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

Example: 3-Man Basketball (CMU, 1998)
(These are movies.)

Modeling and Rendering

4D Model

Synthetic court

Input sequence
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

![Diagram showing Image Coordinate System, Camera Coordinate System, and World Coordinate System]
Summary of Projection Matrix

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

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

$$
\begin{bmatrix}
  u \\
  v \\
  1 \\
\end{bmatrix}
= 
\begin{bmatrix}
  X \\
  Y \\
  Z \\
\end{bmatrix}
$$

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

Intrinsic Parameters (5DOFs)

Extrinsic Parameters (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

$$s \tilde{m} = P \tilde{M}$$
Camera Calibration

Input Data

\[ \begin{align*}
X1,Y1,Z1 & \rightarrow u1,v1 \\
X2,Y2,Z2 & \rightarrow u2,v2 \\
X3,Y3,Z3 & \rightarrow u3,v3 \\
X4,Y4,Z4 & \rightarrow u5,v5 \\
\end{align*} \]

Linear Solution for Estimating \( P \)

\[ s\hat{m} = PM \]

\[ \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \]

\[ (X_n,Y_n,Z_n) \rightarrow (u_n,v_n) \]

\[ \begin{align*}
P_{11}X_n + P_{12}Y_n + P_{13}Z_n + P_{14}X_nu_n - P_{13}Z_nu_n - P_{14}u_n &= 0 \\
P_{21}X_n + P_{22}Y_n + P_{23}Z_n + P_{24}X_nv_n - P_{23}Z_nv_n - P_{24}v_n &= 0 \\
\end{align*} \]

\[ \begin{bmatrix} X_n \\ Y_n \\ Z_n \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -X_n & -Y_n & Z_n & 0 \\ 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -X_n & -Y_n & Z_n \\ 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -X_n & -Y_n & Z_n \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ \end{bmatrix} = \begin{bmatrix} P_{11}u_n \\ P_{12}v_n \\ \end{bmatrix} \]

\[ \begin{bmatrix} P_{21} \\ P_{22} \\ P_{23} \\ P_{24} \end{bmatrix} \]

\[ n > 11/2 \text{ markers are required for estimating } P \]
Method for extracting intrinsic and extrinsic parameters from projection matrix $P$

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

Projection Matrix $P \rightarrow$ Intrinsic Matrix $A$

Extrinsic Matrix and Vector $R, t$

$$P_u = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\
P_{21} & P_{22} & P_{23} & P_{24} \\
P_{31} & P_{32} & P_{33} & P_{34}
\end{bmatrix} = \begin{bmatrix} a_1 \\
a_2 \\
a_3
\end{bmatrix} \quad [R, t] = \begin{bmatrix} R_{11} & R_{21} & R_{31} & t_x \\
R_{12} & R_{22} & R_{32} & t_y \\
R_{13} & R_{23} & R_{33} & t_z
\end{bmatrix} = \begin{bmatrix} r_1 \\
r_2 \\
r_3
\end{bmatrix}$$

Property of Rotation Matrix

$$|r_1| = |r_2| = |r_3| = 1 \quad r_1 \cdot r_2 = 0 \quad r_2 \cdot r_3 = 0 \quad r_3 \cdot r_1 = 0$$

$$r_1 \times r_2 = r_3 \quad r_2 \times r_3 = r_1 \quad r_3 \times r_1 = r_2$$

The property of rotation matrix provide the following solution.

$$r_3 = \mp \frac{a_3}{|a_3|} \quad r_1 = \frac{1}{|a_2 \times a_3|} (a_2 \times a_3) \quad r_2 = r_3 \times r_1$$

Intrinsic parameter matrix $A$ can be obtained as

$$\cos \theta = -\frac{(a_1 \times a_3) \cdot (a_2 \times a_3)}{|a_1 \times a_3| \cdot |a_2 \times a_3|}$$

$$\alpha_u = \frac{|a_1 \times a_3|}{|a_3|^2} \sin \theta, \quad \alpha_v = \frac{|a_2 \times a_3|}{|a_3|^2} \sin \theta$$
**Self Calibration**

- **Camera calibration without markers**
  


**Weak Calibration**

- Markers without known 3D positions
- **Relative geometry among cameras (Fundamental Matrices)**
Fundamental Matrix

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

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

One corresponding point gives the following equation:

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

n corresponding points

\[
\begin{bmatrix}
  x_1 x_1' & x_1 y_1' & x_1 y_1' & y_1 x_1' & y_1 y_1' & y_1 x_1' & y_1 y_1' \\
  x_2 x_2' & x_2 y_2' & x_2 y_2' & y_2 x_2' & y_2 y_2' & y_2 x_2' & y_2 y_2' \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  x_n x_n' & x_n y_n' & x_n y_n' & y_n x_n' & y_n y_n' & y_n x_n' & y_n y_n'
\end{bmatrix}
\begin{bmatrix}
  F_{11} \\
  F_{12} \\
  F_{13} \\
  F_{21} \\
  F_{22} \\
  F_{23} \\
  F_{31} \\
  F_{32} \\
  F_{33}
\end{bmatrix} =
\begin{bmatrix}
  1 \\
  1 \\
  1 \\
  1 \\
  1 \\
  1 \\
  1 \\
  1 \\
  1
\end{bmatrix}
\]

\[F_{33} = 1\]
**Strong/Weak Calibration**

<table>
<thead>
<tr>
<th>Strong Calibration</th>
<th>Weak Calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>To be estimated</td>
<td></td>
</tr>
<tr>
<td>Projection Matrix</td>
<td>Fundamental Matrix</td>
</tr>
<tr>
<td>$\begin{bmatrix} u \ v \ 1 \end{bmatrix} = P \begin{bmatrix} X \ Y \ Z \ 1 \end{bmatrix}$</td>
<td>$\begin{bmatrix} u_1 \ v_1 \ 1 \end{bmatrix} = F \begin{bmatrix} x_2 \ y_2 \ 1 \end{bmatrix}$</td>
</tr>
<tr>
<td>Required data</td>
<td></td>
</tr>
<tr>
<td>3D Space — 2D Image Correspondences</td>
<td>2D Image— 2D Image Correspondences</td>
</tr>
<tr>
<td>(Artificial Markers required)</td>
<td>(Natural features)</td>
</tr>
<tr>
<td>Size</td>
<td></td>
</tr>
<tr>
<td>Available</td>
<td>Unknown</td>
</tr>
<tr>
<td>Shape</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>(Ambiguity in Projective Transform)</td>
</tr>
</tbody>
</table>

**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

- 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

Camera moves toward $x$ direction

Epipolar lines are parallel with $x$ axis

Epipolar Plane Image

\[
\begin{align*}
X &= \frac{uB}{d} \\
Y &= \frac{vB}{d} \\
Z &= \frac{fB}{d}
\end{align*}
\]

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

Line Detection in EPI

Epipolar Plane
**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

- 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

\[
\tilde{W} = \begin{bmatrix}
\tilde{u}_{11} & \cdots & \tilde{u}_{1p} \\
\vdots & \ddots & \vdots \\
\tilde{u}_{F1} & \cdots & \tilde{u}_{FP}
\end{bmatrix}
\begin{bmatrix}
\tilde{v}_{11} & \cdots & \tilde{v}_{1p} \\
\vdots & \ddots & \vdots \\
\tilde{v}_{F1} & \cdots & \tilde{v}_{FP}
\end{bmatrix}
\]

\[
\tilde{u}_{fp} = u_{fp} - a_f \\
\tilde{v}_{fp} = v_{fp} - b_f
\]

\[
a_f = \frac{1}{P} \sum_{p=1}^{P} u_{fp} \\
b_f = \frac{1}{P} \sum_{p=1}^{P} v_{fp}
\]
Shape and Motion under Orthographic Projection

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

\[
\begin{align*}
\mathbf{u}_{fp} &= \mathbf{u}_{fp} - \mathbf{a}_f \\
u_{fp} &= \mathbf{j}_f (\mathbf{s}_p - \mathbf{t}_f) \\
v_{fp} &= \mathbf{j}_f (\mathbf{s}_p - \mathbf{t}_f) \\
\end{align*}
\]

\[
\mathbf{u}_{fp} = \mathbf{i}_f^T \mathbf{s}_p - \mathbf{t}_f \\
u_{fp} = \mathbf{j}_f^T \mathbf{s}_p - \mathbf{t}_f \\
\]

\[
\mathbf{W} = \begin{bmatrix}
\tilde{u}_{t1} & \cdots & \tilde{u}_{tp} \\
\vdots & \ddots & \vdots \\
\tilde{u}_{f1} & \cdots & \tilde{u}_{fp} \\
\tilde{v}_{t1} & \cdots & \tilde{v}_{tp} \\
\vdots & \ddots & \vdots \\
\tilde{v}_{f1} & \cdots & \tilde{v}_{fp} \\
\end{bmatrix}
= \begin{bmatrix}
\mathbf{i}_f^T \mathbf{s}_1 & \cdots & \mathbf{i}_f^T \mathbf{s}_p \\
\vdots & \ddots & \vdots \\
\mathbf{j}_f^T \mathbf{s}_1 & \cdots & \mathbf{j}_f^T \mathbf{s}_p \\
\vdots & \ddots & \vdots \\
\mathbf{j}_f^T \mathbf{s}_1 & \cdots & \mathbf{j}_f^T \mathbf{s}_p \\
\end{bmatrix}
= \begin{bmatrix}
\mathbf{i}_f^T \\
\vdots \\
\mathbf{i}_f^T \\
\mathbf{j}_f^T \\
\vdots \\
\mathbf{j}_f^T \\
\end{bmatrix}
\]

\[
P \text{(number of feature points)}
\]

\[
\text{RANK of } \mathbf{W} \text{ is 3}
\]
Segmentation of Measurement Matrix into Shape and Motion

\[ \tilde{W} = O_1 \Sigma O_2 = \begin{bmatrix} P \mid \Sigma \mid \Sigma' \mid O_1 \mid O_2 \mid \Sigma'' \mid O_2' \mid O_2'' \end{bmatrix} \]

\[ O_1^T O_1 = O_2^T O_2 = I \]

\[ \Sigma = \begin{bmatrix} \sigma_1 & 0 \\ \vdots & \ddots \\ 0 & \sigma_p \end{bmatrix} \]

\[ \sigma_1, \sigma_2, \ldots, \sigma_p : \]

Eigen values of \( \tilde{W} \)

\[ \hat{W} = \begin{bmatrix} O_1' \mid \Sigma' \mid O_2' \mid \Sigma'' \mid O_2'' \end{bmatrix} \]

\[ \hat{R} = \begin{bmatrix} O_1' \mid \Sigma'' / 2 \mid \hat{S} = [\Sigma'' / 2] O_2' \end{bmatrix} \]

Expected factorization

\[ \tilde{W} = \begin{bmatrix} \bar{u}_1 \cdots \bar{u}_p \\ \vdots \vdots \vdots \vdots \vdots \vdots \\ \bar{u}_{FP} \cdots \bar{u}_{FP} \\ \bar{v}_1 \cdots \bar{v}_p \\ \vdots \vdots \vdots \vdots \vdots \\ \bar{v}_{FP} \cdots \bar{v}_{FP} \end{bmatrix} = \begin{bmatrix} i_1^T \cdots i_p^T \\ \vdots \vdots \vdots \vdots \vdots \vdots \\ i_F^T \cdots i_F^T \\ \bar{j}_1 \cdots \bar{j}_p \\ \vdots \vdots \vdots \vdots \vdots \\ \bar{j}_{FP} \cdots \bar{j}_{FP} \end{bmatrix} = \begin{bmatrix} i_1^T \cdots i_p^T \\ i_F^T \cdots i_F^T \\ \bar{j}_1 \cdots \bar{j}_p \\ \bar{j}_{FP} \cdots \bar{j}_{FP} \end{bmatrix} \]

\[ \hat{R} = \hat{RQ} \]

\[ S = Q^{-1} \hat{S} \]

Orthogonal

\[ \hat{i}_j^T Q \hat{i}_f^T = 1 \]

\[ \hat{j}_j^T Q \hat{j}_f^T = 1 \]

\[ \hat{i}_j^T Q \hat{j}_f^T = 0 \]
Factorization: Algorithm

1. Make a measurement matrix $\tilde{W}$ from tracked feature points.
2. SVD $\tilde{W}$, then obtain $O_1, \Sigma, O_2$ ($\tilde{W} = O_1 \Sigma O_2$)
3. Extract $O_1', \Sigma', O_2'$ from $O_1, \Sigma, O_2$
4. Obtain $\hat{R} = O_1' [\Sigma']^{1/2}$, $\hat{S} = [\Sigma']^{1/2} O_2'$
5. Determine $Q$ from the orthogonal condition of $\hat{R}$
   $\hat{i}_f^T QQ^T \hat{i}_f = 1, \hat{j}_f^T QQ^T \hat{j}_f = 1, \hat{i}_f^T QQ^T \hat{j}_f = 0$
6. Determine $R$ and $S$ according to
   $R = \hat{R}Q$, $S = Q^{-1}\hat{S}$

Modeled Shape via Factorization

Input Image Sequence

3D Model

Rendered Virtual Views
Shape Modeling from Reflectance Analysis

What determines image intensity is

- Surface Material
- Surface Orientation
- Light Source Direction
- Viewpoint Direction

Diffuse reflectance surface (Lambertian surface)

$L = \rho \cos \theta = \rho (S \cdot n)$
Photometric Stereo

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

\[ L = \rho \cos \theta_s = \rho (S \cdot n) \]

\[
\begin{bmatrix}
L_1 \\
L_2 \\
L_3
\end{bmatrix} = \rho
\begin{bmatrix}
(S_1 \cdot n) \\
(S_2 \cdot n) \\
(S_3 \cdot n)
\end{bmatrix}
= \rho
\begin{bmatrix}
S_1^T \\
S_2^T \\
S_3^T
\end{bmatrix}
\begin{bmatrix}
n_x \\
n_y \\
n_z
\end{bmatrix}
\]

\[ L = \rho S n \]

\[ n = S^{-1} L \]

\[ \rho = |S^{-1} L| \]

Shape Recovery from Surface Normal $n$

Orthographic Projection Camera

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

Integration (Summation) of $n(x, y) \Rightarrow$ Shape $z(x, y)$

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

\[
n(x, y) = \frac{1}{\sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2 + 1}} \begin{bmatrix}
-\frac{\partial z}{\partial y} \\
\frac{\partial z}{\partial x} \\
1
\end{bmatrix}
\]
Example

3D Shape

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

\[ L = \rho \cos \theta_s \]
\[ = \rho (S \cdot n) \]

\[ \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_N \end{bmatrix} = \rho \begin{bmatrix} (S_1 \cdot n) \\ (S_2 \cdot n) \\ \vdots \\ (S_N \cdot n) \end{bmatrix} = \rho \begin{bmatrix} S_1^T \\ S_2^T \\ \vdots \\ S_N^T \end{bmatrix} n \]

\[ L = \rho S n \]
\[ n = \frac{S^{-1}L}{\rho} \]
\[ \rho = |S^{-1}L| \]

Shape Modeling from Reflectance Analysis

Surface normal
Body reflectance
Specular reflectance
Example of Intensity Curve

Phong’s Reflectance Model

\[ I = d(L \cdot n) + s(r \cdot v)^k \]

- \( d \): body reflectance
- \( s \): specular reflectance
- \( k \): specular sharpness
- \( n \): surface normal
- \( L \): light source direction
- \( v \): camera direction
- \( r = (2n \cdot L) / (|2n \cdot L|) \)
Fitting to body reflection

Rotation angle $\theta$

Fitting to all data

Fitting to selected data as body reflection

Fitting to specular reflection

Rotation angle $\theta$
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

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

Silhouette Intersection - Simple, but impossible to recover concaved shape
Example of Silhouette Intersection

Modeling and Rendering from multiple view videos
Parallel Stereo

Disparity: \( d = u - u' \)

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

Depth \( Z \) is inversely proportional to disparity \( d \).

Non-Parallel Stereo

\[
\begin{bmatrix}
P_{14} - uP_{14} \\
P_{24} - vP_{24} \\
P_{14}' - u'P_{14}' \\
P_{24}' - v'P_{24}'
\end{bmatrix} = \begin{bmatrix}
uP_{31} - P_{11} \\
vP_{32} - P_{12} \\
vP_{33} - P_{13} \\
vP_{34} - P_{14}
\end{bmatrix}
\begin{bmatrix}
P_{21} \\
P_{22} \\
P_{23} \\
P_{24}
\end{bmatrix}
\]

\[
\begin{bmatrix}
P_{31}' - P_{11}' \\
P_{32}' - P_{12}' \\
P_{33}' - P_{13}' \\
P_{34}' - P_{14}'
\end{bmatrix} = \begin{bmatrix}
vP_{31}' - P_{11}' \\
vP_{32}' - P_{12}' \\
vP_{33}' - P_{13}' \\
vP_{34}' - P_{14}'
\end{bmatrix}
\begin{bmatrix}
P_{21}' \\
P_{22}' \\
P_{23}' \\
P_{24}'
\end{bmatrix}
\]

\[b = BM\]
Epipolar Geometry

- Epipolar Lines
- Epipolar Plane
- 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)

Integrate all the SSD curve

Different base-line

Multiple Baseline Stereo: MBS
Multi-Baseline Stereo (MBS)

Stereo between camera $n$ and base camera

$$
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} =
\begin{bmatrix}
\frac{u_{Bn}}{d} \\
\frac{v_{Bn}}{d} \\
\frac{f_{Bn}}{d}
\end{bmatrix}
$$

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

Rectified multiple cameras

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

Depth image via 7 cameras
3D Complete Surface Modeling

Surface representation with implicit function:
\[ F(x, y, z) \begin{cases} > 0 & \text{outside} \\ = 0 & \text{on the surface} \\ < 0 & \text{inside} \end{cases} \]

Method for synthesizing range images

Range Image 1
5m x 5m x 3m
1 cm³ voxel
500 x 500 x 300

Range Image 2

Range Image 3
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
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.

Virtualized Reality : (CMU) [Vedula, Saito, Kanade et.al.98]
Example:

3-Man Basketball

(These are movies.)

Synthetic court

4D Model

Voxel Coloring/Space Curving

[Seitz and Dyer 97]

Checking Photo-Consistency of each voxel

If consistent, the voxel should be on the object surface

Viewing cameras

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

- 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

- Markers without known 3D positions
- Relative geometry among cameras (Fundamental Matrices)

View Interpolation via 3D Model in PGS

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

Moving by hand for tracking the object
Intermediate image sequence of 5 cameras

blue-c Project (ETH Zurich)[Gross03]
http://blue-c.ethz.ch/
Rendering via Pixel Transfer

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

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

\[
\begin{bmatrix}
  u \\ v
\end{bmatrix} = \begin{bmatrix} x \end{bmatrix} f \\
\]

Pixel transfer
Pixel position mapping

Affine Transform

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

\(u = Rx + t\)

In the homogeneous coordinate representation

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

\(\tilde{u} = H\tilde{x}\)
Homography

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

\[\tilde{u} = H\tilde{x}\]

If \(g=h=0\) then equal to affine transform

\[
\begin{align*}
  u &= \frac{ax + by + c}{gx + hy + 1} \\
  v &= \frac{dx + ey + f}{gx + hy + 1}
\end{align*}
\]

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

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

![Diagram of soccer field and camera views]

- cam 1
- cam 2
- cam 3
- cam 4

**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 information
      - HSI transform
    - Geometric information
      - Homography transform

---

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](image1)

![Homography of the ground Plane](image2)
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
— like “The Matrix” —

![Virtual Viewpoint Visual Effect](image)
Intermediate Viewpoint Images

[Images of intermediate viewpoint images with annotations]

Intermediate Viewpoint Images (Magnified)

[Images of magnified intermediate viewpoint images with annotations]
Comparison

Virtual Camera (Cam 2-4 Weight 5:5)  Real Camera (Cam 3)

Input Video  Synthesized Video
At Player’s Viewpoint
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


• 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

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

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

Result 2

Frame 226

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

Stadium Camera 4  
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

HMD

Overlay Image

PA GE 1

PA GE 2

PC for Left Image

PC for Right Image

Observer

Image Processing

Notebook

Marker

2D Matrix Code

Method for Texture Overlay

HMD image

Texture

Recognition of 2D matrix code

Detection of markers

Texture Overlay

Overlay Image

Page number: 2
Book ID: 0

Page number: 2
Book ID: 0

Transformation Matrix

Overlay

Texture

Overlay image

HMD image

Corresponding points

Load

Deform
Implementation

Input images : 640 × 480 pixels

HMD image

Appearance of experiment

Texture images : 350 × 500 pixels


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.

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