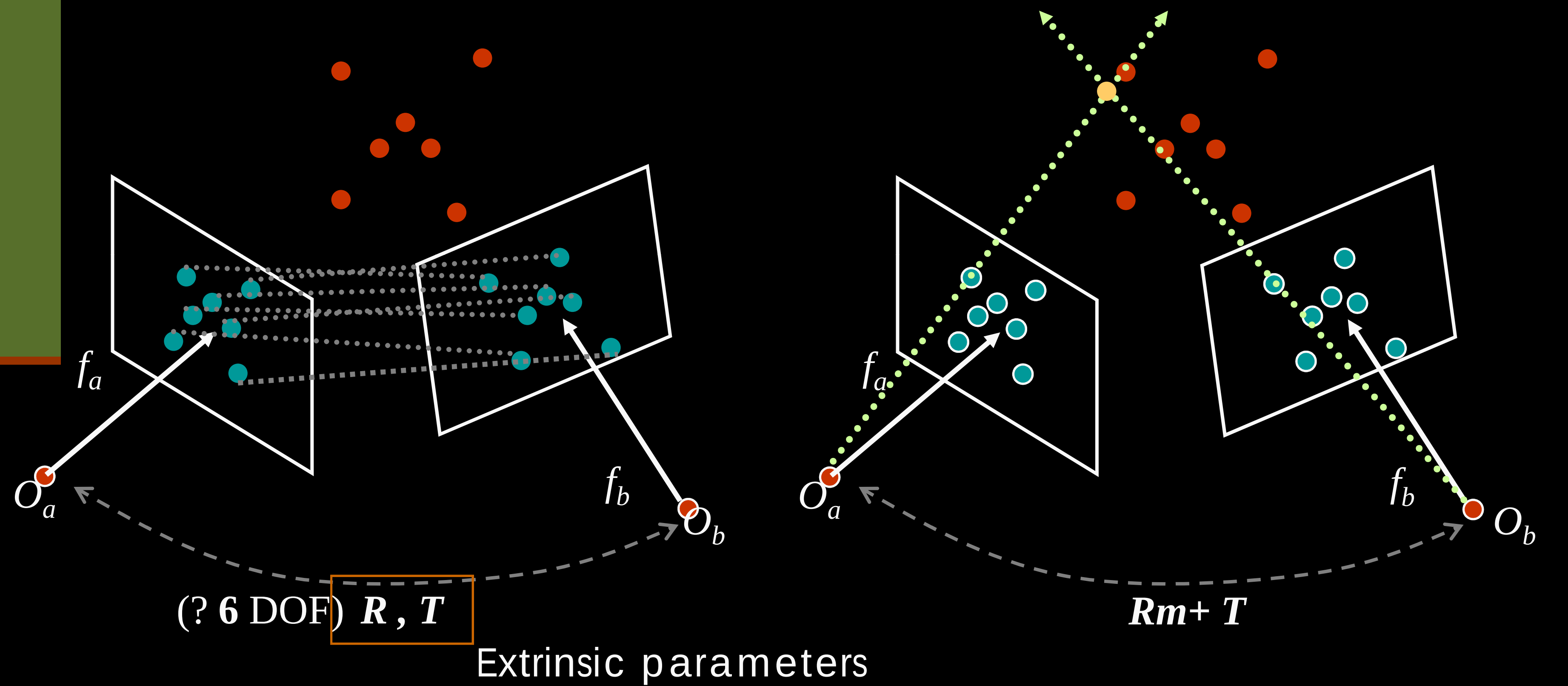
Two Orthogonal Views

- The simplest case:
  - two orthogonal views + parallel projection
- Orthogonal views
  - Calibration problems ?
- Parallel projection
  - When does it work?
- If the applications do not require high accuracy in 3D est., this can be a candidate approach.



Triangulation (Stereo)

- Given some points in correspondence across two or more images (taken from calibrated cameras), $\{(u_j, v_j)\}$, compute the 3D location X



Triangulation (Stereo)

- Constructing 3D structure from two views.
  - H.C. Longuet-Higgins (Nature'81).
  - J.Weng et al. The two-view approach (PAMI'89).

- Given some points in correspondence across two images (in a normalized camera model), $\{(u_j, v_j)\}$,
  - Estimate *R*, *T* from corresponding points.
  - 3D position estimation from triangulation.
  - (optional) non-linear optimization

3D Estimation from Two Views

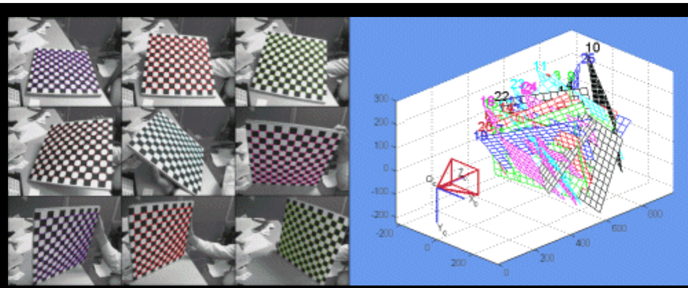
1. Estimate intrinsic camera parameters.
   - E.g. optic center, camera distortion, etc.

2. Estimate extrinsic camera parameters.
   - E.g. motion between two views.

3. Estimate 3D structure by triangulation.

- Refer to textbooks or lecture notes in computer vision (or image-based modeling) courses.

Camera Calibration



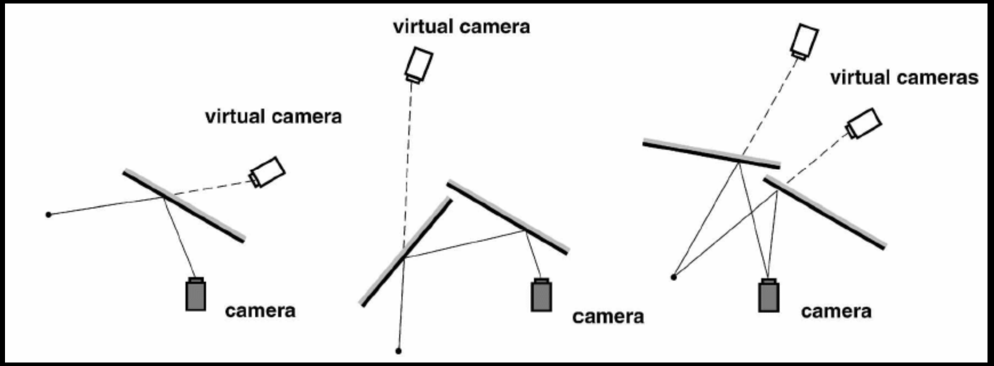
- Public camera calibration tools
  - A flexible new technique for camera calibration
    - http://research.microsoft.com/~zhang/calib/
    - Z. Zhang. A flexible new technique for camera calibration. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.
  - Camera calibration toolbox for matlab
    - http://www.vision.caltech.edu/bouguetj/calib_doc/
  - 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.

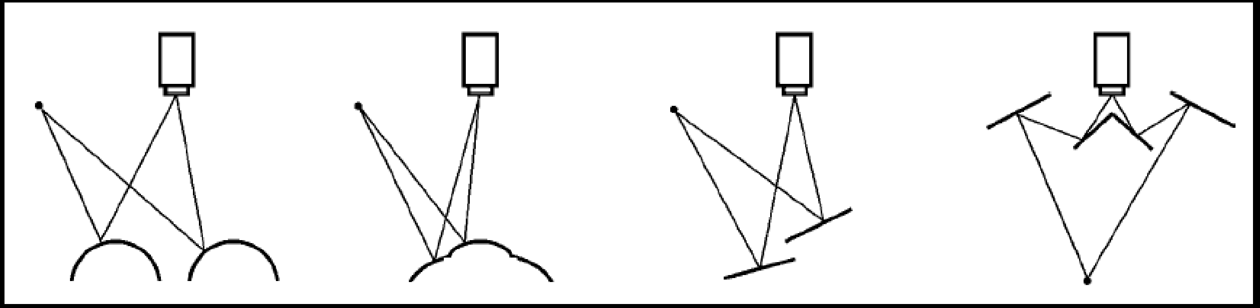
Triangulation limits

- Difficult to reliably estimate structure and motion unless:
  - large (*x* or *y*) rotation
  - large field of view and depth variation
- Camera calibration is important
- Need good feature trackers or manual assistance
- Post-processing of the resulting 3-D points?

Estimating 3D Positions

- Using mirrors, we can acquire multi-views with a single camera.

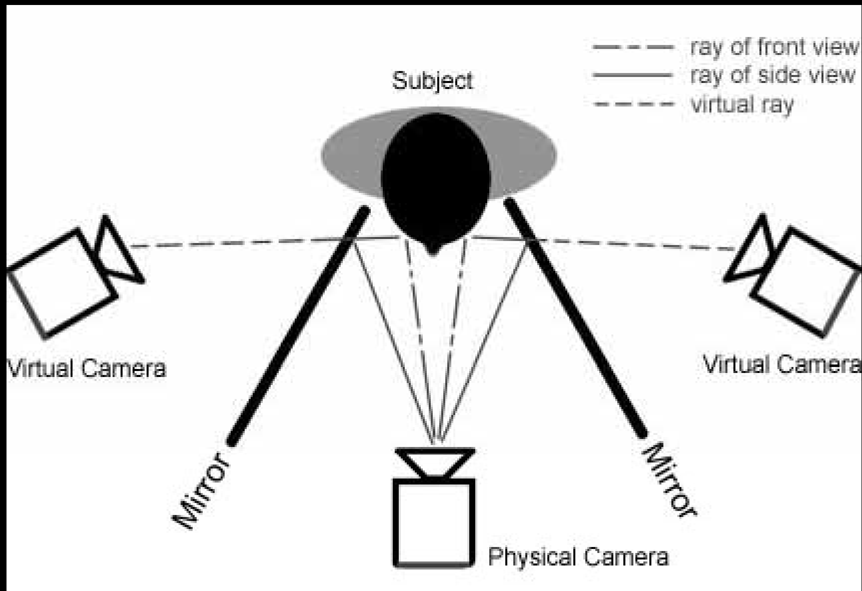




Stereo sensors from a variety of mirrors. J. Gluckman et al. (CVPR’99) (PAMI’02)

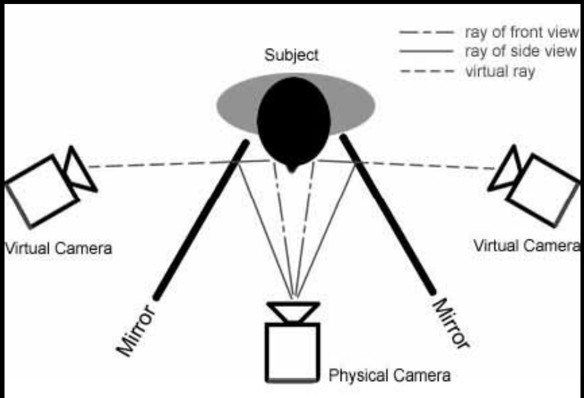
Mirrored Views

- Using planar mirrors can make reconstruction much easier.
- The simplest case:
  - Mirrors placed at 45-degree included angles.
  - Problem:
    - Calibration
    - Projective views
    - ...



Reconstruction from Mirrored Views

- E.C. Patterson et al.(CA'91):
  - Assumed that the mirror and camera was vertical.
- S. Basu et al. (CVPR workshop'97)(ICCV'98):
  - Lip position evaluation via R, t estimation between virtual cameras.



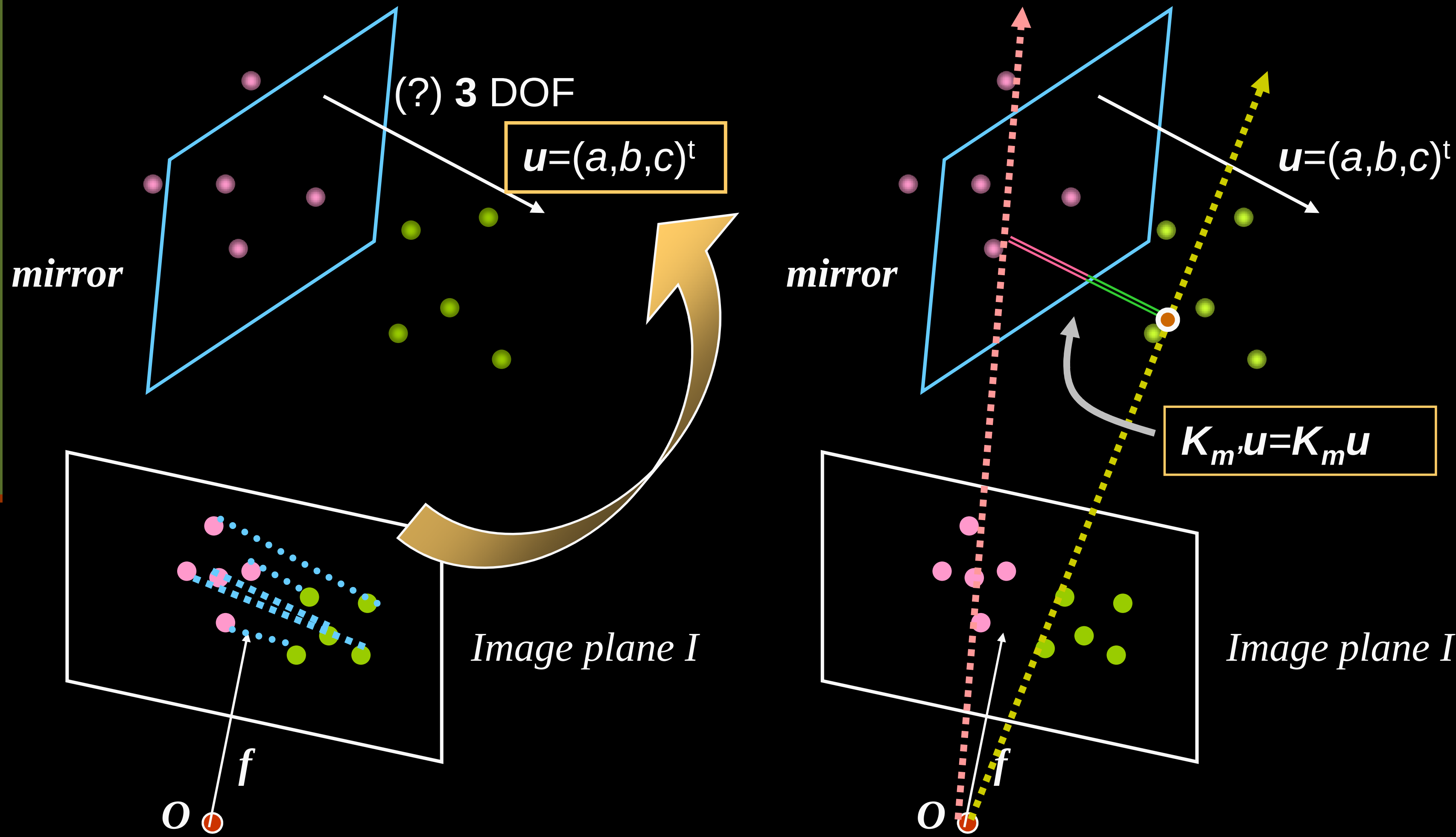
- I.-C Lin et al. (CG&A'02)(TVC' 05)
  - Efficient and reliable algorithms for calibration and 3D reconstruction in planar mirror configuration.
  - For dense facial motion tracking.

3D Tracking in Mirror-reflected Multi-views

“Mirror Mocap” (illuminated by UV light)

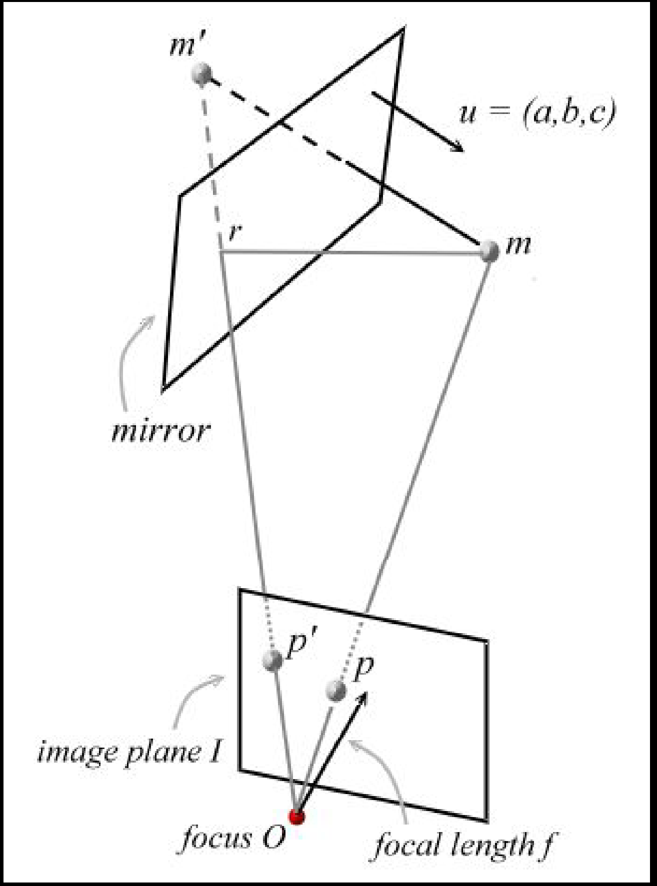
3D Position estimation

- Estimating 3D positions by evaluating the mirror plane's parameters.



3D Position estimation

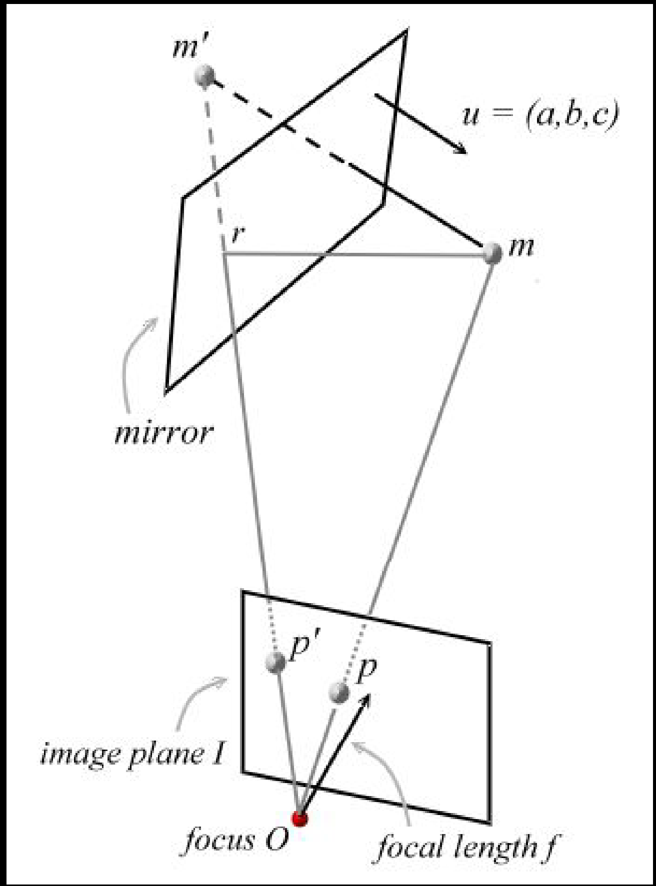
- Given real vs. mirrored projected point correspondences.
- Known: p_i , p_i' , f.
- Unknown: m_i, m_i', u, d.
- Calculating 3D positions via mirror plane estimation.



The geometric representation of physical point *m*, reflected point *m'*, and the projected point *p* and *p'*.

3D Position estimation (cont.)

- We assumed that the mirror is flat.
- Calculating 3D positions via evaluation of the mirror plane.



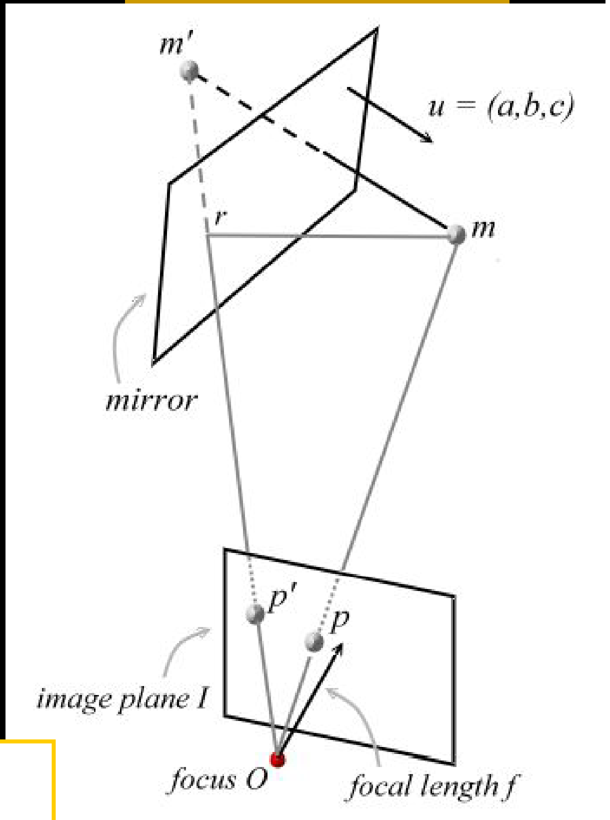
- Properties:
  - $ax+by+cz=d$, $u=(a, b, c)^t$, $|u|=1$. (1)
  - $m_i'=m_i + ku$. (2)
  - $(m_i' - \Theta)=H_u(m_j - \Theta)$, where H_u is the Householder matrix. (3)

3D Position estimation (cont.)

- Since m_i ,m_i' and u are coplanar,

$$(p_i')^t U p_i = 0, \text{ where } U = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}$$

- Approximate the mirror plane by a least square method.



$$\begin{bmatrix} (y_{p1}-y'_{p1})f & (-x_{p1}+x'_{p1})f & (x_{p1}y'_{p1}-y_{p1}x'_{p1}) \\ (y_{p2}-y'_{p2})f & (-x_{p2}+x'_{p2})f & (x_{p2}y'_{p2}-y_{p2}x'_{p2}) \\ \vdots & \vdots & \vdots \\ (y_{pn}-y'_{pn})f & (-x_{pn}+x'_{pn})f & (x_{pn}y'_{pn}-y_{pn}x'_{pn}) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

3D Position estimation (cont.)

- Deducing from symmetric properties, depths are in proportion to *d*. (similar to *T* in stereovision)

$$\begin{bmatrix} \left(\frac{2a^2-1}{2f}\right)x_{pi} + \left(\frac{ab}{f}\right)y_{pi} + ac & \frac{x'_{pi}}{2f} \\ \left(\frac{ab}{f}\right)x_{pi} + \left(\frac{2b^2-1}{2f}\right)y_{pi} + bc & \frac{y'_{pi}}{2f} \\ \left(\frac{ac}{f}\right)x_{pi} + \left(\frac{bc}{f}\right)y_{pi} + \frac{2c^2-1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} z_{mi} \\ z'_{mi} \end{bmatrix} = d \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

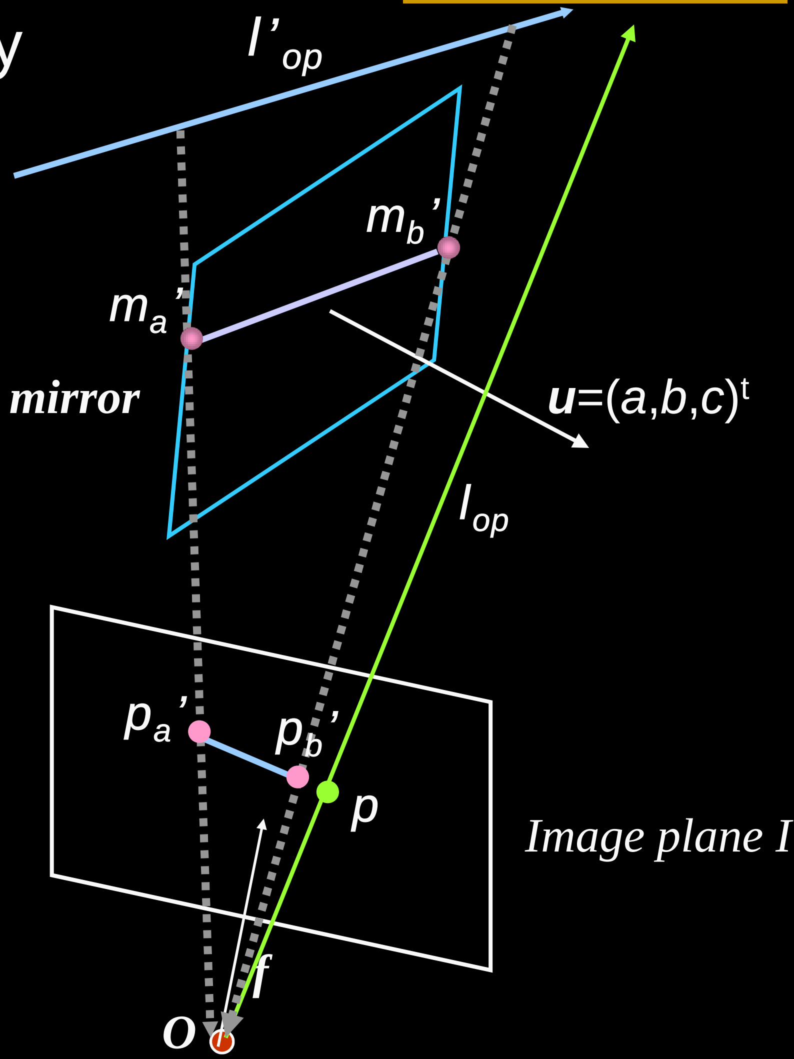
- (z_{mi} and $z_{mi'}$)can be estimated by a least square method. The 3D positions are reconstructed by scaling data.

Potential 3D Candidates

- Constructing 3D candidates by *mirrored epipolar lines*.

$$(p')^t U p = 0$$

$$\begin{bmatrix} x'_p & y'_p & 1 \end{bmatrix} \begin{bmatrix} -cy_p + b \\ cx_p - a \\ -bx_p + ay_p \end{bmatrix} = 0$$



Conclusion

- Efficient (or simple) 3D reconstruction algorithms are introduced.
- 3D vision-based controls can provide more flexibility.
- The process time and device requirement will increase for 3D.
- Choose an appropriate method according to accuracy, budgets, process time, devices, etc.

Appendix

- H.C. Longuet-Higgins (Nature'81).
- J.Weng et al. The two-view approach (PAMI'89).

The Two-view Approach

- Without loss of generality, the images of different view direction *d1*, *d2* is regarded as a rigid-body motion of an object between *t1*, *t2*.

$\boldsymbol{x_i} = (x_i, y_i, z_i)$t is the 3D position of point P_i at time t_1.

$\boldsymbol{x_i}$'$=(x_i', y_i', z_i')$t is the 3D position of point P_i at time t_2.

$\boldsymbol{X_i}=(u_i, v_i, 1)$t is the projected vector of P_i at time t_1.

$\boldsymbol{X_i}$'$=(u_i', v_i', 1)$t $= (x_i'/z_i', y_i'/z_i', 1)$t is the projected vector of P_i at time t_2.

The Two-view Approach (1)

- Step (1). Solving for essential matrix E.

$$A=\begin{bmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu'_n & u_nv'_n & u_n & v_nu'_n & v_nv'_n & v_n & u'_2 & v'_n & 1 \end{bmatrix}$$

- $\min_h$ || Ah || = 1, subject to || h || = 1.

$$E=\begin{bmatrix} E_1 & E_2 & E_3 \end{bmatrix}=\sqrt{2}\begin{bmatrix} h_1 & h_4 & h_7 \\ h_2 & h_5 & h_8 \\ h_3 & h_6 & h_9 \end{bmatrix}$$

The solution of h is the unit eigenvector of A^tA associated with the smallest eigenvalue.

The Two-view Approach (2)

- Step (2). Determining a unit vector T_s with $T_0 = \pm T_s$.
  - $\min_{T_s} \| E^t T_s \|$, subject to $\| T_s \| = 1$.

The solution of T_s is the unit eigenvector of EE^t associated with the smallest eigenvalue.

$$E = T_\times R = \begin{bmatrix} E_1 & E_2 & E_3 \end{bmatrix} = \begin{bmatrix} T_\times R_1 & T_\times R_2 & T_\times R_3 \end{bmatrix}$$

$$\therefore E_1, E_2, E_3 \perp T \quad \Rightarrow \quad E^t T_s = 0$$

  - if($\Sigma_i (T_s \times X_i') \bullet (E\, X_i) < 0$), $T_s = -T_s$.

The Two-view Approach (3)

- Step (3). Determining rotation matrix *R*.
  - Without noise, $W=R$

    $$W = [(E_1 \times T_s + E_2 \times E_3) \quad (E_2 \times T_s + E_3 \times E_1) \quad (E_3 \times T_s + E_1 \times E_2)]$$

    Using the identity equation: (a x b) x c = (a · c) b – (b · c) a

  - With noise,

    $\min_R \| R - W \|$, subject to: *R* is a rotation matrix.

The Two-view Approach (3 app.)

- $\min_R \| RC - D \|$, subject to: R is a rotation matrix.

$$B = \sum_{i=1}^{3} B_i^t B_i$$

- Define a 4x4 matrix B by

$$B_i = \begin{bmatrix} 0 & (C_i - D_i)^t \\ D_i - C_i & [D_i + C_i]_\times \end{bmatrix}_{4\times 4}$$

  where

- **q** = $(q_0, q_1, q_2, q_3)^t$ is the unit eigenvector of B associated with the smallest eigenvalue.

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_2q_1 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_3q_1 - q_0q_2) & 2(q_3q_2 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

The Two-view Approach (4)

- Step (4). Checking $T = 0$, If $T \neq 0$, determine the sign of T_0.

  if for all $i = 1 \sim n$, then report $T \approx 0$.

  else if ($\Sigma_i(T_s \times X_i') \bullet (R\, X_i) > 0$), then $T_0 = T_s$, otherwise $T_0 = -T_s$.

The Two-view Approach (5)

- Step (5). If $T \neq 0$, estimate relative depths.
- To find $Z_i = \left(\frac{z_i'}{\|T\|}, \frac{z_i}{\|T\|} \right)^t = \left(\tilde{z}_i', \tilde{z}_i \right)$

by $\min \left\| \left[X_i' \quad -RX_i \right] Z_i - T^0 \right\|$

The Nonlinear Optimization

- Two-view linear algorithms are often easily disturbed by noise.
  - More calibration points.
  - Nonlinear optimization.
- First, take the result of the two-view linear algorithm as an initial guess.
- Approximate the *R*, *T* by $\min_m \{|| f(u,m) ||\}$ in a nonlinear least square approach
  - E.g. the Levenberg-Marquardt method, or the Gauss-Newton method.
  - $f(u,m) = prj(m, y(u,m)) - u$

where *u* is the observed projected position,
m is the motion parameters(*R*,*T*), *y*(*u*,*m*) is the best 3D positions of *P*,
and *prj*(*m*, *x*) is the projected position of the input structure x and motion m.