Image/Video Based 3D Modeling, Rendering, and Registration for Virtual Reality

Hideo Saito

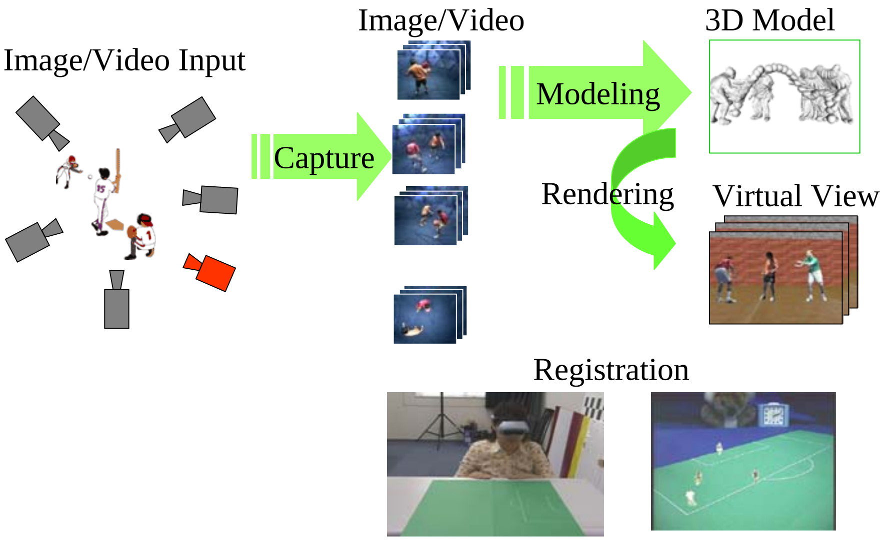
saito@ozawa.ics.keio.ac.jp
www.ozawa.ics.keio.ac.jp/Saito

Dept. Information and Computer Science,
Keio University, Japan

Contents of this tutorial

- Introduction
- Capturing Image
    - Camera Geometry
    - Calibration
- Modeling and Rendering from Multiple Images
    - Modeling and Rendering from Image Sequences
    - Modeling and Rendering from Multiple Cameras
- Registration
    - Application of Registration to Augmented Reality

Image/Video Based ...



Image/Video Input

Capture

Image/Video

Modeling

3D Model

Rendering

Virtual View

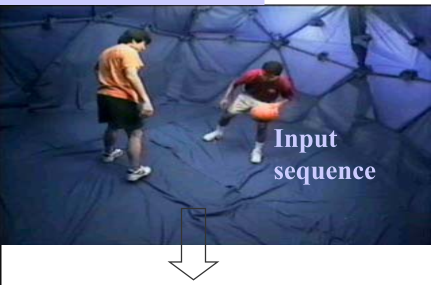
Registration

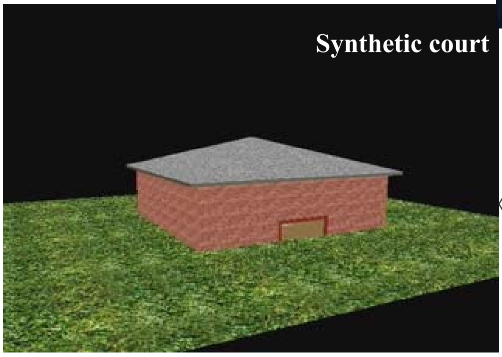
Modeling and Rendering

Example:

3-Man Basketball (CMU, 1998)

(These are movies.)





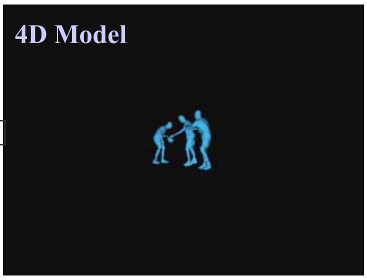
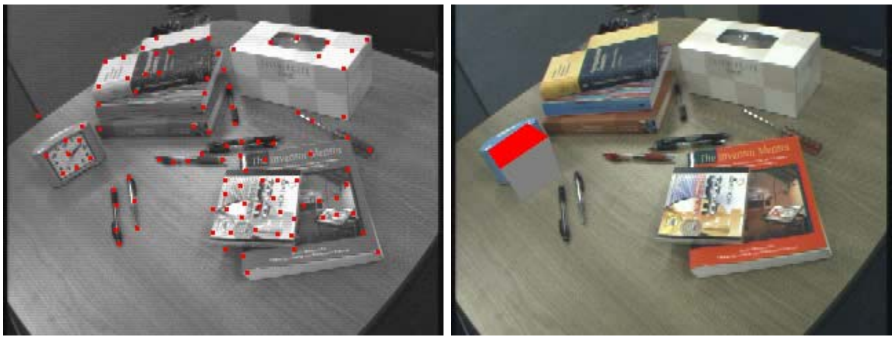


Image Based Registration Example



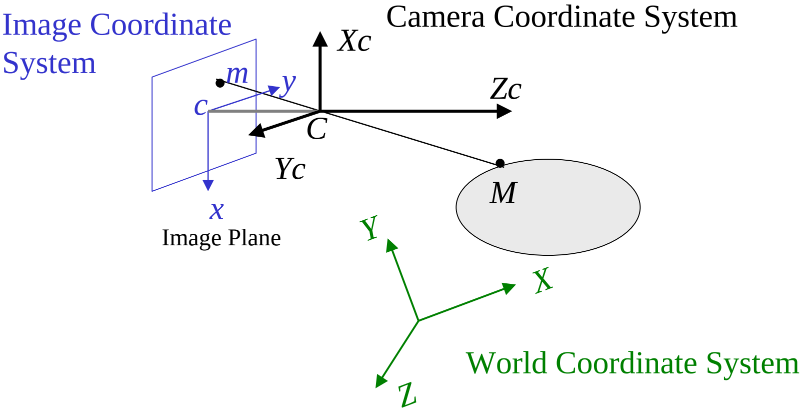
Base Technologies

- Capturing Image
  - Camera Geometry, Calibration
- Modeling and Rendering
  - Shape Recovery
  - Volume Reconstruction
  - Surface Extraction
- Registration
  - Camera Calibration
  - Tracking

Capturing Image

Camera Geometry

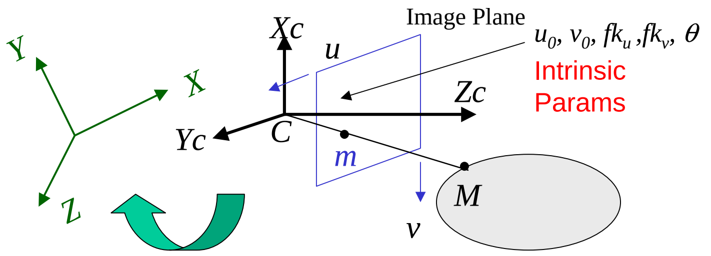
- Image : Projection of 3D World onto 2D Plane
  - How the projected position is determined ?
  - Relationship between world coordinate, camera coordinate, and image coordinate



Summary of Projection Matrix

- Projection **P:** from World Coord. **M** to Image Coord. **m**

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}} \qquad s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



u_0, v_0, fk_u ,fk_v, θ

Intrinsic Params

(**R**, **t**):Extrinsic Params(6DOFs)

Camera Calibration

- Intrinsic Parameters（5DOFs）
- Extrinsic Parameters（6DOFs）

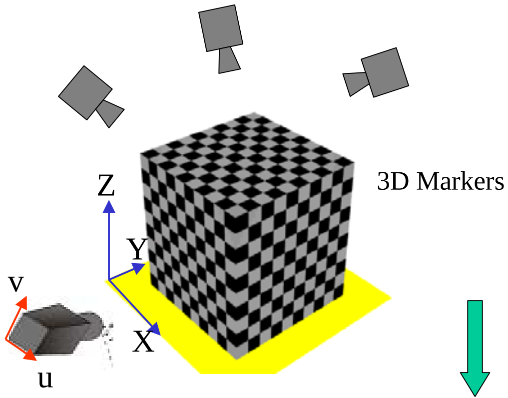
Basic Method for Camera Calibration

Distribute markers with known 3D positions (X,Y,Z) in objective space

Find image positions (u,v) onto which the markers are projected

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}} \qquad s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera Calibration



Input Data

X1,Y1,Z1 → u1,v1

X2,Y2,Z2 → u2,v2

X3,Y3,Z3 → u3,v3

X4,Y4,Z4 → u5,v5

.

.

Projection Matrix ***P***

Linear Solution for Estimating **P**

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}} \qquad s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

(X_n, Y_n, Z_n) --- (u_n, v_n)

$$\Downarrow$$

$$P_{11}X_n + P_{12}Y_n + P_{13}Z_n + P_{14} - P_{31}X_n u_n - P_{32}Y_n u_n - P_{33}Z_n u_n - P_{34}u_n = 0$$
$$P_{21}X_n + P_{22}Y_n + P_{23}Z_n + P_{24} - P_{31}X_n v_n - P_{32}Y_n v_n - P_{33}Z_n v_n - P_{34}v_n = 0$$

$$2n\left\{\overbrace{\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1u_1 & -Y_1u_1 & Z_1u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1v_1 & -Y_1v_1 & Z_1v_1 \\ & & & & & & & \cdots & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -X_nu_n & -Y_nu_n & Z_nu_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -X_nv_n & -Y_nv_n & Z_nv_n \end{bmatrix}}^{11}\right. \underbrace{\begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{32} \\ P_{33} \end{bmatrix}}_{11} = \underbrace{\begin{bmatrix} P_{34}u_1 \\ P_{34}v_1 \\ \vdots \\ P_{34}u_n \\ P_{34}v_n \end{bmatrix}}_{11}$$

n>11/2 markers are required for estimating **P**

Method for extracting intrinsic and extrinsic parameters from projection matrix **P**

$$\mathbf{P}=\mathbf{A}[\mathbf{R},\mathbf{t}]=\begin{bmatrix}\alpha_u & -\alpha_u\cot\theta & u_0\\ 0 & \alpha_v/\sin\theta & v_0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}R_{11} & R_{21} & R_{31} & t_X\\ R_{12} & R_{22} & R_{32} & t_Y\\ R_{13} & R_{23} & R_{33} & t_Z\end{bmatrix}$$

Projection Matrix **P**→Intrinsic Matrix **A**
Extrinsic Matrix and Vector **R, t**

$$\mathbf{P}_w=\begin{bmatrix}P_{11} & P_{12} & P_{13} & P_{14}\\ P_{21} & P_{22} & P_{23} & P_{24}\\ P_{31} & P_{32} & P_{33} & P_{34}\end{bmatrix}=\begin{bmatrix}\mathbf{a}_1 & \\ \mathbf{a}_2 & \mathbf{b}\\ \mathbf{a}_3 & \end{bmatrix}$$

$$[\mathbf{R},\mathbf{t}]=\begin{bmatrix}R_{11} & R_{21} & R_{31} & t_X\\ R_{12} & R_{22} & R_{32} & t_Y\\ R_{13} & R_{23} & R_{33} & t_Z\end{bmatrix}=\begin{bmatrix}\mathbf{r}_1 & \\ \mathbf{r}_2 & \mathbf{t}\\ \mathbf{r}_3 & \end{bmatrix}$$

Property of Rotation Matrix

$$|\mathbf{r}_1|=|\mathbf{r}_2|=|\mathbf{r}_3|=1 \qquad \mathbf{r}_1\bullet\mathbf{r}_2=0 \qquad \mathbf{r}_2\bullet\mathbf{r}_3=0 \qquad \mathbf{r}_3\bullet\mathbf{r}_1=0$$

$$\mathbf{r}_1\times\mathbf{r}_2=\mathbf{r}_3 \qquad \mathbf{r}_2\times\mathbf{r}_3=\mathbf{r}_1 \qquad \mathbf{r}_3\times\mathbf{r}_1=\mathbf{r}_2$$

The property of rotation matrix provide the following solution.

$$\mathbf{r}_3=\mp\frac{\mathbf{a}_3}{|\mathbf{a}_3|} \qquad \mathbf{r}_1=\frac{1}{|\mathbf{a}_2\times\mathbf{a}_3|}(\mathbf{a}_2\times\mathbf{a}_3) \qquad \mathbf{r}_2=\mathbf{r}_3\times\mathbf{r}_1$$

Intrinsic parameter matrix **A** can be obtained as

$$\cos\theta=-\frac{(\mathbf{a}_1\times\mathbf{a}_3)\bullet(\mathbf{a}_2\times\mathbf{a}_3)}{|\mathbf{a}_1\times\mathbf{a}_3||\mathbf{a}_2\times\mathbf{a}_3|}$$

$$\alpha_u=\frac{|\mathbf{a}_1\times\mathbf{a}_3|}{|\mathbf{a}_3|^2}\sin\theta,\ \ \alpha_v=\frac{|\mathbf{a}_2\times\mathbf{a}_3|}{|\mathbf{a}_3|^2}\sin\theta$$

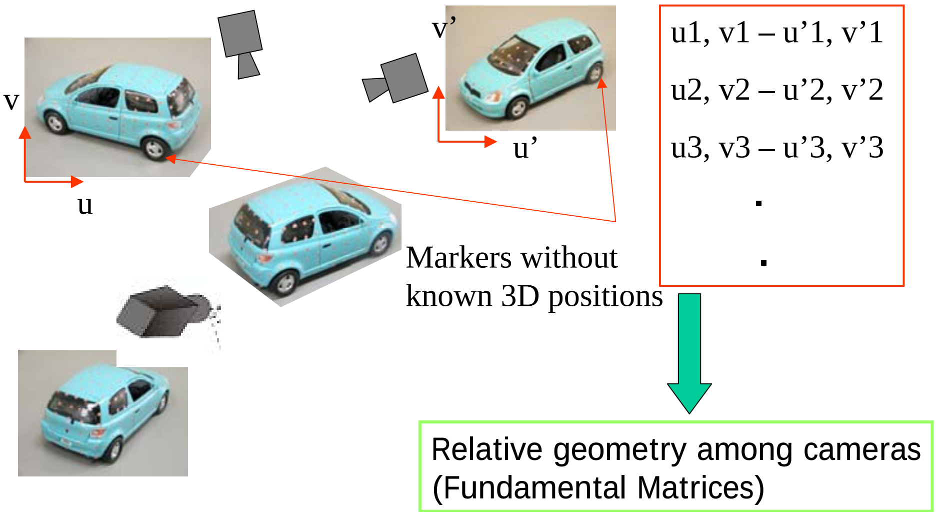
Self Calibration

- Camera calibration without markers

  C. Matsunaga and K. Kanatani, “Calibration of a moving camera using a planar pattern: Optimal computation, reliability evaluation and stabilization by the geometric AIC,” Electronics and Communications in Japan, Part 3:Fundamental Electronic Science,Vol. 84, No. 7 pp. 12-21,2001.

  M. Pollefeys, R. Koch, and L. V. Gool, “Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters,” International Journal of Computer Vision, 32(1), pp. 7-25, 1999.

Weak Calibration



u1, v1 – u’1, v’1
u2, v2 – u’2, v’2
u3, v3 – u’3, v’3
.
.

Markers without known 3D positions

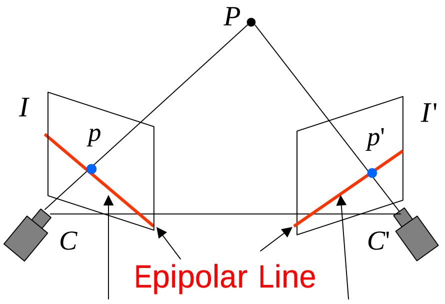
Relative geometry among cameras (Fundamental Matrices)

Fundamental Matrix

$$\begin{pmatrix} a\\ b\\ c\end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13}\\ f_{21} & f_{22} & f_{23}\\ f_{31} & f_{32} & f_{33}\end{pmatrix}\begin{pmatrix} x'\\ y'\\ 1\end{pmatrix}$$

$$= \mathbf{F}\begin{pmatrix} x'\\ y'\\ 1\end{pmatrix}$$

$$\begin{pmatrix} a'\\ b'\\ c'\end{pmatrix} = \mathbf{F}^T\begin{pmatrix} x\\ y\\ 1\end{pmatrix}$$



$ax+by+c=0$ $\quad a'x'+b'y'+c'=0$

$$\begin{pmatrix} a\\ b\\ c\end{pmatrix} = \mathbf{F}\begin{pmatrix} x'\\ y'\\ 1\end{pmatrix}, \quad \begin{pmatrix} a'\\ b'\\ c'\end{pmatrix} = \mathbf{F}^T\begin{pmatrix} x\\ y\\ 1\end{pmatrix}, \quad (x, y, 1)\mathbf{F}\begin{pmatrix} x'\\ y'\\ 1\end{pmatrix} = 0$$

One corresponding point gives the following equation.

$$x(x'F_{11} + y'F_{12} + F_{13}) + y(x'F_{21} + y'F_{22} + F_{23}) + (x'F_{31} + y'F_{32} + F_{33}) = 0$$

n corresponding points

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1'\\ x_2x_2' & x_2y_2 & x_2 & y_2x_2' & y_2y_2' & y_2 & x_2' & y_2'\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ x_nx_n' & x_ny_n & x_n & y_nx_n' & y_ny_n' & y_n & x_n' & y_n' \end{bmatrix} \begin{bmatrix} F_{11}\\ F_{12}\\ F_{13}\\ F_{21}\\ F_{22}\\ F_{23}\\ F_{31}\\ F_{32}\end{bmatrix} = -\begin{bmatrix} 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\end{bmatrix}$$

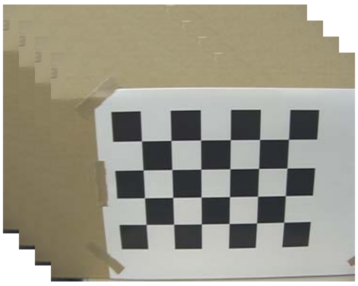
$$F_{33} = 1$$

Strong/Weak Calibration

	Strong Calibration	Weak Calibration
To be estimated	Projection Matrix $s\begin{bmatrix}u\\v\\1\end{bmatrix}=\mathbf{P}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$	Fundamental Matrix $\begin{bmatrix}a_1\\b_1\\1\end{bmatrix}=\mathbf{F}\begin{bmatrix}x_2\\y_2\\1\end{bmatrix}$
Required data	3D Space ― 2D Image Correspondences (Artificial Markers required)	2D Image― 2D Image Correspondences (Natural features)
Size Shape	Available	Unknown (Ambiguity in Projective Transform)

Plane-Based Calibration

A Flexible New Technique for Camera Calibration
http://research.microsoft.com/~zhang/calib/



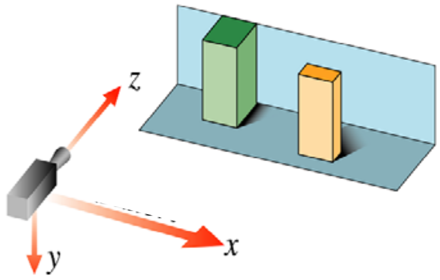
Multiple Images of
Checker Pattern Board
at Arbitrary Pose

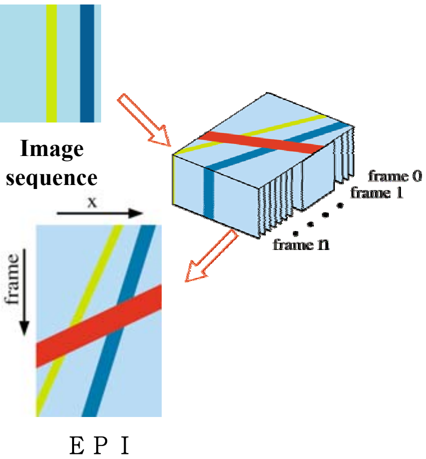
Modeling and Rendering

Modeling and Rendering

- Image Sequence captured by a camera
  - Motion Stereo
  - Feature point tracking
  - Reflectance analysis
- Multiple Cameras
  - Silhouette Intersection
  - Merge Stereo
  - Space Carving/Voxel Coloring

Motion Stereo

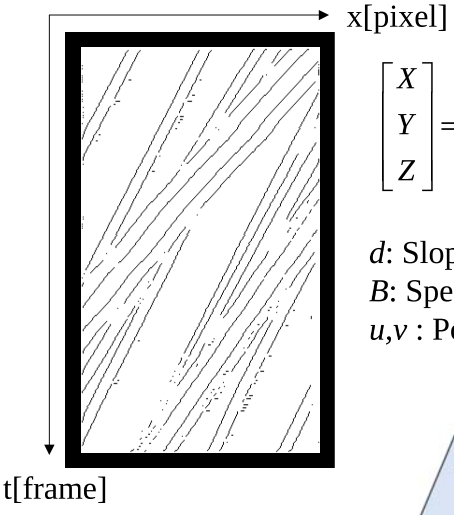




Epipolar Plane Image

Camera moves toward *x* direction

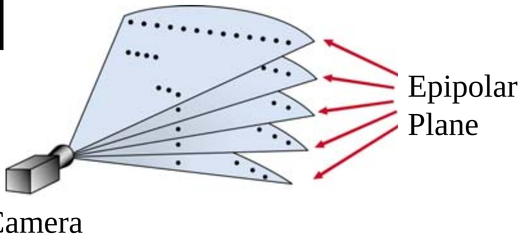
➡ Epipolar lines are paralell with *x* axis



Line Detection in EPI

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} uB/d \\ vB/d \\ fB/d \end{bmatrix}$$

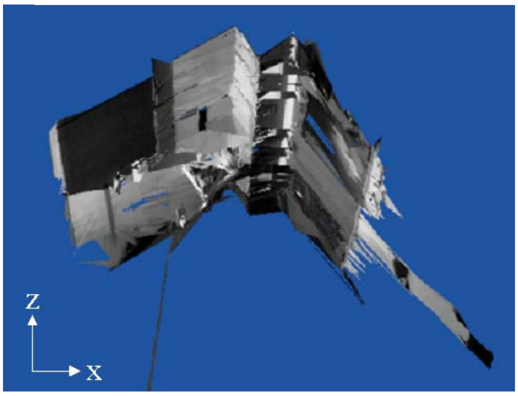
d: Slope of line [pixel/frame]
B: Speed of camera [m/frame]
u,*v* : Position of line [pixel]



120th frame

240th frame

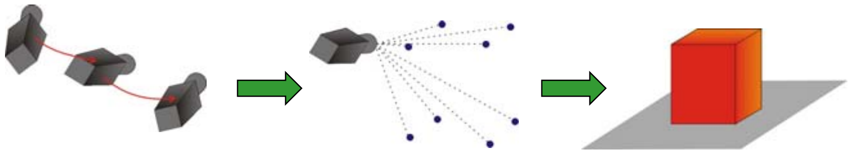
360th frame



Recovered Shape with Texture of Input Images

Feature point tracking

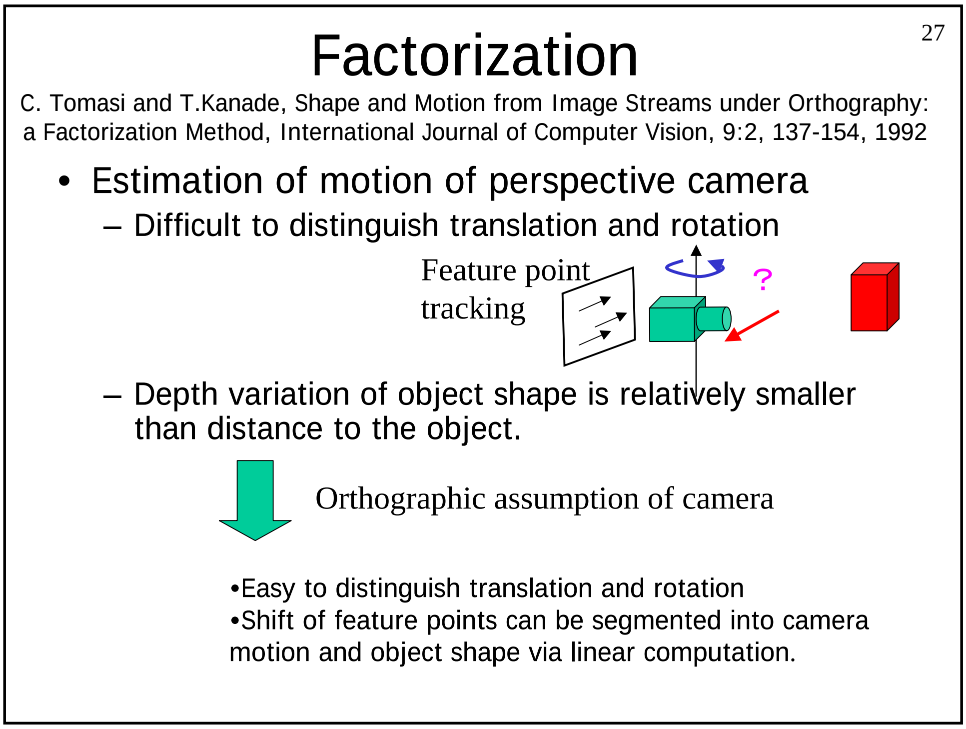
- Feature points are tracked.
- Positions of tracked feature points are used for shape modeling.
- Standard feature tracking algorithm
  - **Kanade-Lucas-Tomasi Feature Tracker (http://robotics.stanford.edu/~birch/klt//)**



Moving Camera

Feature Point Tracking

Shape



Factorization

C. Tomasi and T.Kanade, Shape and Motion from Image Streams under Orthography: a Factorization Method, International Journal of Computer Vision, 9:2, 137-154, 1992

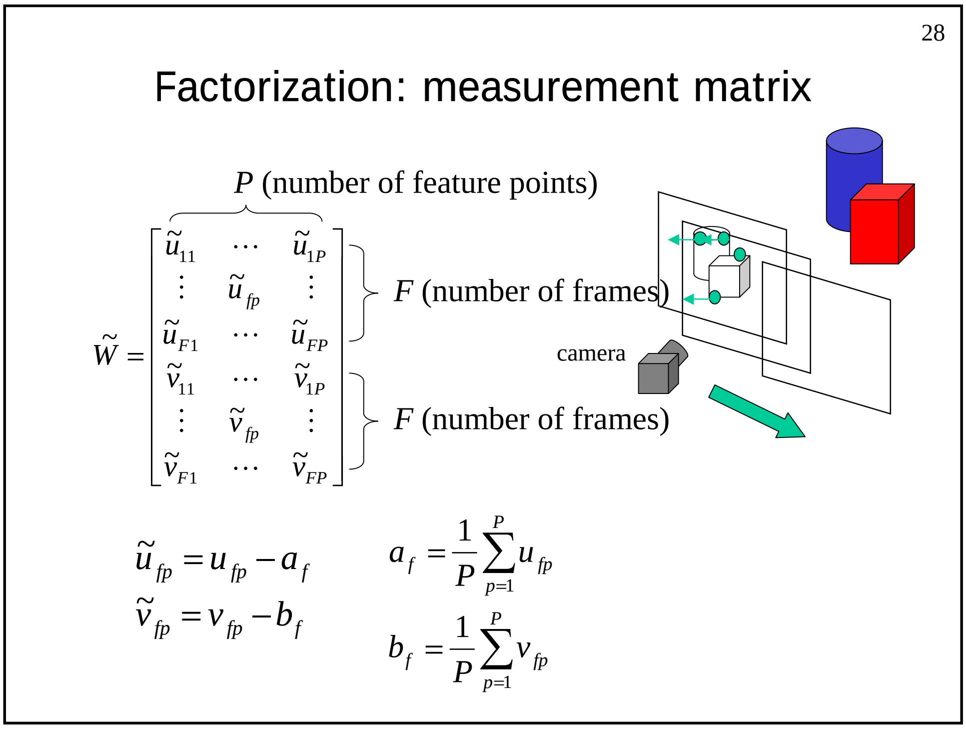
- Estimation of motion of perspective camera
  - Difficult to distinguish translation and rotation

Feature point
tracking

  - Depth variation of object shape is relatively smaller than distance to the object.

Orthographic assumption of camera

•Easy to distinguish translation and rotation
•Shift of feature points can be segmented into camera motion and object shape via linear computation.



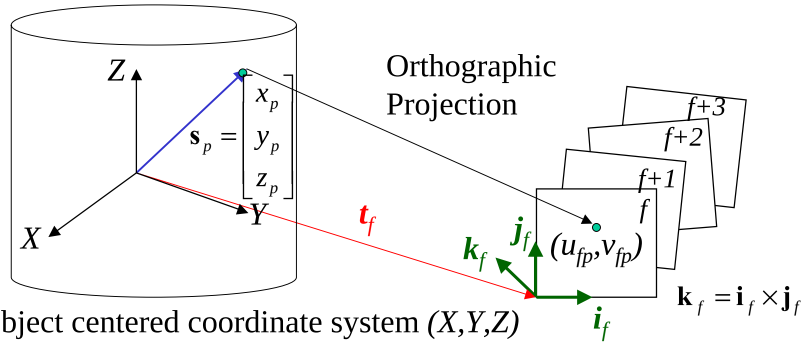
Factorization: measurement matrix

$$\tilde{W} = \begin{array}{c} \overbrace{\qquad\qquad\qquad\qquad}^{P \text{ (number of feature points)}} \\ \left[\begin{array}{ccc} \tilde{u}_{11} & \cdots & \tilde{u}_{1P} \\ \vdots & \tilde{u}_{fp} & \vdots \\ \tilde{u}_{F1} & \cdots & \tilde{u}_{FP} \\ \tilde{v}_{11} & \cdots & \tilde{v}_{1P} \\ \vdots & \tilde{v}_{fp} & \vdots \\ \tilde{v}_{F1} & \cdots & \tilde{v}_{FP} \end{array}\right] \end{array} \begin{array}{l} \\ \left.\vphantom{\begin{array}{c} a \\ a \\ a \end{array}}\right\} F \text{ (number of frames)} \\ \left.\vphantom{\begin{array}{c} a \\ a \\ a \end{array}}\right\} F \text{ (number of frames)} \end{array}$$

$$\tilde{u}_{fp} = u_{fp} - a_f \qquad a_f = \frac{1}{P}\sum_{p=1}^{P} u_{fp}$$

$$\tilde{v}_{fp} = v_{fp} - b_f \qquad b_f = \frac{1}{P}\sum_{p=1}^{P} v_{fp}$$

Shape and Motion under Orthographic Projection



Object centered coordinate system *(X,Y,Z)*

$$\tilde{u}_{fp} = u_{fp} - a_f$$

$$u_{fp} = \mathbf{i}_f^T(\mathbf{s}_p - \mathbf{t}_f)$$

$$v_{fp} = \mathbf{j}_f^T(\mathbf{s}_p - \mathbf{t}_f)$$

$$= \mathbf{i}_f^T(\mathbf{s}_p - \mathbf{t}_f) - \frac{1}{P}\sum_{q=1}^{P} \mathbf{i}_f^T(\mathbf{s}_q - \mathbf{t}_f)$$

$$= \mathbf{i}_f^T\left(\mathbf{s}_p - \frac{1}{P}\sum_{q=1}^{P} \mathbf{s}_q\right) = \mathbf{i}_f^T \mathbf{s}_p$$

$$\tilde{u}_{fp} = \mathbf{i}_f^T \mathbf{s}_p$$

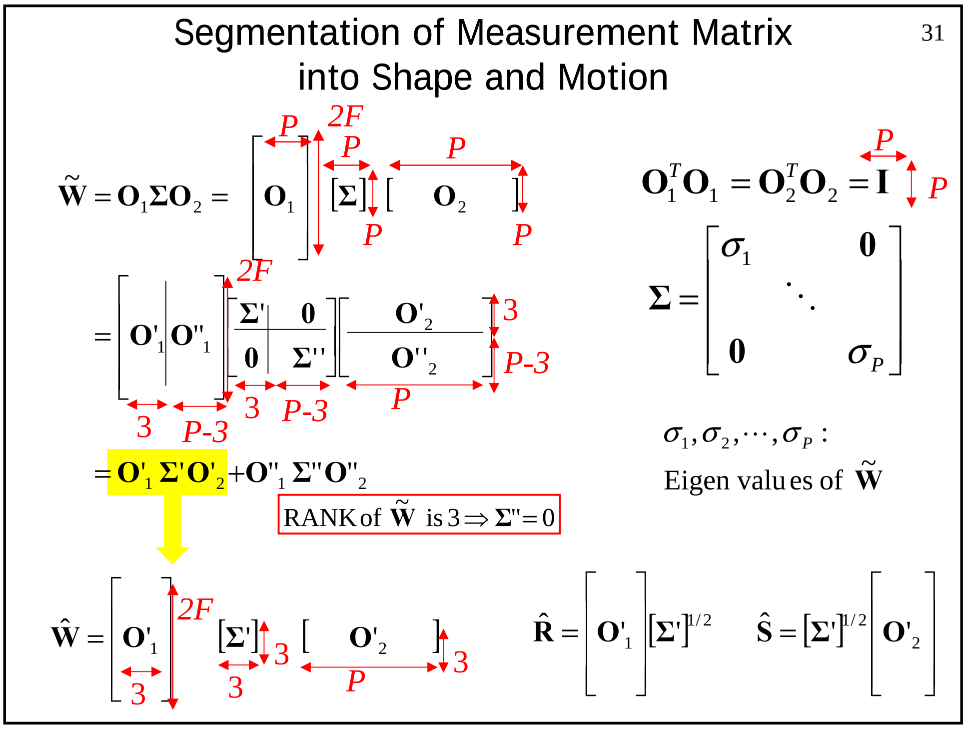
$$\tilde{v}_{fp} = \mathbf{j}_f^T \mathbf{s}_p$$

$$\tilde{\mathbf{W}} = \overbrace{\begin{bmatrix} \tilde{u}_{11} & \cdots & \tilde{u}_{1P} \\ \vdots & \tilde{u}_{fp} & \vdots \\ \tilde{u}_{F1} & \cdots & \tilde{u}_{FP} \\ \tilde{v}_{11} & \cdots & \tilde{v}_{1P} \\ \vdots & \tilde{v}_{fp} & \vdots \\ \tilde{v}_{F1} & \cdots & \tilde{v}_{FP} \end{bmatrix}}^{P\text{(number of feature points)}} = \begin{bmatrix} \mathbf{i}_1^T\mathbf{s}_1 & \cdots & \mathbf{i}_1^T\mathbf{s}_P \\ \vdots & \mathbf{i}_f^T\mathbf{s}_p & \vdots \\ \mathbf{i}_F^T\mathbf{s}_1 & \cdots & \mathbf{i}_F^T\mathbf{s}_P \\ \mathbf{j}_1^T\mathbf{s}_1 & \cdots & \mathbf{j}_1^T\mathbf{s}_P \\ \vdots & \mathbf{j}_f^T\mathbf{s}_p & \vdots \\ \mathbf{j}_F^T\mathbf{s}_1 & \cdots & \mathbf{j}_F^T\mathbf{s}_P \end{bmatrix} = \left.\overset{3}{\begin{bmatrix} \mathbf{i}_1^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_1^T \\ \vdots \\ \mathbf{j}_F^T \end{bmatrix}} 2F \; \overset{P}{\begin{bmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_P \end{bmatrix}} 3 = \mathbf{RS} \right\} 2F$$

Camera Rotation Matrix

Shape Matrix

$$\boxed{\text{RANK of } \tilde{\mathbf{W}} \text{ is } 3}$$



Segmentation of Measurement Matrix into Shape and Motion

$$\tilde{\mathbf{W}} = \mathbf{O}_1 \boldsymbol{\Sigma} \mathbf{O}_2 = \begin{bmatrix} \mathbf{O}_1 \end{bmatrix} [\boldsymbol{\Sigma}] [\ \mathbf{O}_2\ ]$$

$$= \begin{bmatrix} \mathbf{O}'_1 & \mathbf{O}''_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}' & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}'' \end{bmatrix} \begin{bmatrix} \mathbf{O}'_2 \\ \mathbf{O}''_2 \end{bmatrix}$$

$$= \mathbf{O}'_1 \boldsymbol{\Sigma}' \mathbf{O}'_2 + \mathbf{O}''_1 \boldsymbol{\Sigma}'' \mathbf{O}''_2$$

$\boxed{\text{RANK of } \tilde{\mathbf{W}} \text{ is } 3 \Rightarrow \boldsymbol{\Sigma}'' = 0}$

$$\hat{\mathbf{W}} = \begin{bmatrix} \mathbf{O}'_1 \end{bmatrix} [\boldsymbol{\Sigma}'] [\ \mathbf{O}'_2\ ]$$

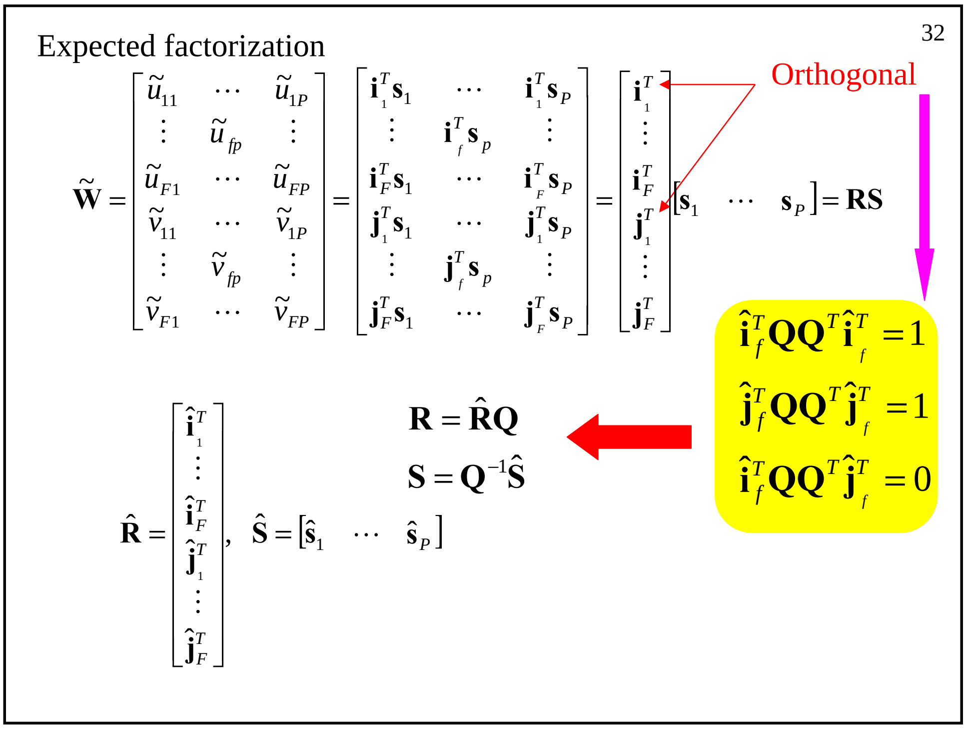
$$\mathbf{O}_1^T \mathbf{O}_1 = \mathbf{O}_2^T \mathbf{O}_2 = \mathbf{I}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_P \end{bmatrix}$$

$\sigma_1, \sigma_2, \cdots, \sigma_P$:

Eigen values of $\tilde{\mathbf{W}}$

$$\hat{\mathbf{R}} = \begin{bmatrix} \mathbf{O}'_1 \end{bmatrix} [\boldsymbol{\Sigma}']^{1/2} \qquad \hat{\mathbf{S}} = [\boldsymbol{\Sigma}']^{1/2} \begin{bmatrix} \mathbf{O}'_2 \end{bmatrix}$$



Expected factorization

$$\tilde{\mathbf{W}} = \begin{bmatrix} \tilde{u}_{11} & \cdots & \tilde{u}_{1P} \\ \vdots & \tilde{u}_{fp} & \vdots \\ \tilde{u}_{F1} & \cdots & \tilde{u}_{FP} \\ \tilde{v}_{11} & \cdots & \tilde{v}_{1P} \\ \vdots & \tilde{v}_{fp} & \vdots \\ \tilde{v}_{F1} & \cdots & \tilde{v}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1^T \mathbf{s}_1 & \cdots & \mathbf{i}_1^T \mathbf{s}_P \\ \vdots & \mathbf{i}_f^T \mathbf{s}_p & \vdots \\ \mathbf{i}_F^T \mathbf{s}_1 & \cdots & \mathbf{i}_F^T \mathbf{s}_P \\ \mathbf{j}_1^T \mathbf{s}_1 & \cdots & \mathbf{j}_1^T \mathbf{s}_P \\ \vdots & \mathbf{j}_f^T \mathbf{s}_p & \vdots \\ \mathbf{j}_F^T \mathbf{s}_1 & \cdots & \mathbf{j}_F^T \mathbf{s}_P \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_1^T \\ \vdots \\ \mathbf{j}_F^T \end{bmatrix} [\mathbf{s}_1 \ \cdots \ \mathbf{s}_P] = \mathbf{RS}$$

Orthogonal

$$\hat{\mathbf{i}}_f^T \mathbf{Q} \mathbf{Q}^T \hat{\mathbf{i}}_f^T = 1$$
$$\hat{\mathbf{j}}_f^T \mathbf{Q} \mathbf{Q}^T \hat{\mathbf{j}}_f^T = 1$$
$$\hat{\mathbf{i}}_f^T \mathbf{Q} \mathbf{Q}^T \hat{\mathbf{j}}_f^T = 0$$

$$\mathbf{R} = \hat{\mathbf{R}} \mathbf{Q}$$
$$\mathbf{S} = \mathbf{Q}^{-1} \hat{\mathbf{S}}$$

$$\hat{\mathbf{R}} = \begin{bmatrix} \hat{\mathbf{i}}_1^T \\ \vdots \\ \hat{\mathbf{i}}_F^T \\ \hat{\mathbf{j}}_1^T \\ \vdots \\ \hat{\mathbf{j}}_F^T \end{bmatrix}, \quad \hat{\mathbf{S}} = [\hat{\mathbf{s}}_1 \ \cdots \ \hat{\mathbf{s}}_P]$$

Factorization : Algorithm

1. Make a meaurement matrix $\tilde{\mathbf{W}}$ from tracked feature points.
2. SVD $\tilde{\mathbf{W}}$, then obtain $\mathbf{O}_1\mathbf{\Sigma}\mathbf{O}_2$ $(\tilde{\mathbf{W}} = \mathbf{O}_1\mathbf{\Sigma}\mathbf{O}_2)$
3. Extract $\mathbf{O}'_1, \mathbf{\Sigma}', \mathbf{O}'_2$ from $\mathbf{O}_1, \mathbf{\Sigma}, \mathbf{O}_2$
4. Obtain $\hat{\mathbf{R}} = \mathbf{O}'_1 [\mathbf{\Sigma}']^{1/2}, \quad \hat{\mathbf{S}} = [\mathbf{\Sigma}']^{1/2}\mathbf{O}'_2$
5. Determine $\mathbf{Q}$ from the orthogonal condition of $\hat{\mathbf{R}}$

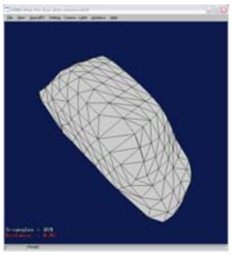
$$\hat{\mathbf{i}}_f^T \mathbf{Q}\mathbf{Q}^T \hat{\mathbf{i}}_f^T = 1, \; \hat{\mathbf{j}}_f^T \mathbf{Q}\mathbf{Q}^T \hat{\mathbf{j}}_f^T = 1, \; \hat{\mathbf{i}}_f^T \mathbf{Q}\mathbf{Q}^T \hat{\mathbf{j}}_f^T = 0$$

6. Determine $\mathbf{R}$ and $\mathbf{S}$ according to

$$\mathbf{R} = \hat{\mathbf{R}}\mathbf{Q}, \quad \mathbf{S} = \mathbf{Q}^{-1}\hat{\mathbf{S}}$$

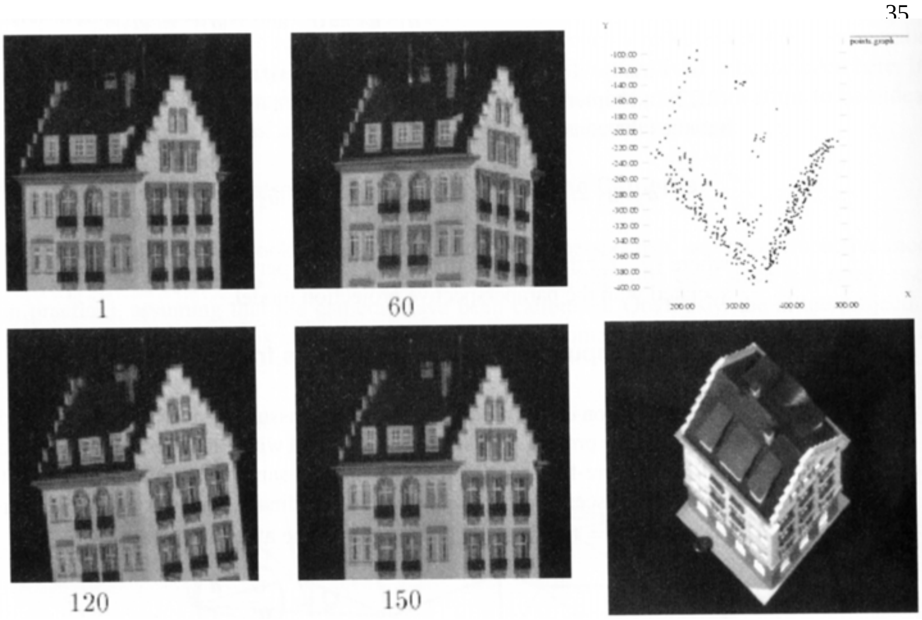
Modeled Shape via Factorization

Input Image Sequence



3D Model

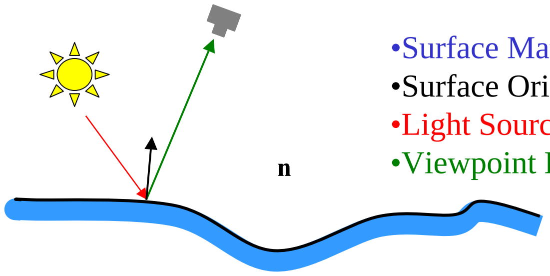
Rendered
Virtual Views



From Computer Vision／David A. Forsyth , Jean Ponce, P.268

Shape Modeling from Reflectance Analysis

What determines image intensity is



- Surface Material ρ
- Surface Orientation $\mathbf{n}$
- Light Source Direction $\mathbf{s}$
- Viewpoint Direction・・・・

Diffuse reflectance surface (Lambertian surface)

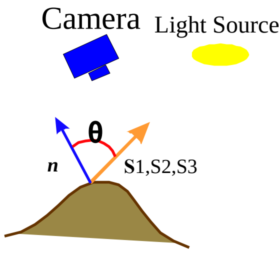
↳ Independent from Viewpoint Direction

$$L = \rho \cos\theta = \rho(\mathbf{s} \bullet \mathbf{n})$$

Photometric Stereo

Three images are captured under
three different light source directions ($\mathbf{S_1},\mathbf{S_2},\mathbf{S_3}$)

$$\begin{aligned} L &= \rho\cos\theta_s \\ &= \rho(\mathbf{S}\bullet\mathbf{n}) \end{aligned}$$



$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \rho \begin{bmatrix} (\mathbf{S}_1\bullet\mathbf{n}) \\ (\mathbf{S}_2\bullet\mathbf{n}) \\ (\mathbf{S}_3\bullet\mathbf{n}) \end{bmatrix} = \rho \begin{bmatrix} \mathbf{S}_1{}^T \\ \mathbf{S}_2{}^T \\ \mathbf{S}_3{}^T \end{bmatrix} \mathbf{n}$$

$$= \rho \begin{bmatrix} S_{1x} & S_{1y} & S_{1z} \\ S_{2x} & S_{2y} & S_{2z} \\ S_{3x} & S_{3y} & S_{3z} \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$$

$$\mathbf{L} = \rho\mathbf{S}\mathbf{n}$$

$$\mathbf{n} = \frac{\mathbf{S}^{-1}\mathbf{L}}{\rho}$$

$$\rho = |\,\mathbf{S}^{-1}\mathbf{L}\,|$$

Shape Recovery from Surface Normal **n**

Orthographics Projection Camera

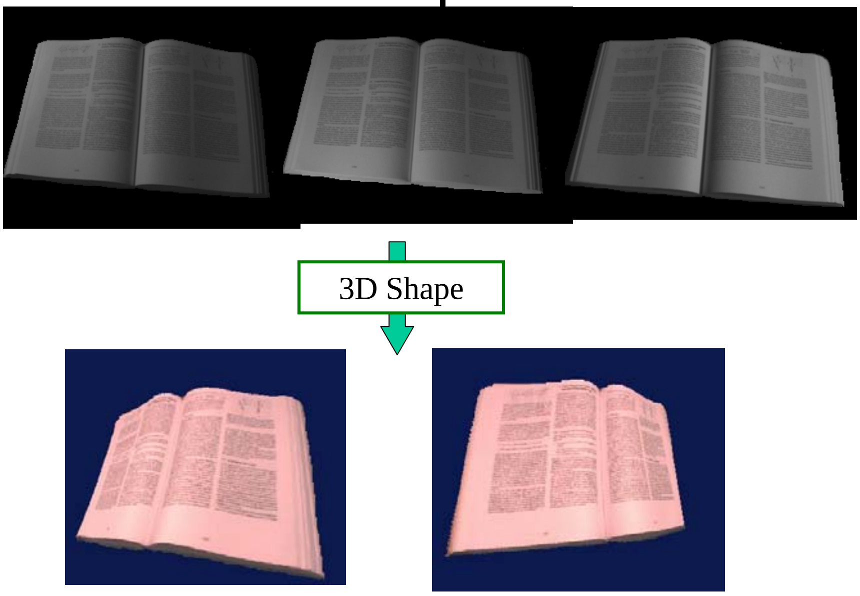
➡ *(x,y)* in image is coincide with *(X,Y)* in 3D space

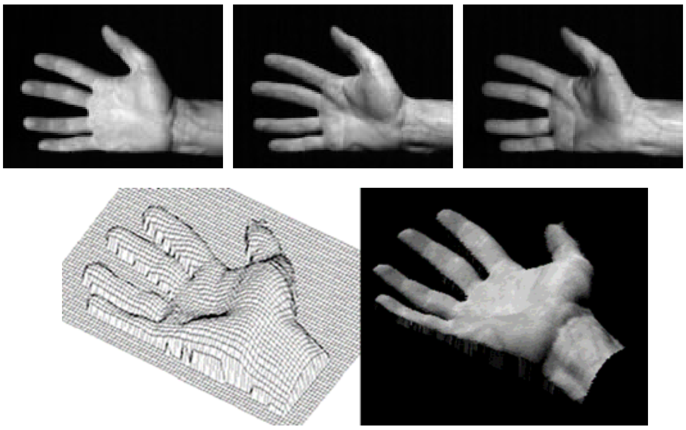
Integration (Summation) of **n**(*x*,y) ⇒ Shape *z(x,y)*

Shape Z*(x,y,z(x,y))* and Surface Normal **n***(x,y)*

$$\mathbf{n}(x,y) = \frac{1}{\sqrt{(\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2 + 1}} \left[-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1 \right]$$

Example





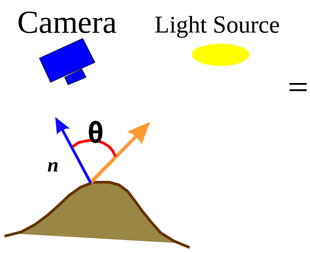
Reinhard Klette, Ryszard Kozera, and Karsten Schlüns, Shape from Shading and Photometric Stereo Methods, CITR-TR-20, May 1998

Computer Science Department of The University of Auckland, CITR at Tamaki Campus (http://www.tcs.auckland.ac.nz)

Photometric Stereo with N Light Sources

$$\begin{aligned} L &= \rho\cos\theta_s \\ &= \rho(\mathbf{S}\bullet\mathbf{n}) \end{aligned}$$

$$\begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_N \end{bmatrix} = \rho\begin{bmatrix} (\mathbf{S}_1\bullet\mathbf{n}) \\ (\mathbf{S}_2\bullet\mathbf{n}) \\ \vdots \\ (\mathbf{S}_N\bullet\mathbf{n}) \end{bmatrix} = \rho\begin{bmatrix} \mathbf{S}_1{}^T \\ \mathbf{S}_2{}^T \\ \vdots \\ \mathbf{S}_N{}^T \end{bmatrix}\mathbf{n}$$



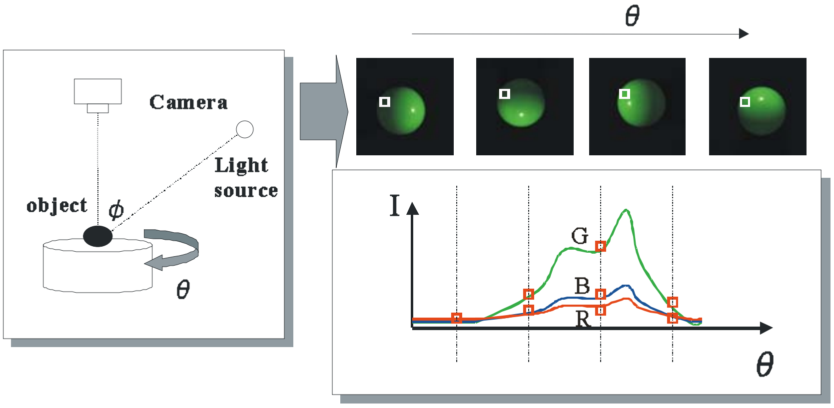
$$= \rho\begin{bmatrix} S_{1x} & S_{1y} & S_{1z} \\ S_{2x} & S_{2y} & S_{2z} \\ & \vdots & \\ S_{Nx} & S_{Ny} & S_{Nz} \end{bmatrix}\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$$

$$\mathbf{L} = \rho\mathbf{S}\mathbf{n}$$

$$\mathbf{n} = \frac{\mathbf{S}^{-1}\mathbf{L}}{\rho}$$

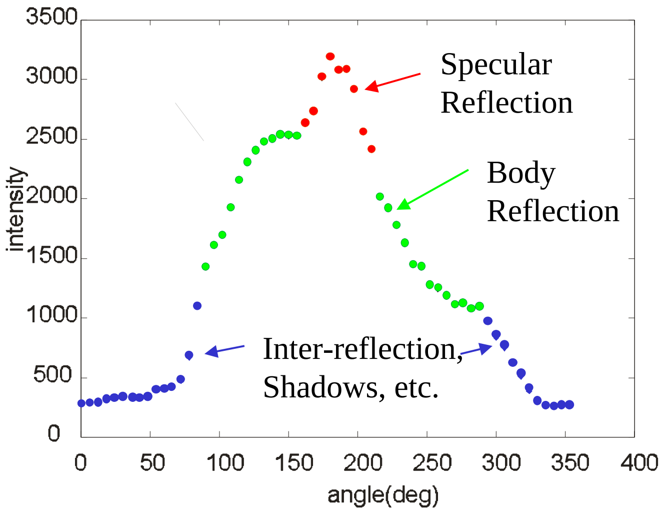
$$\rho = |\,\mathbf{S}^{-1}\mathbf{L}\,|$$

Shape Modeling from Reflectance Analysis

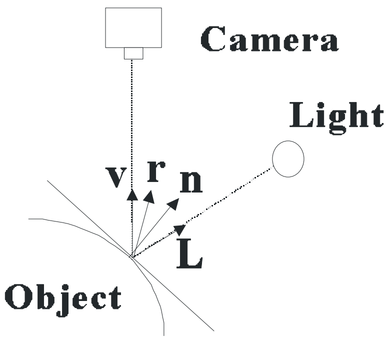


Surface normal
Body reflectance
Specular reflectance

Example of Intensity Curve



Phong’s Reflectance Model



$I=d(\mathbf{L}\cdot\mathbf{n})+s(\mathbf{r}\cdot\mathbf{v})^k$

d: body reflectance

s: specular reflectance

k: specular sharpness

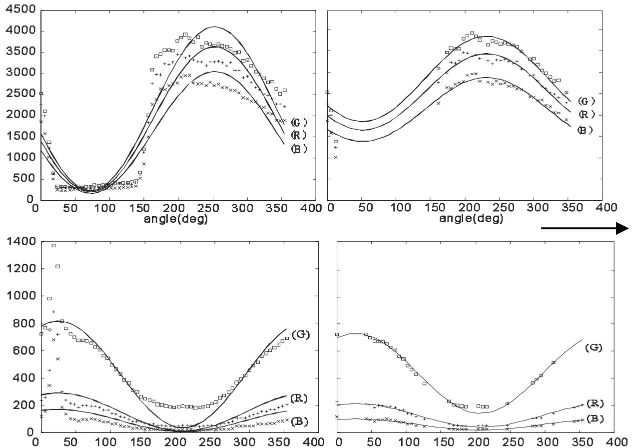
$\mathbf{n}$: surface normal

$\mathbf{L}$: light source direction

$\mathbf{v}$: camera direction

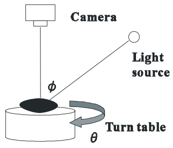
$\mathbf{r}=(2\mathbf{n}-\mathbf{L})/(|2\mathbf{n}-\mathbf{L}|)$

Fitting to body reflection



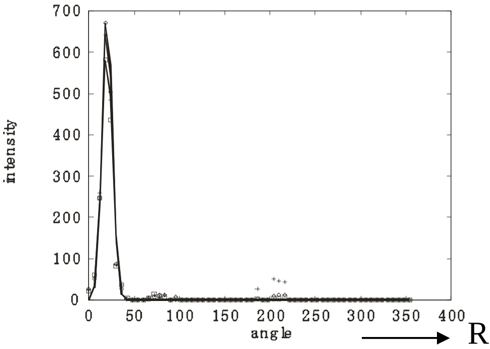
Fitting to all data

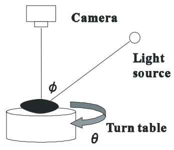
Fitting to selected data as body reflection



→ Rotation angle θ

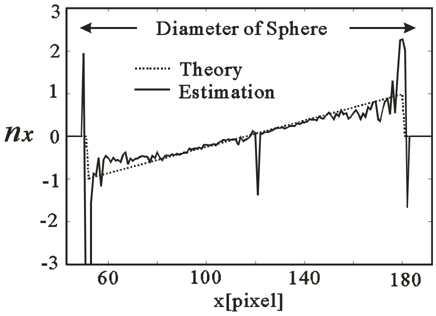
Fitting to specular reflection



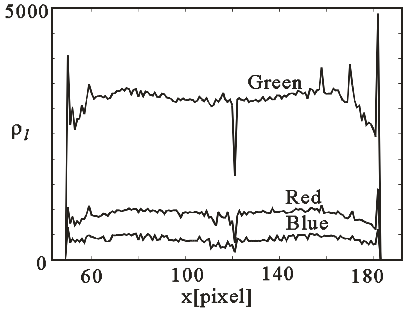


→ Rotation angle θ

Results for Sphere

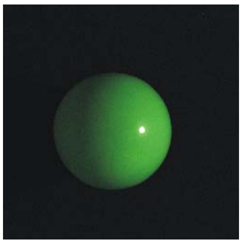


Comparison of estimated surfaced normal with theoretical value

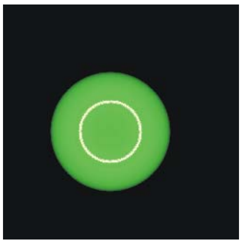


Estimated body reflectance

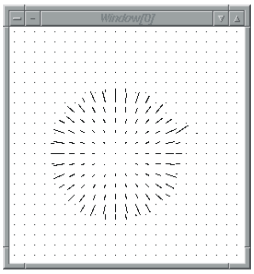
Results for Sphere



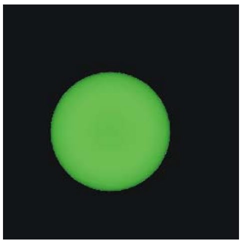
Example of input image



Recovered body reflectance from all intensity data



Recovered needle map

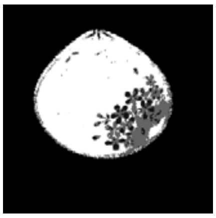


Recovered body reflectance from selected intensity data

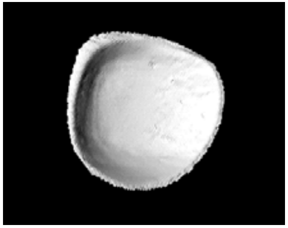
Shape and Reflectance

Example of input image

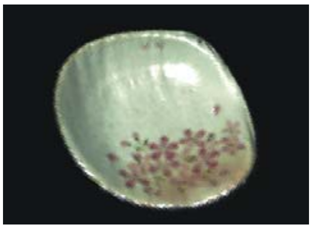
Body reflectance



Specular reflectance



3D Shape

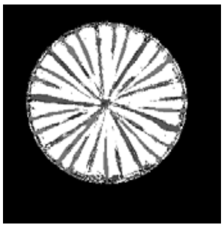


Rendered image at virtual view

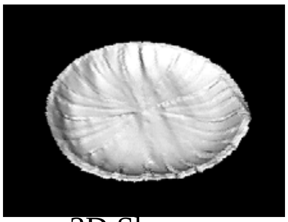
Shape and Reflectance

Example of input image

Body reflectance



Specular reflectance



3D Shape

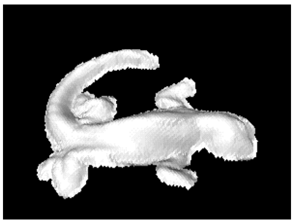
Rendered image at virtual view

Shape and Reflectance

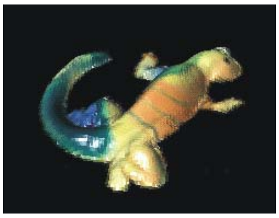
Example of
input image

Body reflectance

Specular reflectance

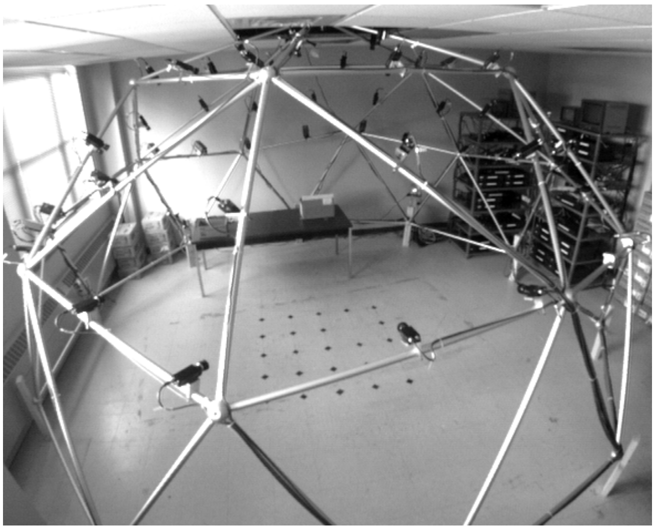


3D Shape



Rendered image at virtual view

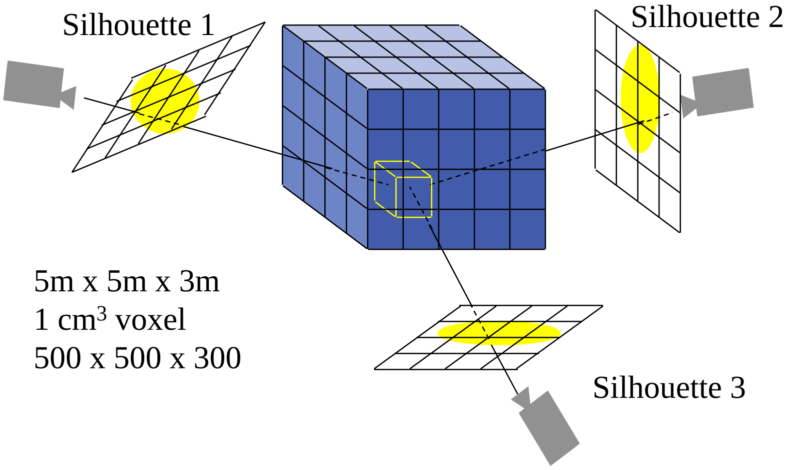
Modeling and Rendering via Multiple Cameras



Virtualized Reality: Dome with 51 Cameras in CMU

Silhouette Intersection

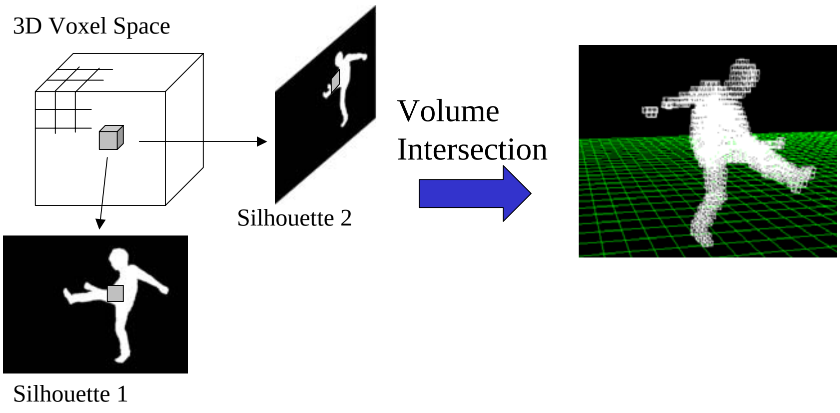
$$F(X,Y,Z)=\begin{cases}1\,;\ \text{if all the projected points are included in the silhouette}\\0\,;\ \text{otherwise}\end{cases}$$



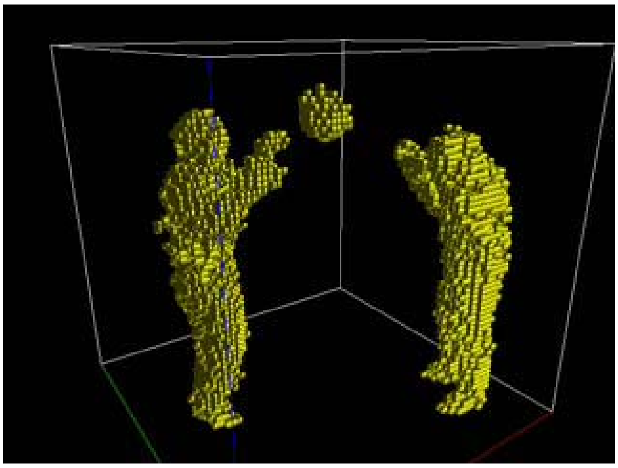
5m x 5m x 3m
1 cm^3 voxel
500 x 500 x 300

Silhouette Intersection

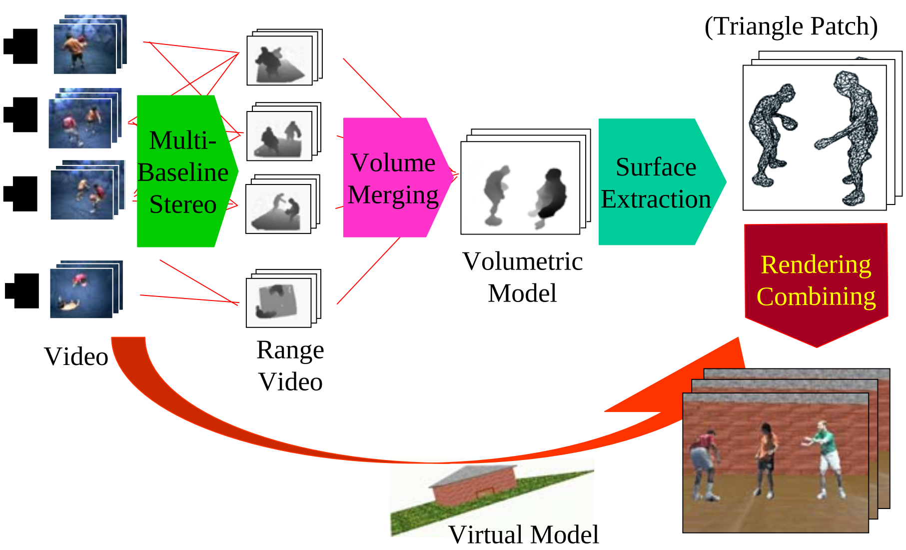
– Simple, but impossible to recover concaved shape



Example of Silhouette Intersection



Modeling and Rendering from multiple view videos



Surface Model (Triangle Patch)

Multi-Baseline Stereo

Volume Merging

Surface Extraction

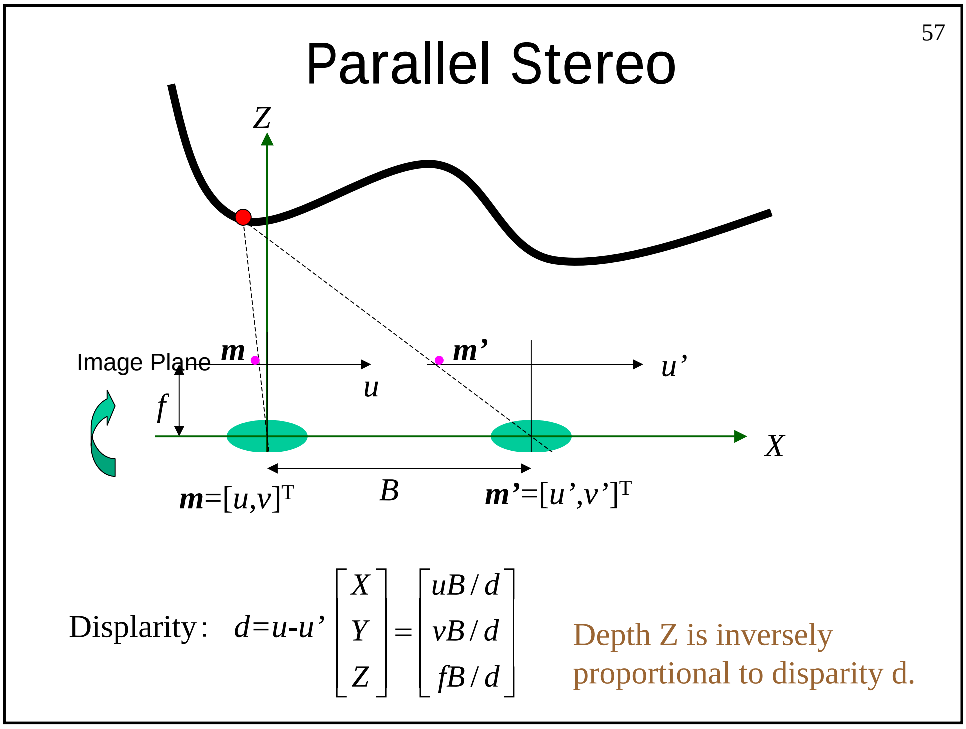
Rendering Combining

Volumetric Model

Video

Range Video

Virtual Model

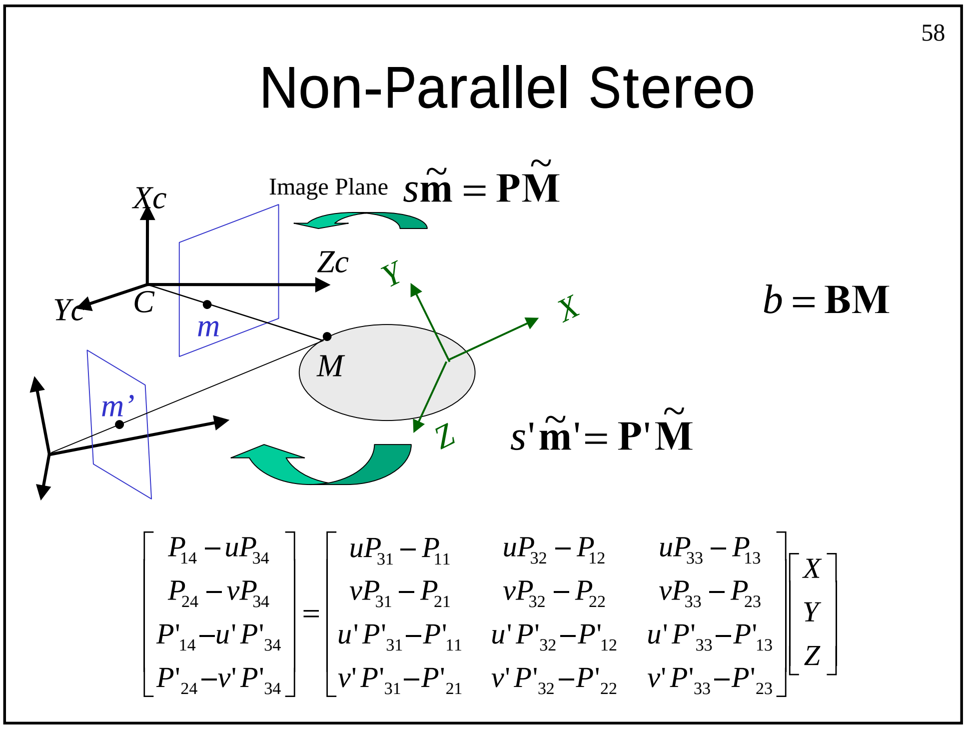


Parallel Stereo

Displarity: d=u-u’

$$\begin{bmatrix}X\\Y\\Z\end{bmatrix}=\begin{bmatrix}uB/d\\vB/d\\fB/d\end{bmatrix}$$

Depth Z is inversely proportional to disparity d.



Non-Parallel Stereo

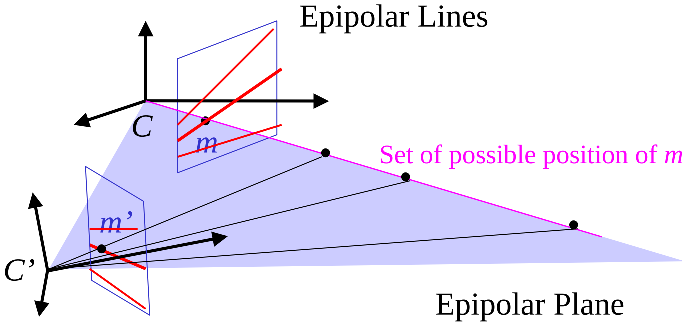
$$s\tilde{\mathbf{m}}=\mathbf{P}\tilde{\mathbf{M}}$$

$$b=\mathbf{BM}$$

$$s'\tilde{\mathbf{m}}'=\mathbf{P}'\tilde{\mathbf{M}}$$

$$\begin{bmatrix}P_{14}-uP_{34}\\P_{24}-vP_{34}\\P'_{14}-u'P'_{34}\\P'_{24}-v'P'_{34}\end{bmatrix}=\begin{bmatrix}uP_{31}-P_{11} & uP_{32}-P_{12} & uP_{33}-P_{13}\\vP_{31}-P_{21} & vP_{32}-P_{22} & vP_{33}-P_{23}\\u'P'_{31}-P'_{11} & u'P'_{32}-P'_{12} & u'P'_{33}-P'_{13}\\v'P'_{31}-P'_{21} & v'P'_{32}-P'_{22} & v'P'_{33}-P'_{23}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix}$$

Epipolar Geometry



Corresponding point should be on the epipolar line

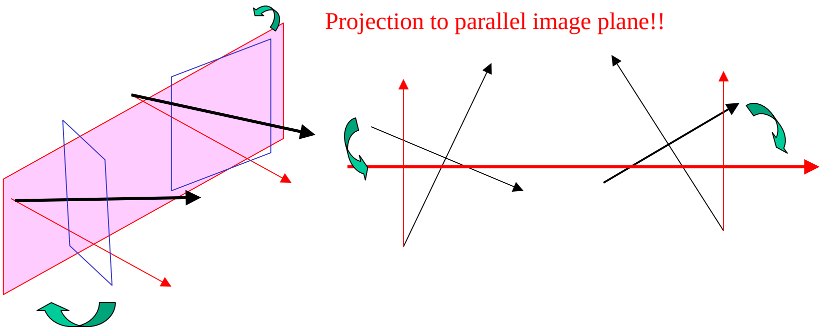
⇒ Searching of corresponding point is 1D searching.

Rectification

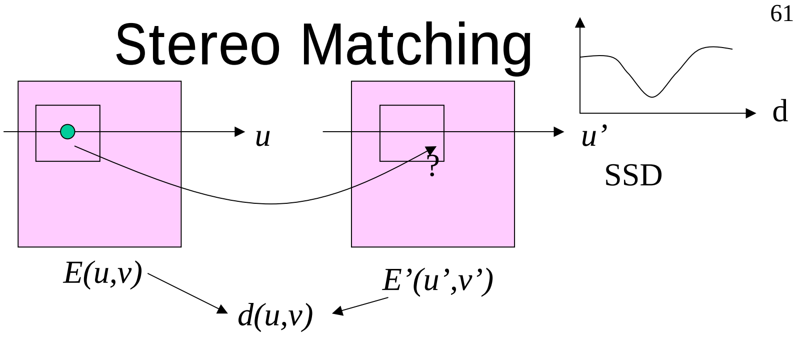
- Stereo Correspondence
  - Parallel Cameras : Epipolar lines are parallel to u axis.
  - Non-Parallel Cameras : Search along epipolar lines
- Convert to image pair captured with parallel cameras

⇒ Rectification

Projection to parallel image plane!!



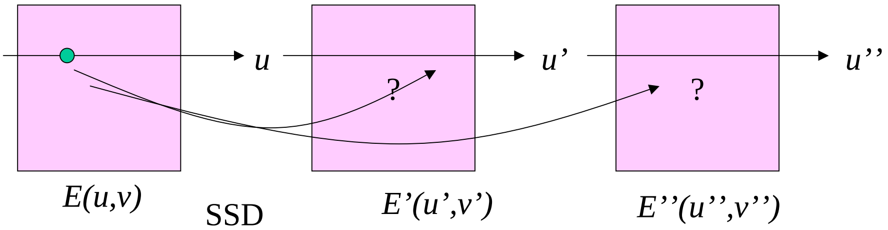
Stereo Matching

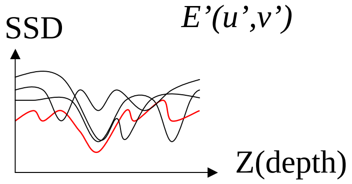


Pattern matching of small local area

$$SSD = \sum\sum | E(u+d, v) - E'(u', v') |^2$$

Multi-Baseline Stereo(MBS)





SSD curve of different camera pairs

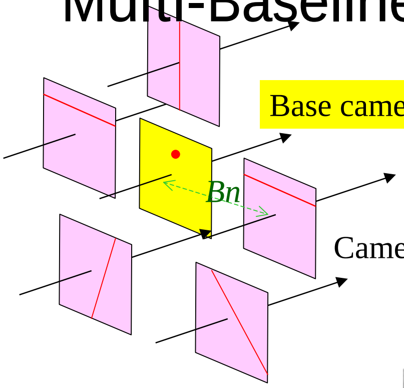
Integrate all the SSD curve

⇩

Different base-line

Multiple Baseline Stereo:MBS

Multi-Baseline Stereo(MBS)

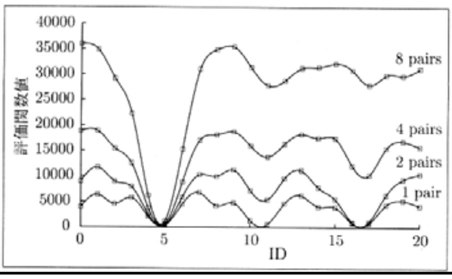


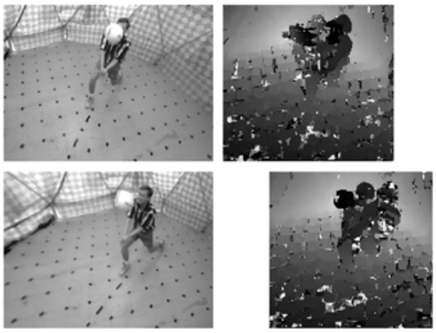
Rectified multiple cameras

Stereo between camera *n* and base camera

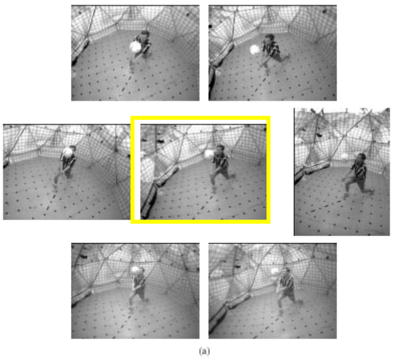
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} uBn/d \\ vBn/d \\ fBn/d \end{bmatrix}$$

$$\frac{1}{Z} = \frac{d}{fBn} \quad ; \quad \text{Inverse Distance(ID)}$$



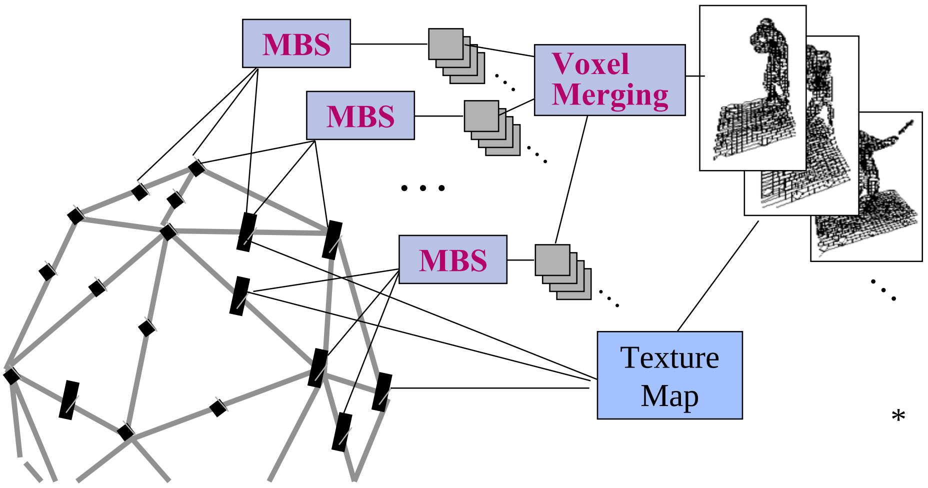


Depth images via two cameras
Up :Depth at left camera.
Dn: Depth at right camera.



Depth image via 7 cameras

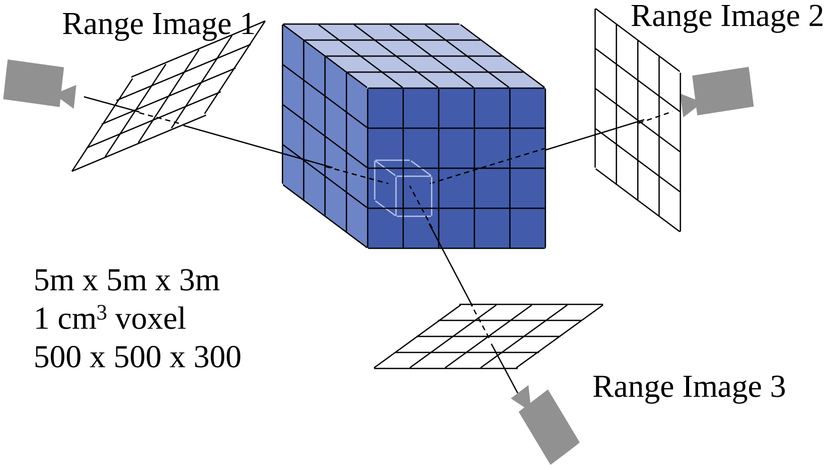
3D Complete Surface Modeling



Method for synthesizing range images

Surface representation with implicit function :

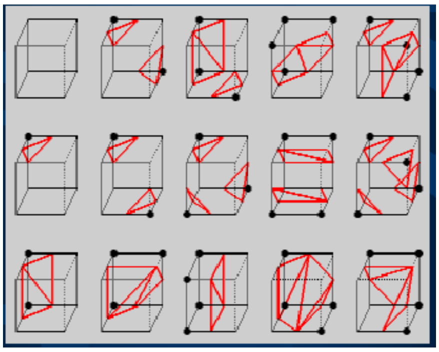
$$F(x,y,z)\begin{cases} > 0 & \text{outside} \\ = 0 & \text{on the surface} \\ < 0 & \text{inside} \end{cases}$$



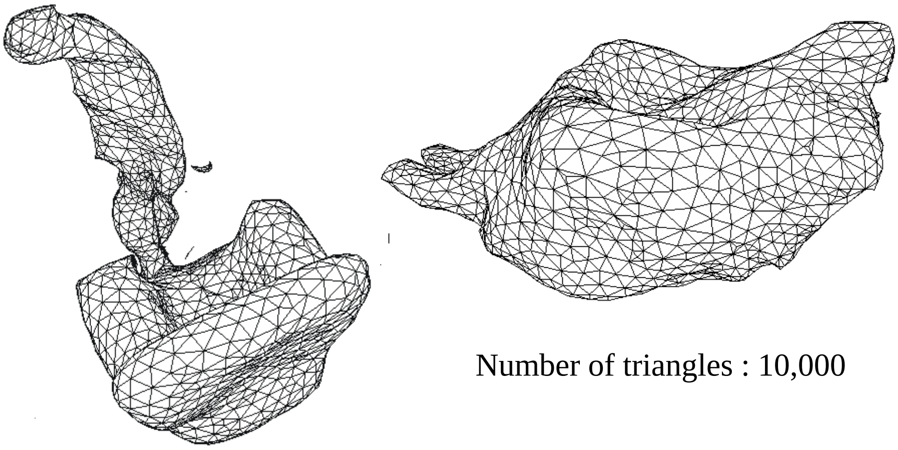
5m x 5m x 3m
1 cm^3 voxel
500 x 500 x 300

Surface extraction from the volume

- *Set of F(x,y,z)=0* →Object surface
- Marching Cubes Algorithm
  - Extract surface voxels: Voxel at which sign of F(x,y,z) changes.
  - Sign pattern around the surface voxel: Mesh pattern
  - Organizing polygon mesh



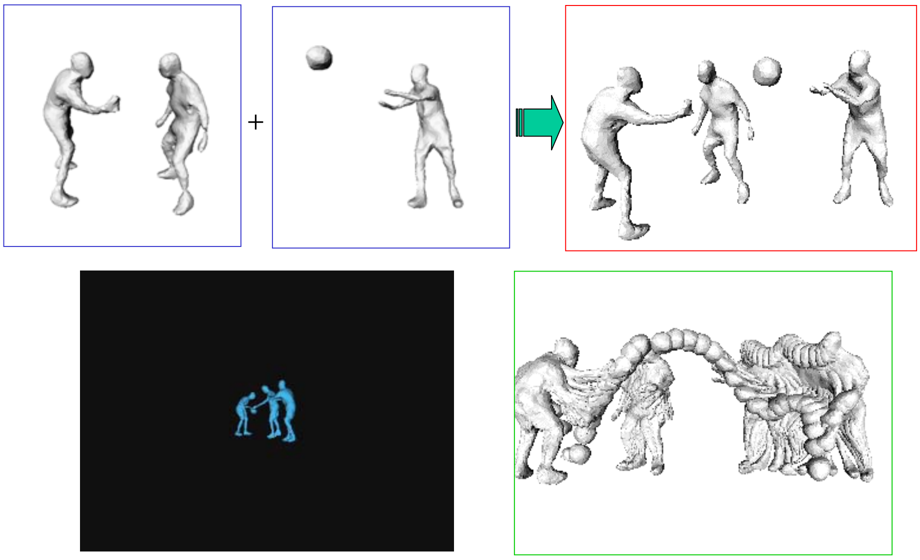
Recovered polygon mesh surface



Number of triangles : 10,000

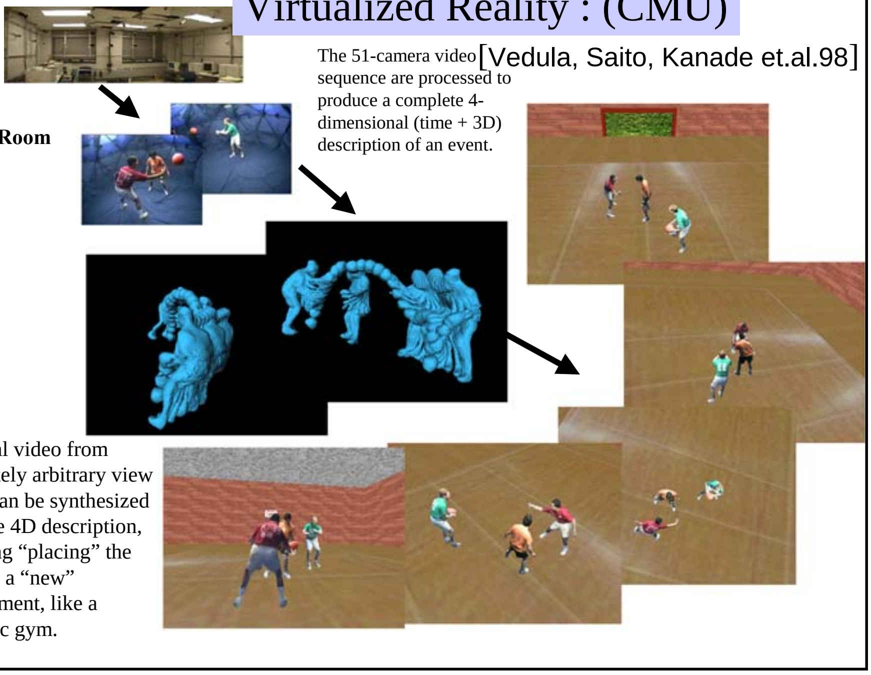
Example of mesh simplify algorithm
http://graphics.cs.uiuc.edu/~garland/software/qslim.html

3D surface mesh with motion



Virtualized Reality : (CMU)

[Vedula, Saito, Kanade et.al.98]



3D Room

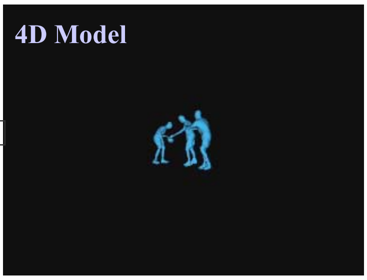
The 51-camera video sequence are processed to produce a complete 4-dimensional (time + 3D) description of an event.

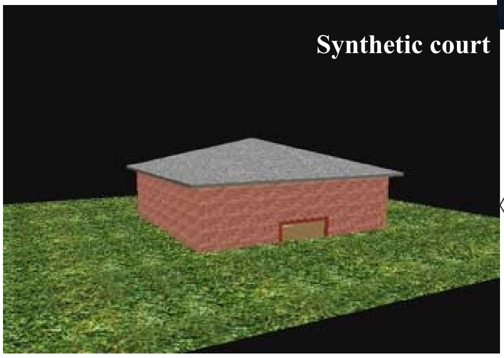
A virtual video from completely arbitrary view points can be synthesized from the 4D description, including "placing" the event in a "new" environment, like a synthetic gym.

Example:

3-Man Basketball

(These are movies.)



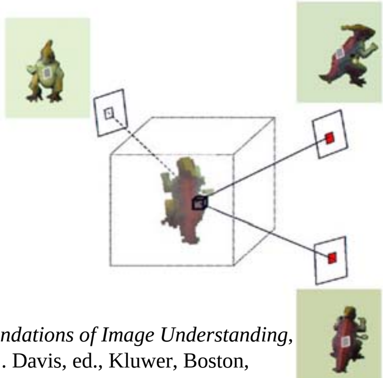


Voxel Coloring/Space Curving

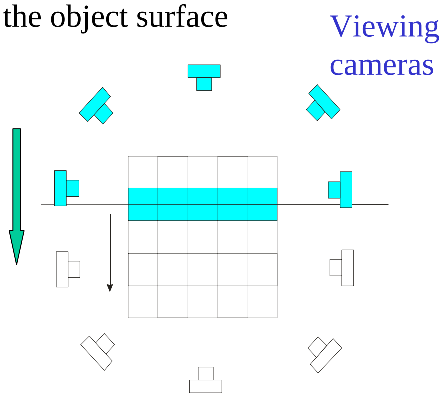
[Seitz and Dyer 97]

Checking Photo-Consistency of each voxel

If consistent, the voxel should be on the object surface

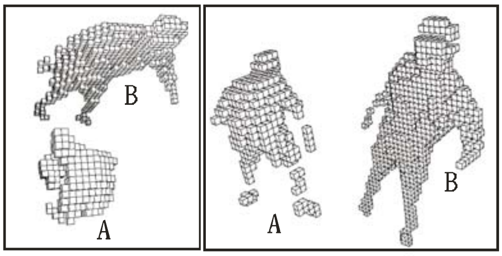


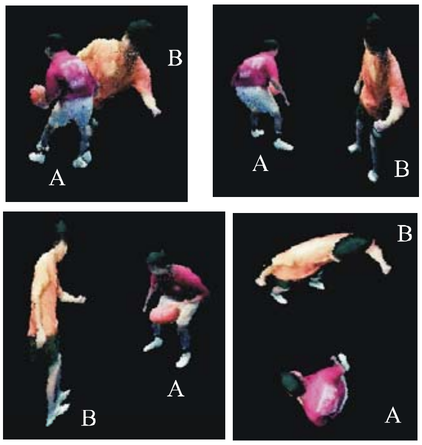
Foundations of Image Understanding, L. S. Davis, ed., Kluwer, Boston, 2001, pp. 469-489(Chapter 16, VOLUMETRIC SCENE RECONSTRUCTION FROM MULTIPLE VIEWS/Charles R. Dyer)



Shape Modeling and Rendering via Voxel Coloring in Projective Grid Space

[Saito and Kanade 99]





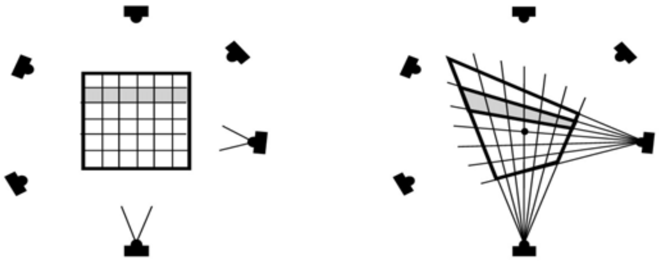
- PGS is represented in orthographic grid space.
- Displayed voxel size in (a) is 4 times larger in length than the voxel size in actual shape reconstruction.

Projective Grid Space by Weak Calibration

H. Saito, T. Kanade, Proc. CVPR’99, Vol.2, pp.49–54, 1999.

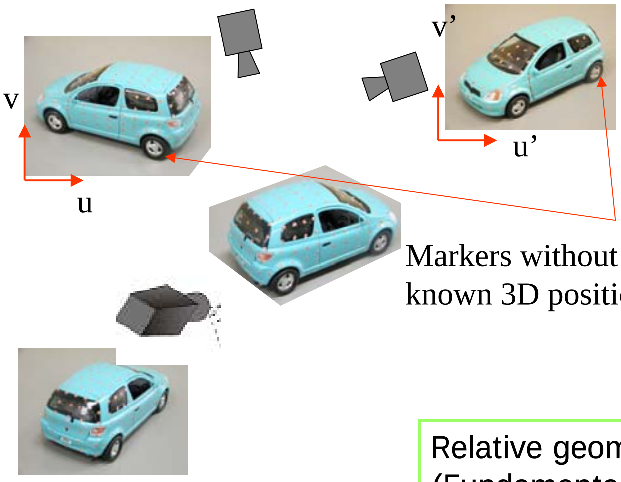
- Euclid Reconstruction (3D Space is defined independently from cameras)
  →Artificial Maker with Known 3D Position

- Projective Reconstruction (3D Space is defined dependently on cameras)
  →Natural Maker without 3D Position



(a)Euclidian Grid Space (b)Projective Grid Space

Weak Calibration

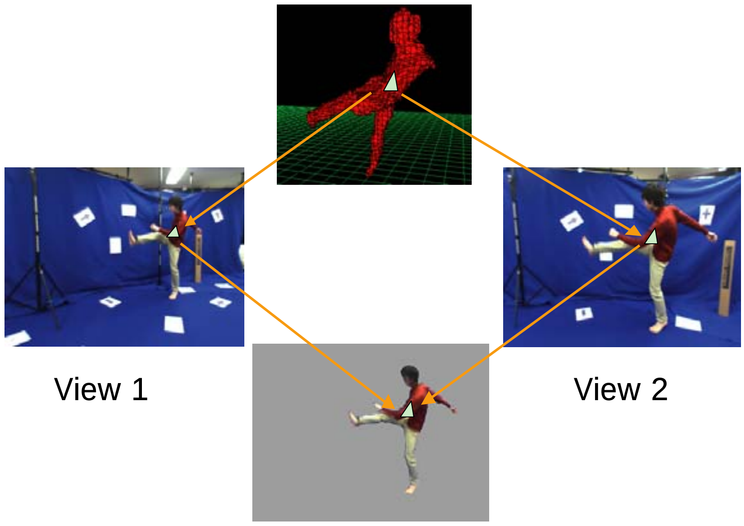


u1, v1 – u’1, v’1
u2, v2 – u’2, v’2
u3, v3 – u’3, v’3
.
.

Markers without known 3D positions

Relative geometry among cameras (Fundamental Matrices)

View Interpolation via 3D Model in PGS



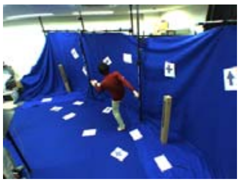
View 1

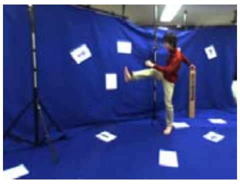
View 2

Interpolated View

Hand-Held Moving Multiple Cameras with Two Fixed Base-Cameras


Fixed Base Camera 1

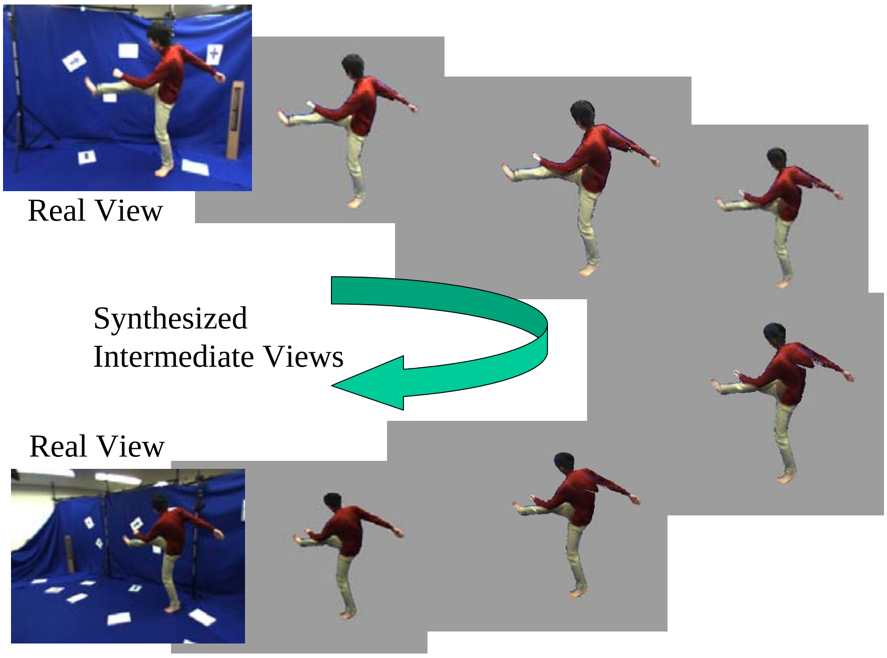

Fixed Base Camera 2


Moving Camera A


Moving Camera B

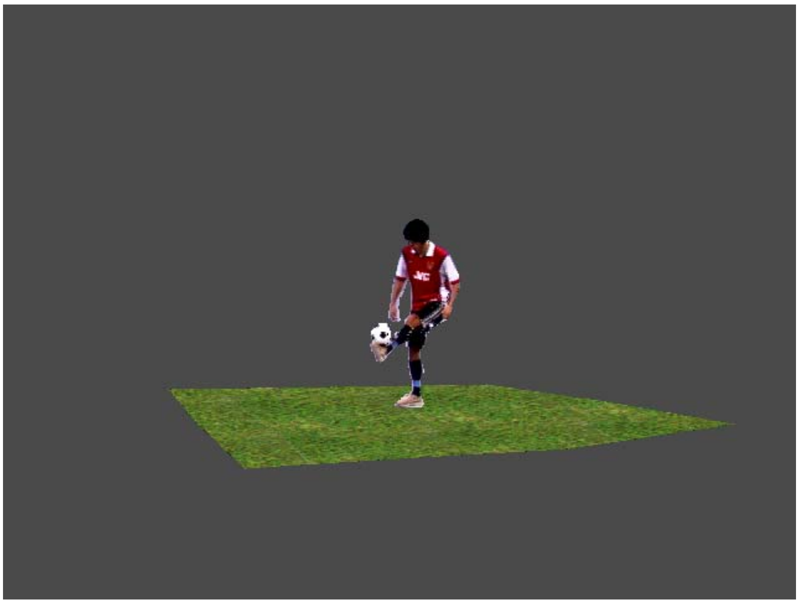

Moving Camera C

Moving by hand for tracking the object


Real View

Synthesized
Intermediate Views

Real View



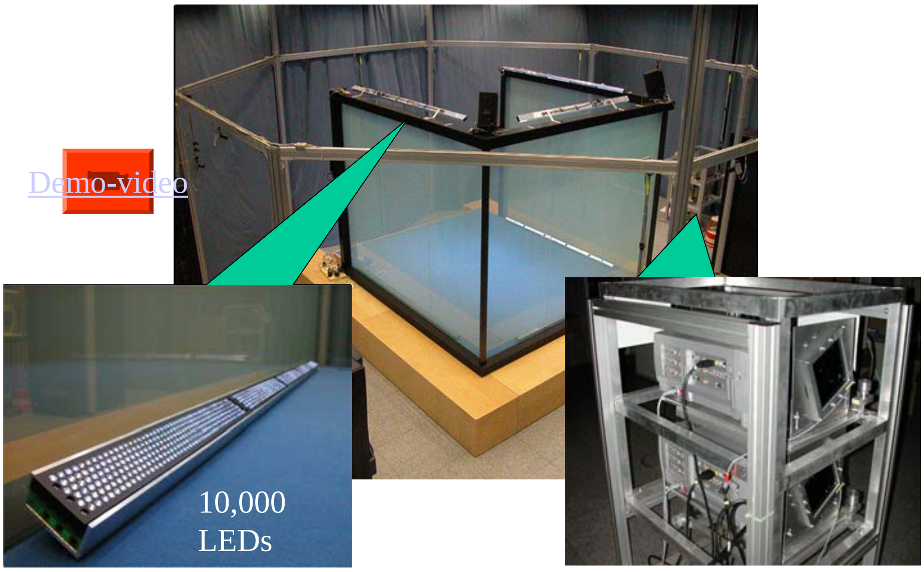
Intermediate image sequence of 5 cameras

---

blue-c Project (ETH Zurich)[Gross03]

http://blue-c.ethz.ch/

Demo-video



Rendering via Pixel Transfer

Pixel position of input image is transferred for synthesizing new viewpoint images.

Original position (x,y) -> Transferred position (u,v)

$$\begin{bmatrix} u \\ v \end{bmatrix} = f \begin{bmatrix} x \\ y \end{bmatrix}$$

Pixel transfer
Pixel position mapping

Affine Transform

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

$$\mathbf{u} = \mathbf{R}\mathbf{x} + \mathbf{t}$$

In the homogeneous coordinate representation

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

$$\tilde{\mathbf{u}} = \mathbf{H}\tilde{\mathbf{x}}$$

Homography

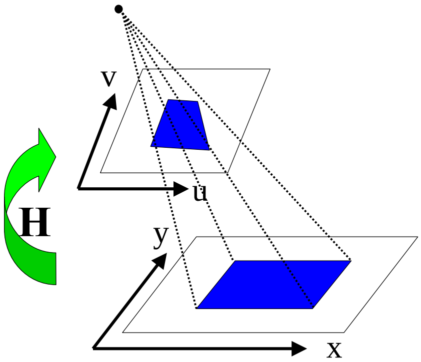
$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}a & b & c\\d & e & f\\g & h & 1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

$$\tilde{\mathbf{u}}=\mathbf{H}\tilde{\mathbf{x}}$$

If g=h=0
then equal to affine transform

$$u=\frac{ax+by+c}{gx+hy+1}$$

$$v=\frac{dx+ey+f}{gx+hy+1}$$

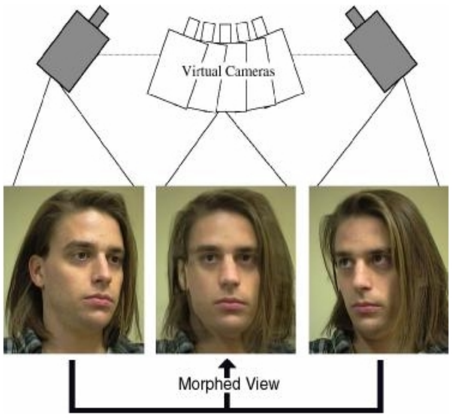


Free Viewpoint Observation

View Morphing

[Seitz and Dyer 96]

http://www-2.cs.cmu.edu/~seitz/vmorph/vmorph.html



Texture Mapping v.s. Pixel Transfer

Texture Mapping

Pixel Transfer
(View Interpolation)

View Interpolation via 3D Model

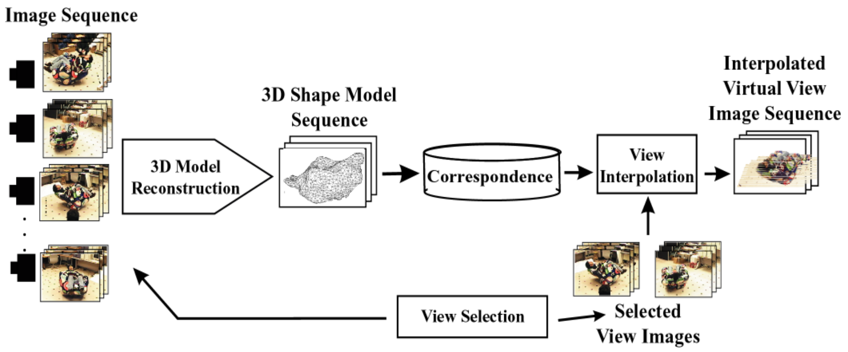


Image Sequence

3D Model Reconstruction

3D Shape Model Sequence

Correspondence

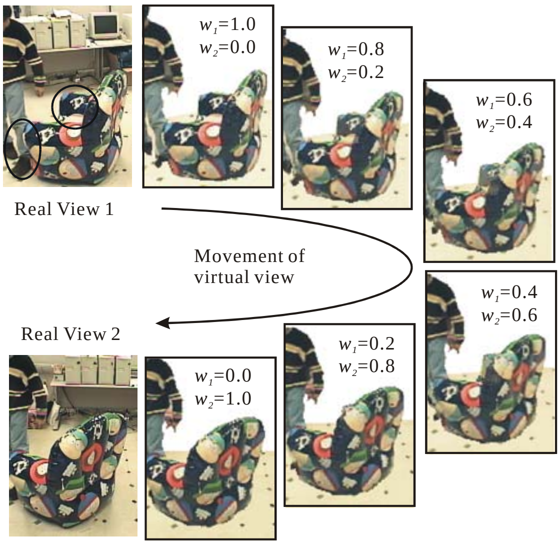
View Interpolation

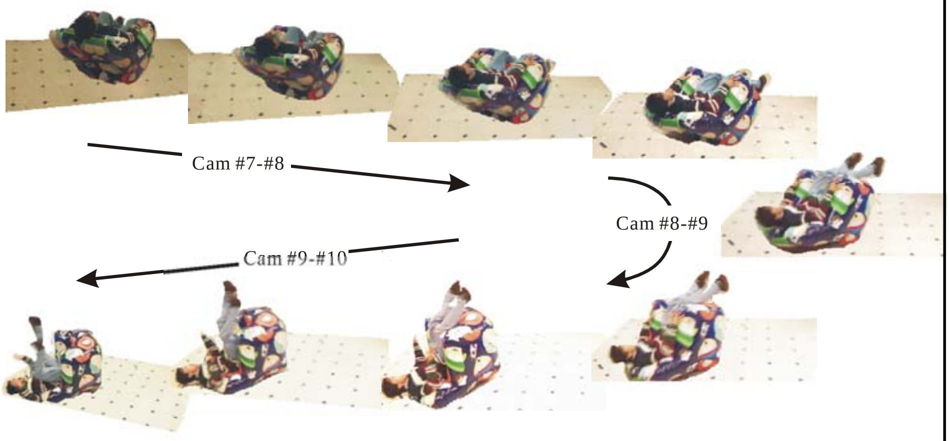
Interpolated Virtual View Image Sequence

View Selection

Selected View Images

Virtual View Images by Interpolation of Two Views





Free Viewpoint Synthesis for Soccer Scene

- Calibration of Multiple Cameras

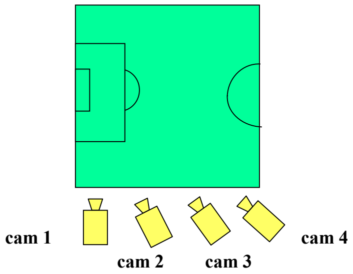
  Difficult to calibrate the cameras
  - Weak Calibration

- Shape Recovery Techniques

  Almost impossible to recover 3D shapes
  - Simple Shape Representation

- Rendering Methods
  - View Interpolation/Morphing

For soccer scenes

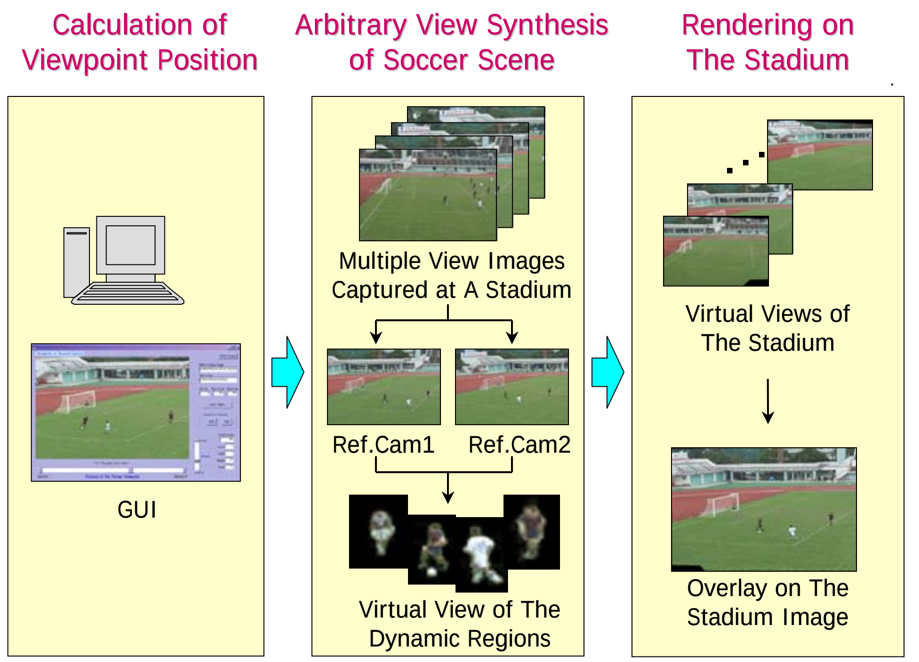



cam1


cam2


cam3


cam4



Calculation of Viewpoint Position

GUI

Arbitrary View Synthesis of Soccer Scene

Multiple View Images Captured at A Stadium

Ref.Cam1 Ref.Cam2

Virtual View of The Dynamic Regions

Rendering on The Stadium

Virtual Views of The Stadium

Overlay on The Stadium Image

View Synthesis for Dynamic Regions
(1/2)

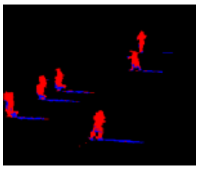
- Detection
  - Subtraction of the background
- Segmentation
  - Player/ball regions
  - Shadow regions

⇨ ・Color infomation

HSI transform

・Geometric information

Homography transform



🟥 :Player/ball regions

🟦 :Shadow regions

View Synthesis for Dynamic Regions
(2/2)

- Player Regions

⇨ Region Correspondence by Homography

Pixel Correspondence by F-Matrix

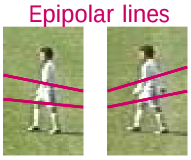
Morphing

- Shadow Regions

⇨ Pixel Correspondence by Homography

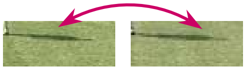
Morphing

Epipolar lines

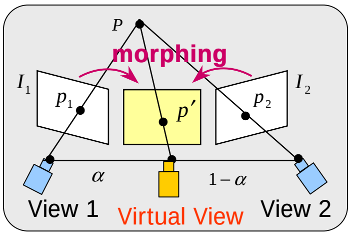


View A View B

Homography of the ground Plane

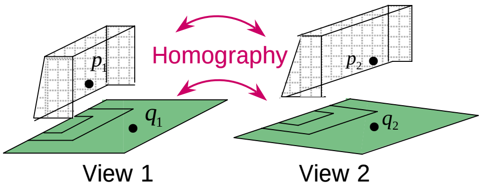


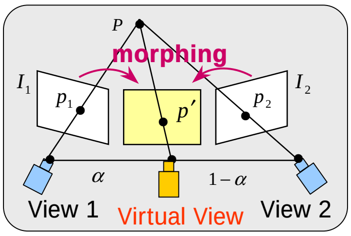
View A View B



View Synthesis for Soccer Stadium

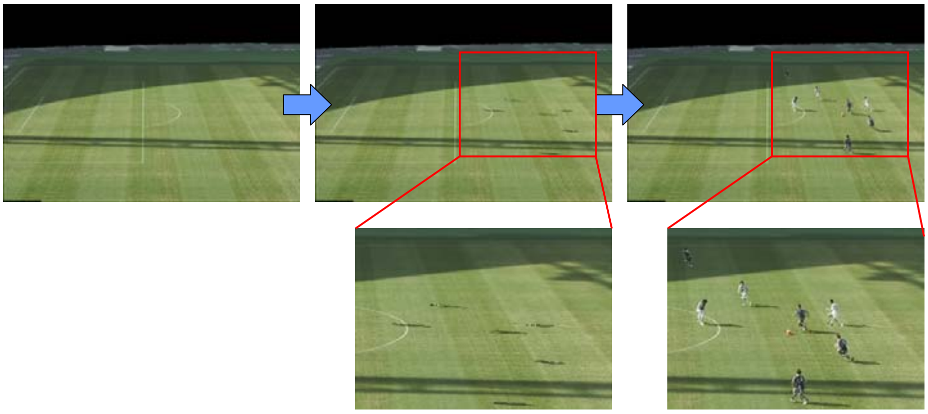
- Background Region
  : approximated to plane at infinity
  ⇨ Image mosaicking
- Field Regions
  : approximated to planes
  ⇨ Pixel correspondence by Homography
  Morphing





Soccer Scene Representation

- Superimposing dynamic regions on the soccer stadium.



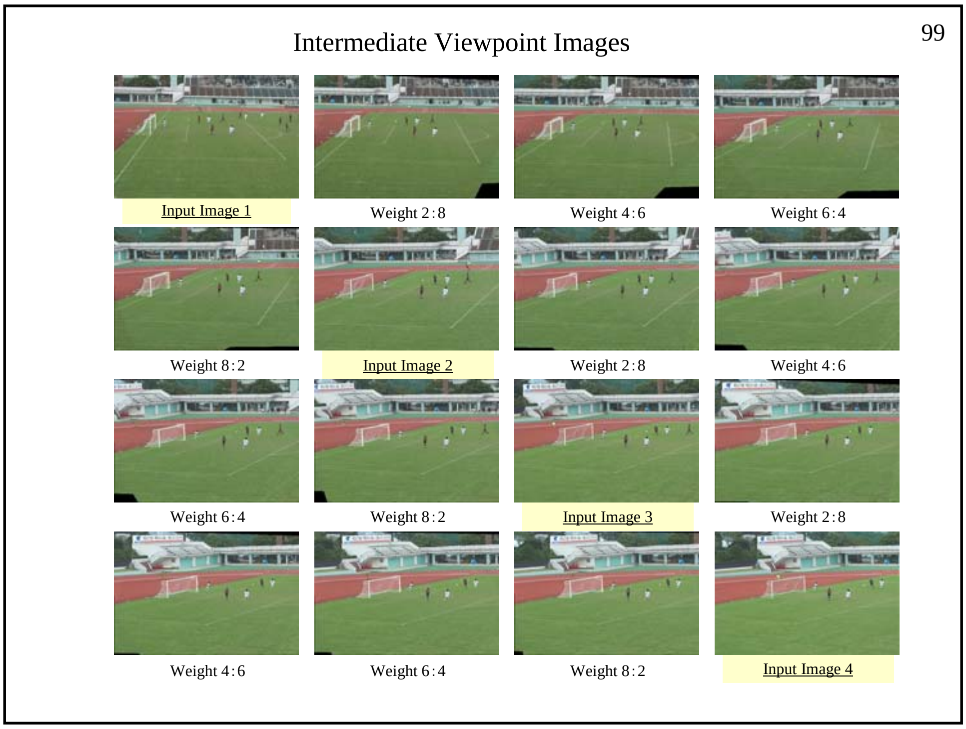
The Soccer Stadium

Overlaid shadow regions

Overlaid player regions

Virtual Viewpoint Visual Effect

ー like “The Matrix” ー



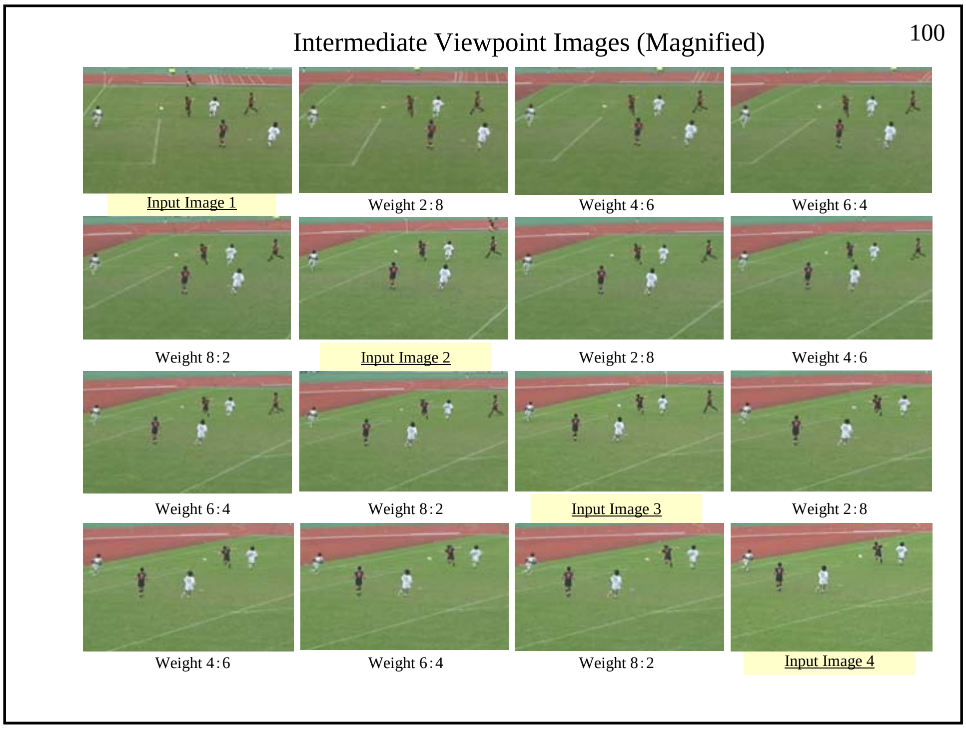
Intermediate Viewpoint Images

Input Image 1
Weight 2:8
Weight 4:6
Weight 6:4

Weight 8:2
Input Image 2
Weight 2:8
Weight 4:6

Weight 6:4
Weight 8:2
Input Image 3
Weight 2:8

Weight 4:6
Weight 6:4
Weight 8:2
Input Image 4



Intermediate Viewpoint Images (Magnified)

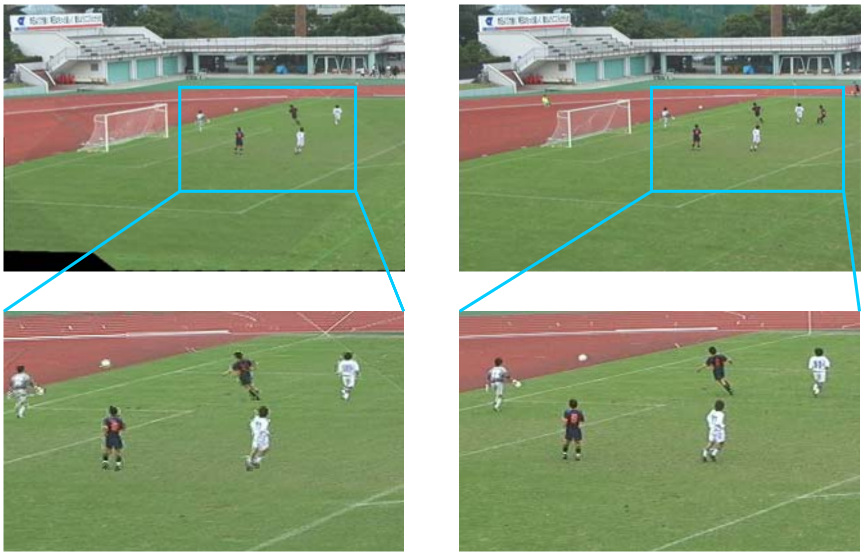
Input Image 1
Weight 2:8
Weight 4:6
Weight 6:4

Weight 8:2
Input Image 2
Weight 2:8
Weight 4:6

Weight 6:4
Weight 8:2
Input Image 3
Weight 2:8

Weight 4:6
Weight 6:4
Weight 8:2
Input Image 4

Comparison



Virtual Camera（ Cam 2-4 Weight 5:5 ）

Real Camera（Cam 3 ）

Input Video

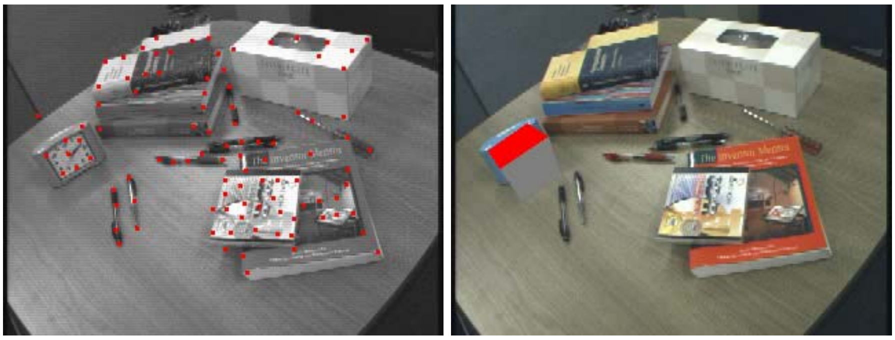
Synthesized Video
At Player’s Viewpoint

Registration

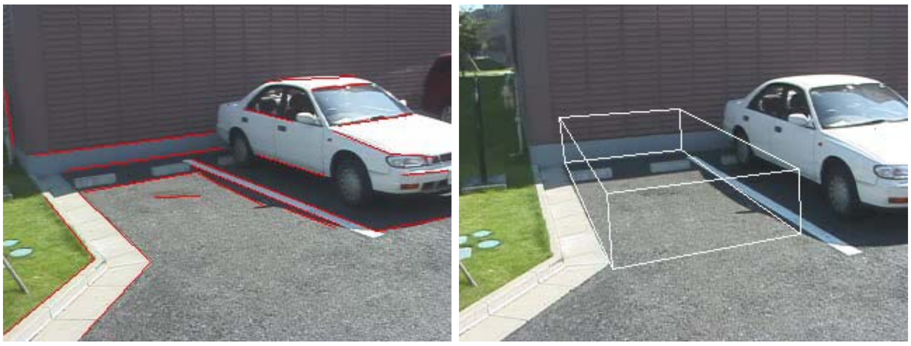
Registration for AR/MR

- Feature Tracking is Key Technology.
- Estimate Motion of Camera for Registration.

Feature point based registration



Line-segment Based Registration

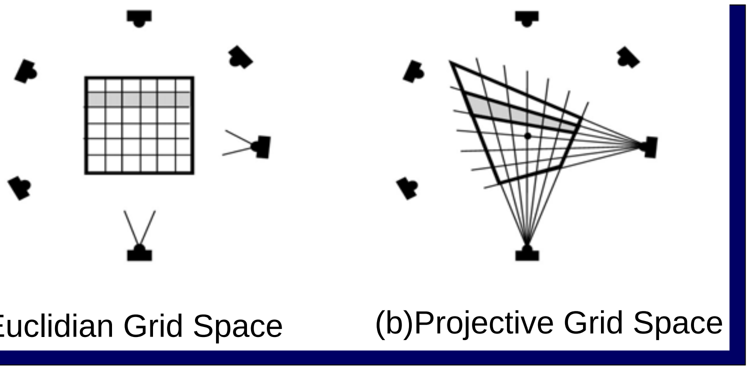


Projective Grid Space by Weak Calibration

H. Saito, T. Kanade, Proc. CVPR’99, Vol.2, pp.49–54, 1999.

•Euclid Reconstruction (3D Space is defined independently from cameras)
→**Artificial Maker with Known 3D Position**

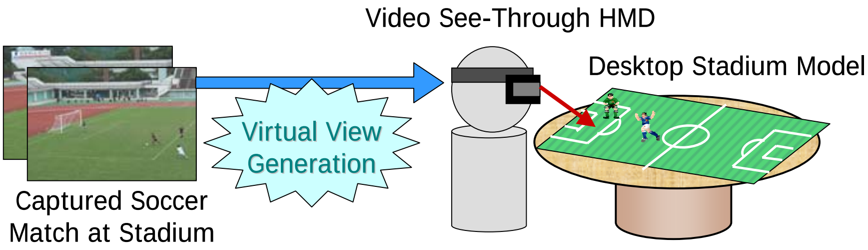
•Projective Reconstruction (3D Space is defined dependently on cameras)
→**Natural Maker without 3D Position**



(a)Euclidian Grid Space (b)Projective Grid Space

Example of Image Based Registration -Immersive Observation System-

User sees a desktop stadium model in the real world with video see-through HMD and observes dynamic objects of soccer scene overlaid onto the display.



Result 1

Frame
335

Stadium Camera 1

Stadium Camera 2

On Real Stadium Image
(Camera 1-2 w = 0.5)

On Tabletop Stadium Model
(Camera 1-2 w = 0.47 z = 1.07)

Result 2

Frame
226

Stadium Camera 3

Stadium Camera 4

On Real Stadium Image
(Camera 3-4 w = 0.9)

On Tabletop Stadium Model
(Camera 3-4 w = 0.87 z = 1.06)

Immersive Observation System

Free Viewpoint Observation on the Desktop Stadium Mode with HMD

Stadium Model Captured by HMD Camera

Result

Arbitrary View Observation with HMD

Overlaid Soccer Scene on Tabletop Stadium Model

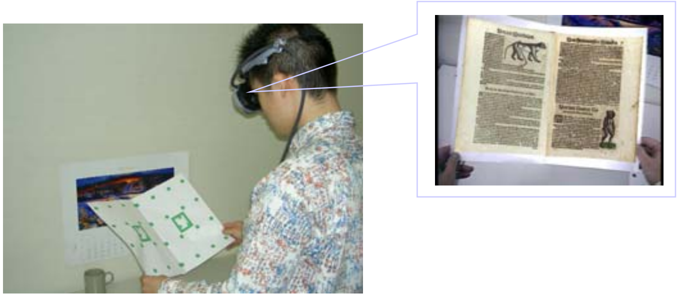
Example of Image Based Registration

Texture Overlay onto Deformable Surface Using HMD

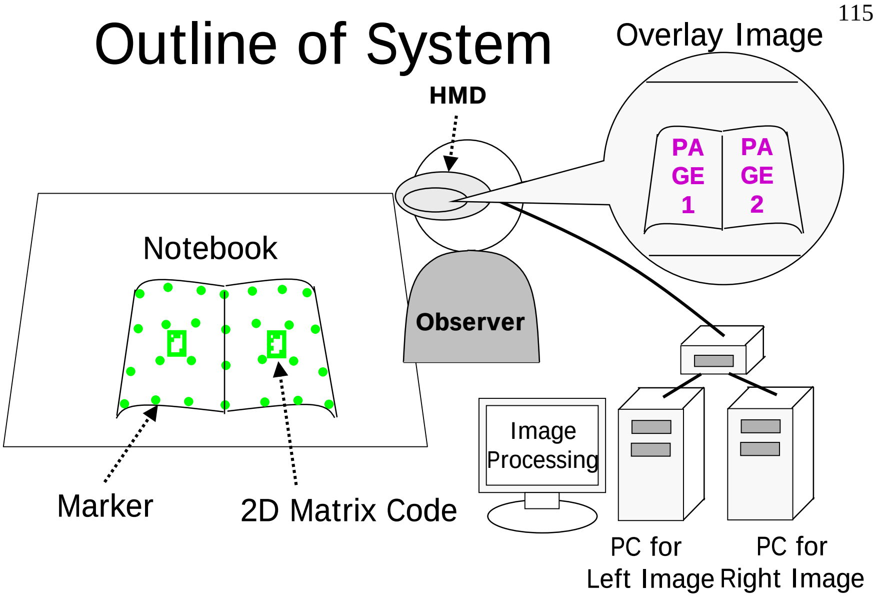
Purpose

Our propose is to overlay textures onto a deformable surface of an object in real time using a video see-through HMD.

Observer feels as if he was reading a real book.



Outline of System



Overlay Image

HMD

PA GE 1

PA GE 2

Notebook

Observer

Image Processing

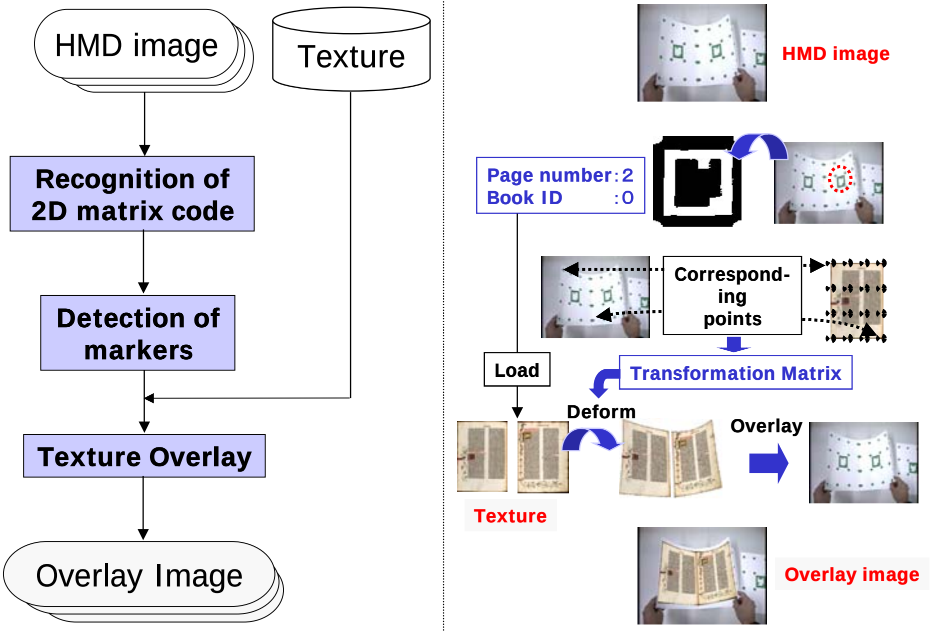
Marker

2D Matrix Code

PC for Left Image

PC for Right Image

Method for Texture Overlay



HMD image

Texture

Recognition of 2D matrix code

Detection of markers

Texture Overlay

Overlay Image

HMD image

Page number : 2
Book ID : 0

Correspond-ing points

Load

Transformation Matrix

Deform

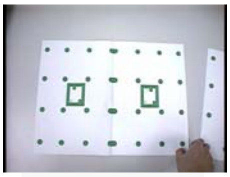
Overlay

Texture

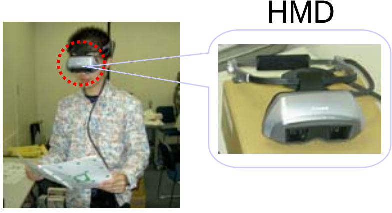
Overlay image

Implementation

Input images : 640 × 480 pixels



HMD image



Appearance of experiment

Texture images : 350 × 500 pixels

book0, page1

book0, page2

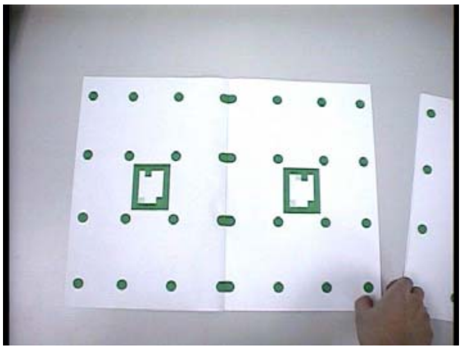
book1, page3

book1, page4

the Gutenberg Bible

the Conrad Gesner's Thierbuch

Result 1



Original image

Overlay image

Result 2

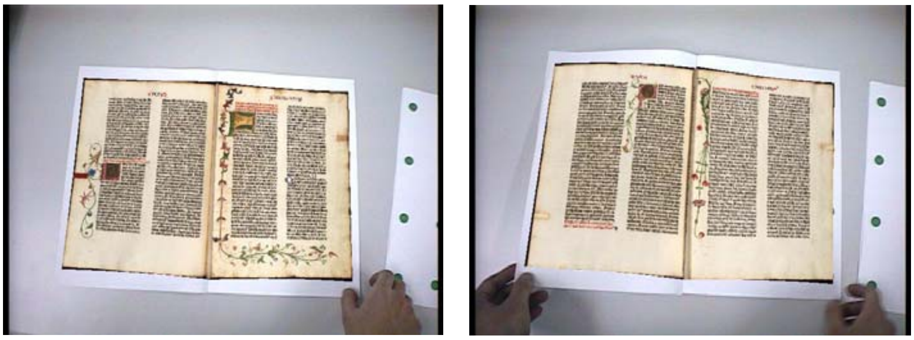
A case that a book is turned upside down.

Overlay image 1

Overlay image 2

Result 3

A case that we turn a page.

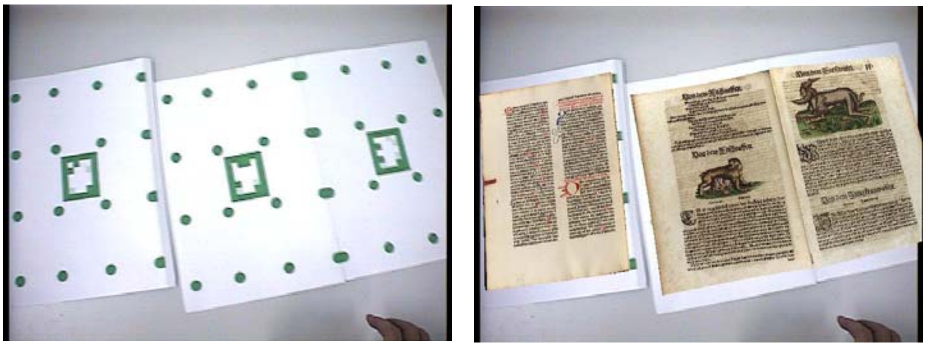


Page 1,2

Page 3,4

Result 4

A case that we see multiple books.

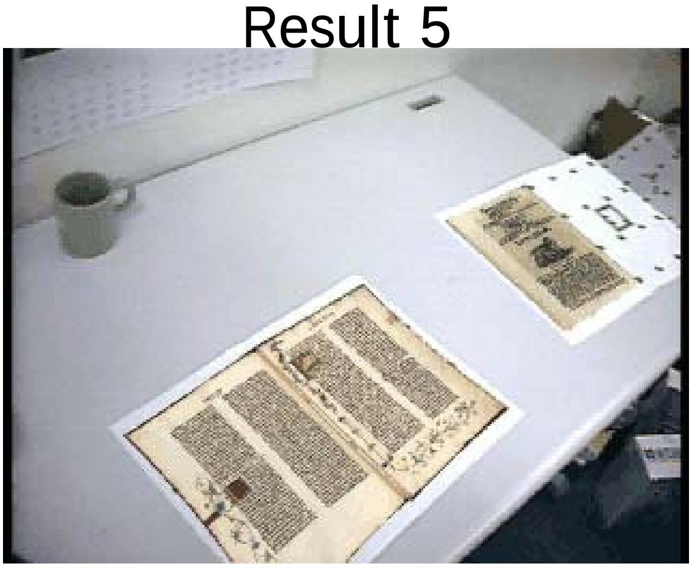


Original image

Overlay image

These results(2 ~ 4) shows that 2D matrix code is recognized correctly.

Result 5



Overlay images generated by the system

Conclusion

- Image/Video-based modeling/rendering, and registration for virtual reality application are introduced.
  - Camera geometry for capturing images
  - Modeling and rendering from image sequence, multiple cameras, reflectance analysis
  - Application of modeling and rendering to sporting scene
  - Application of image-based registration for AR/MR