Camera Parameters and Calibration

Camera parameters

- From last time….

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix}$$

Homogeneous Coordinates (Again)

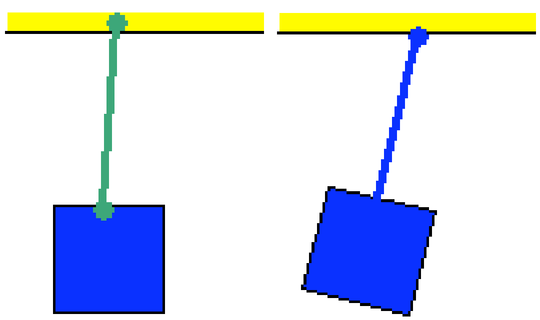
$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

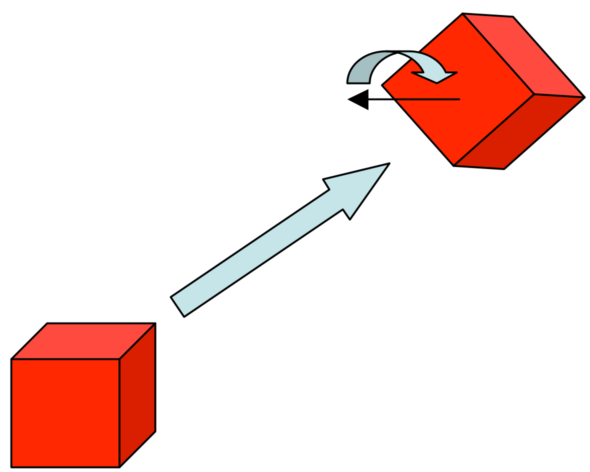
$$(U, V, W) \rightarrow \left(\frac{U}{W}, \frac{V}{W}\right) = (u, v)$$

Extrinsic Parameters: Characterizing Camera position

Chasles's theorem:

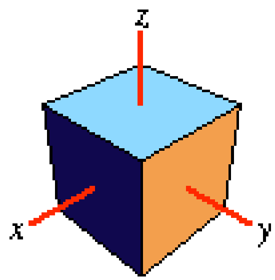
Any motion of a solid body can be composed of a translation and a rotation.



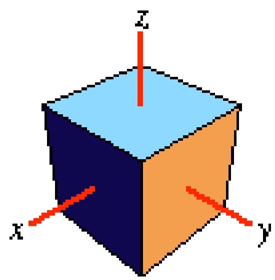


3D Rotation Matrices

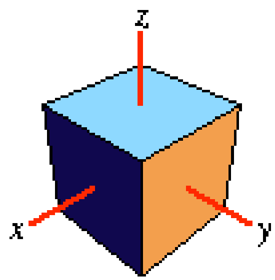
$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$$



$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix}$$



$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

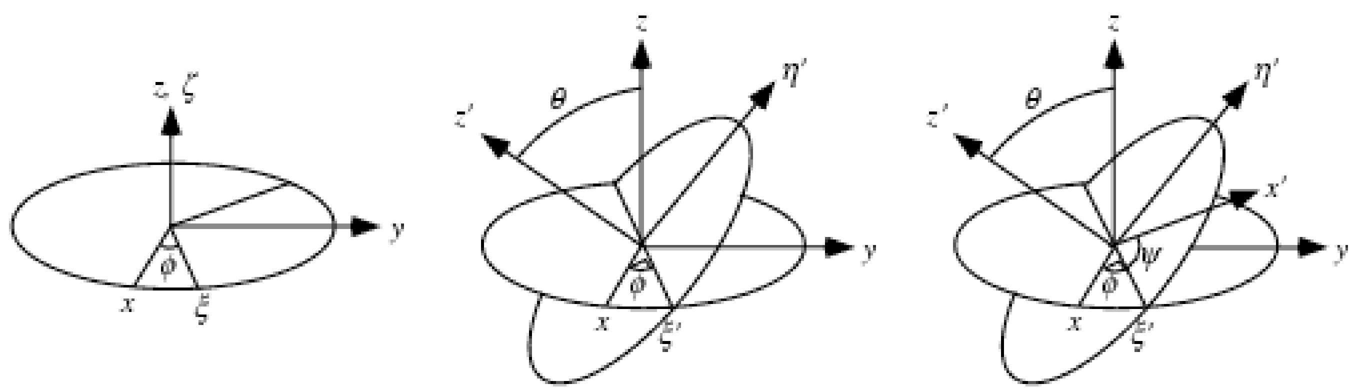


3D Rotation Matrices cont'd

Euler's Theorem: An arbitrary rotation can be described by 3 rotation parameters

For example: $R = R_x(\alpha)\,R_y(\beta)\,R_z(\gamma)$

More Generally:



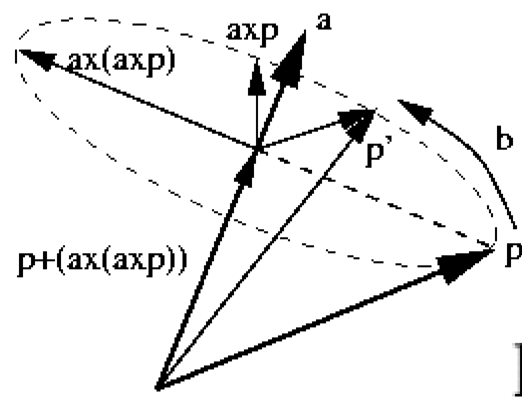
Most General:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Rodrigue’s Formula

Take any rotation axis *a* and angle θ: What is the matrix?

$$\mathbf{p}'(t) = \mathbf{a} \times \mathbf{p}(t)$$



$$\mathbf{a} \times \mathbf{v} = \begin{bmatrix} a_y v_z - a_z v_y \\ a_z v_x - a_x v_z \\ a_x v_y - a_y v_x \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

$$= \mathbf{A}\mathbf{v}$$

with $\mathbf{A} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$

$$\mathbf{p}'(t) = \mathbf{A}\mathbf{p}(t)$$

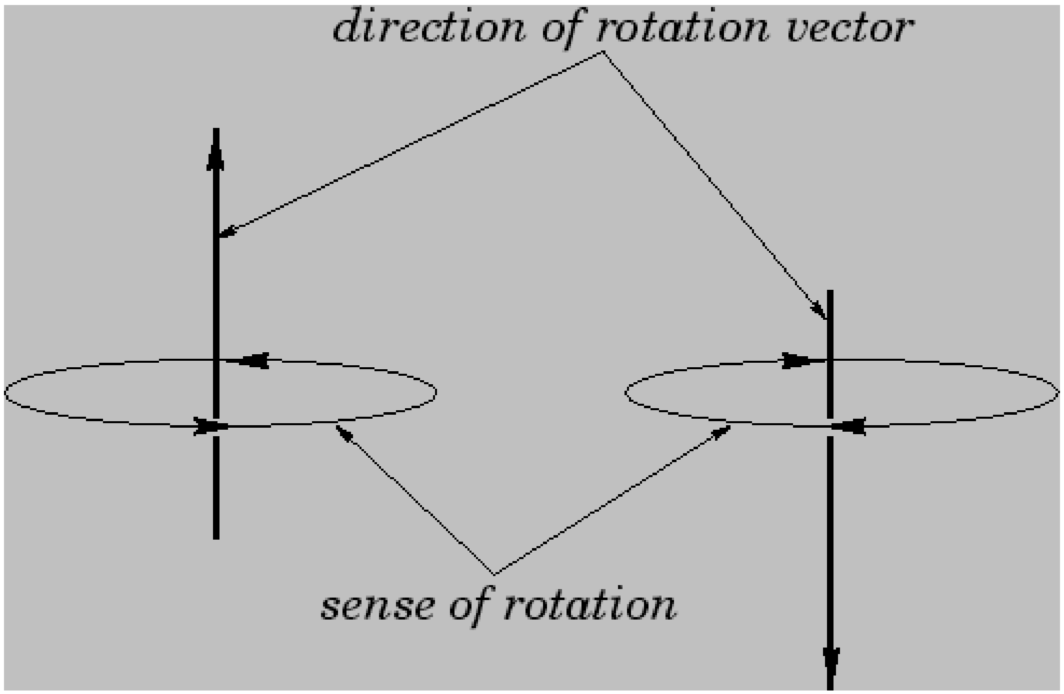
$$\mathbf{p}(t) = e^{\mathbf{A}t}\mathbf{p}(0)$$

$$\mathbf{p}(\theta) = e^{\mathbf{A}\theta}\mathbf{p}(0)$$

$$e^{\mathbf{A}\theta} = I + \mathbf{A}\sin\theta + \mathbf{A}^2\left[1 - \cos\theta\right]$$

$$R = e^{\mathbf{A}\theta}$$

Rotations can be represented as points in space (with care)

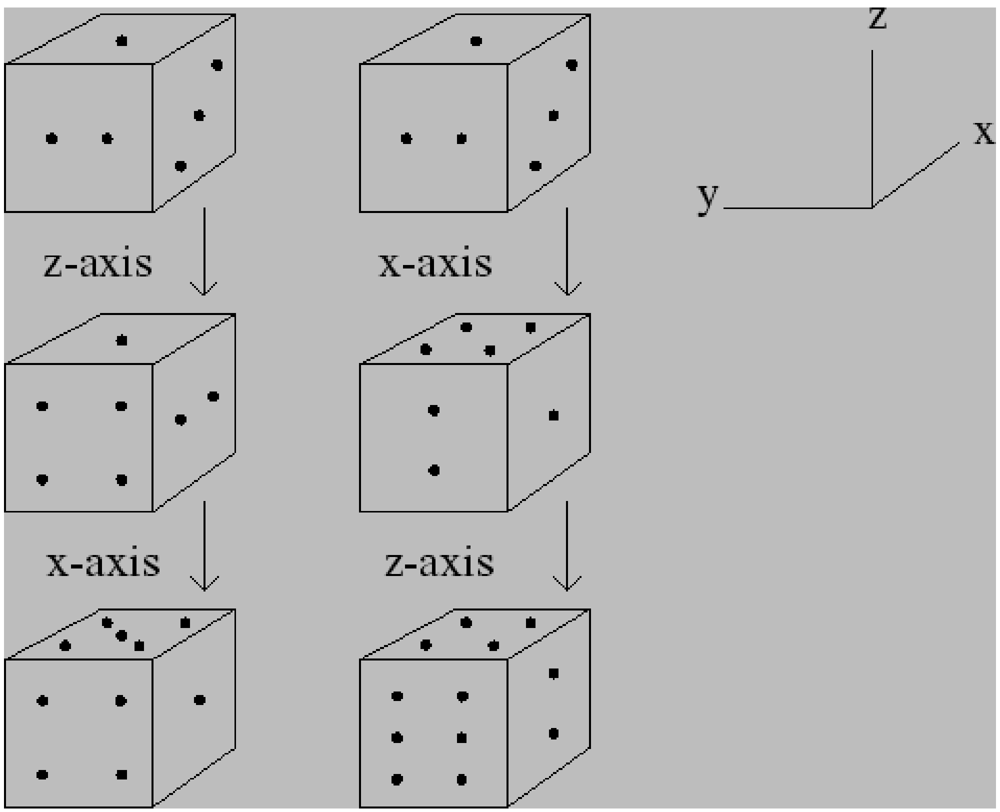


Turn vector length into angle, direction into axis:

Useful for generating random rotations, understanding angular errors, characterizing angular position, etc.

Problem: not unique
Not commutative

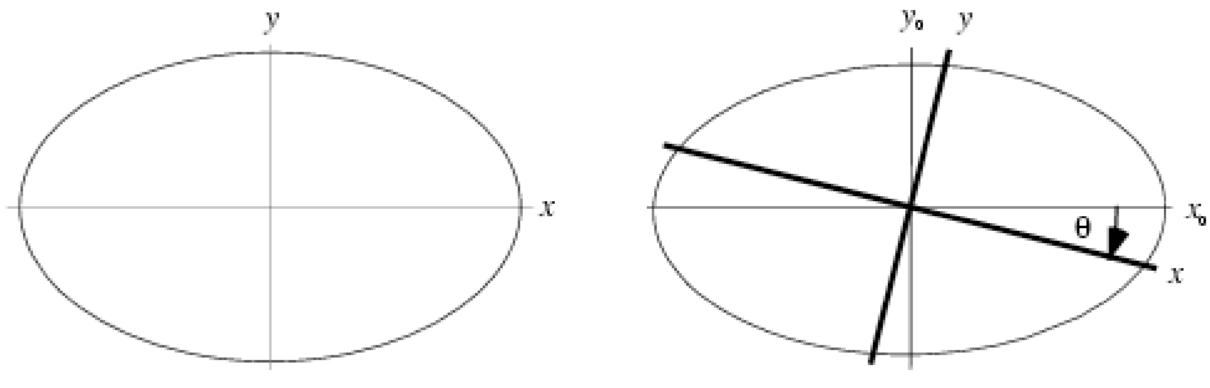
Other Properties of rotations



NOT Commutative

R1*R2 ≠ R2*R1

Rotations



axes rotated by angle θ

To form a rotation matrix, you can plug in the columns of new coordinate points

For Example:
The unit x-vector goes to *x'*:

$$x' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Other Properties of Rotations

Inverse: $R^{-1} = R^T$

rows, columns are othonormal

$r_i^T r_j = 0$ if $i \neq j$, else $r_i^T r_i = 1$

Determinant:

det(R) = 1

The effect of a coordinate rotation on a function:

$x' = R x$

$F(x') = F(R^{-1} x)$

Extrinsic Parameters

$p' = \mathrm{R}\,p + t$

R = rotation matrix

t = translation vector

In Homogeneous coordinates,

$p' = \mathrm{R}\,p + t$ =>

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & ty \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Intrinsic Parameters

Differential Scaling

$$\begin{bmatrix} u * Su \\ v * Sv \\ 1 \end{bmatrix} = \begin{bmatrix} Su & 0 & 0 \\ 0 & Sv & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Camera Origin Offset

$$\begin{bmatrix} u + u_0 \\ v + v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

2D Example

- Rotation –90°

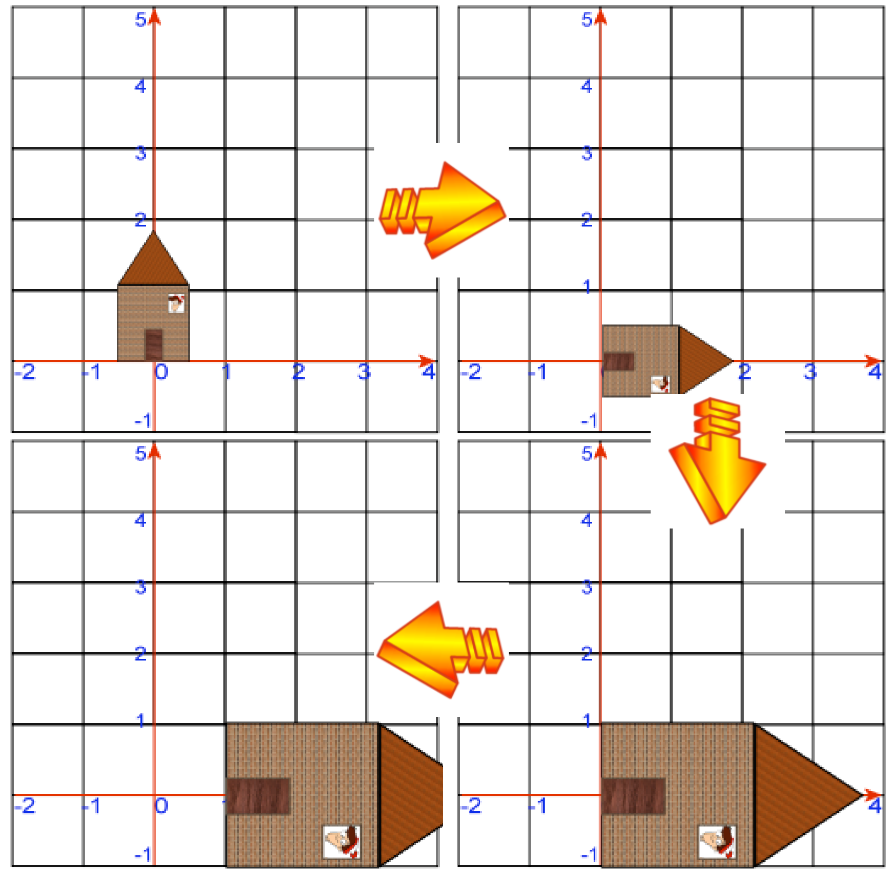
$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Scaling *2

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Translation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$



House Points

$$p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The Whole (Linear) Transformation

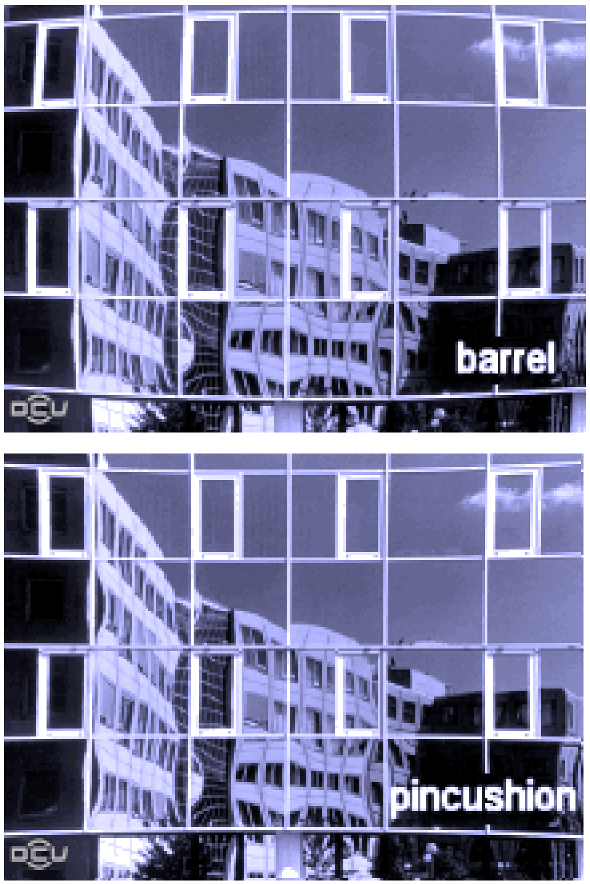
$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix}$$

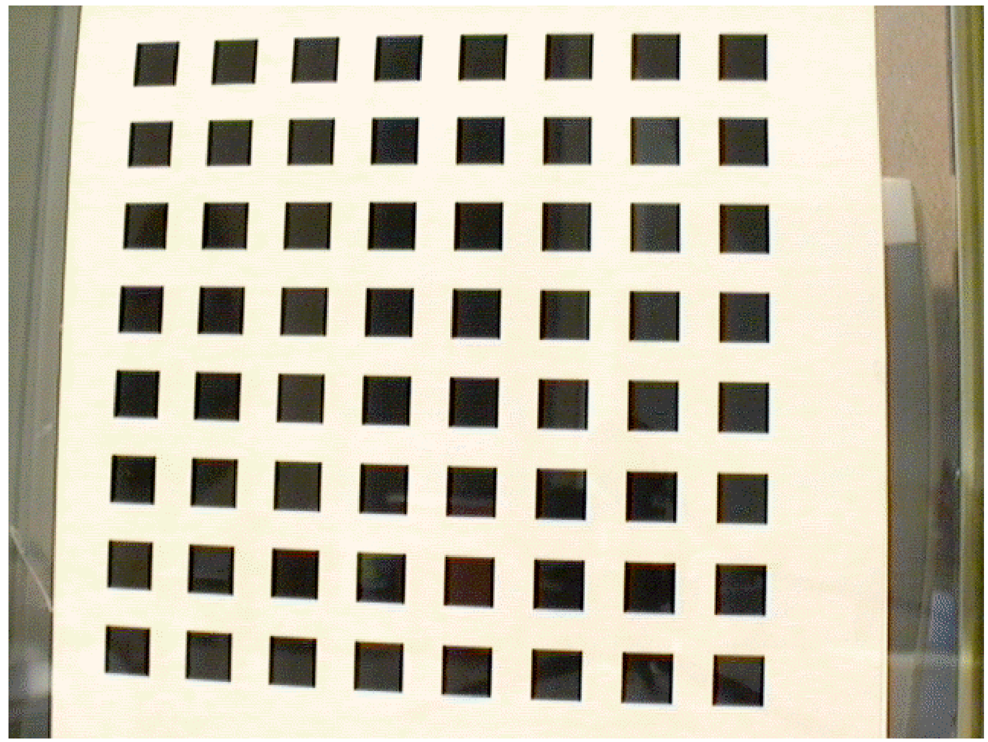
$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} -f \cdot s_u & 0 & u_0 \\ 0 & -f \cdot s_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ T \end{bmatrix}$$

Final image coordinates

$$u = U/W$$
$$v = U/W$$

Non-linear distortions (not handled by our treatment)



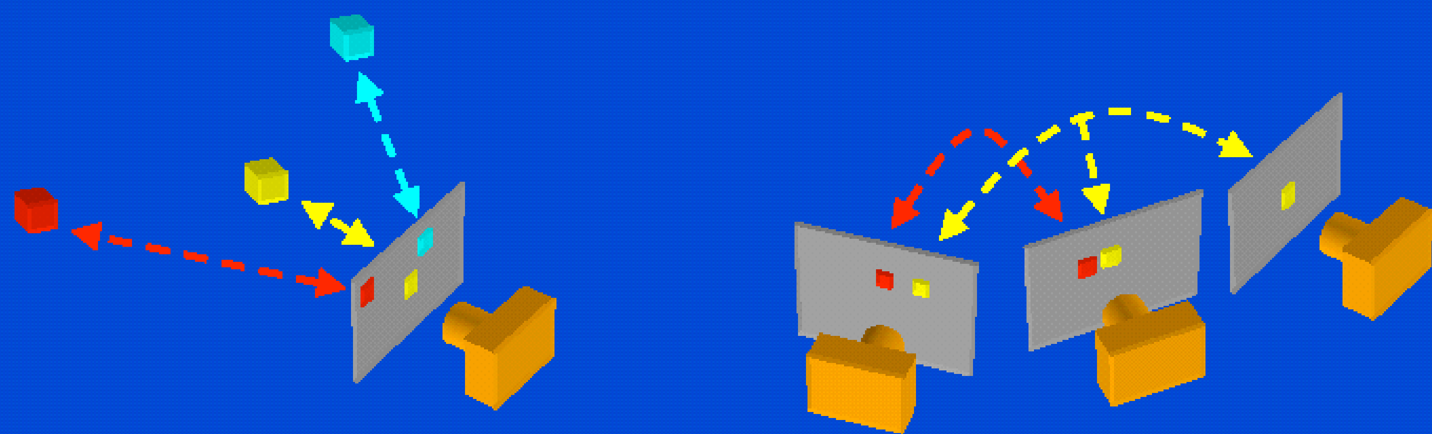


Camera Calibration

- You need to know something beyond what you get out of the camera
  - Known World points (object)
    - Traditional camera calibration
    - Linear and non-linear parameter estimation
  - Known camera motion
    - Camera attached to robot
    - Execute several different motions
  - Multiple Cameras

Augmented pin-hole camera model

- Focal point, orientation
- Focal length, aspect ratio, center, lens distortion

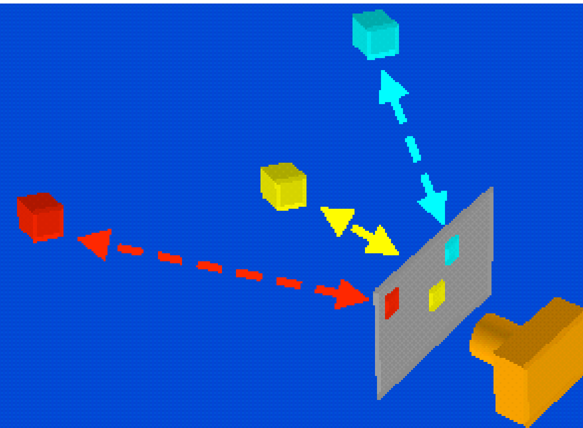


2D ⇔ 3D correspondence	2D ⇔ 2D correspondence
"Classical" calibration	SFM, "Self-calibration"

Classical Calibration

Know 3D coords, 2D coords

- **Find projection matrix Π**



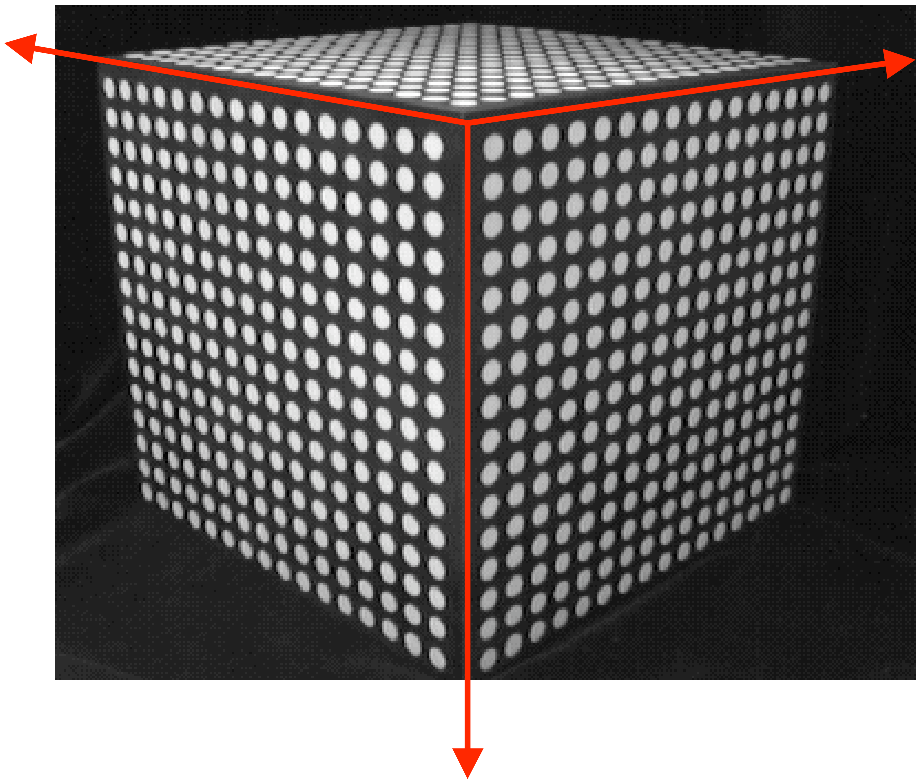
$$d\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{u} = \Pi \quad \mathbf{X}$$

11 unknowns (up to scale)
2 equations per point
(eliminate d)

6 points is sufficient

Camera Calibration



Take a known set of points.

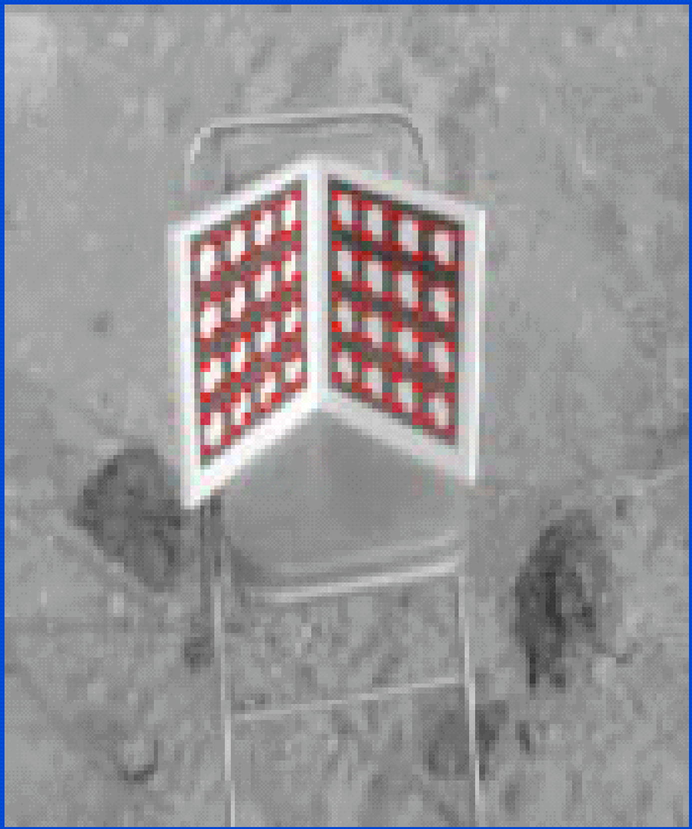
Typically 3 orthogonal planes.

Treat a point in the object as the World origin

Points x1, x2, x3,

Project to y1,y2,y3

Calibration Patterns



Calibration grid
Z. Zhang, Microsoft Research

Chromaglyphs
Bruce Culbertson, HP-labs

Classical Calibration

Put the set of points known object points x_i into a matrix

$$X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$$

And projected points u_i into a matrix

$$U = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}$$

Note this is only for instructional purposes. It does not work as a real procedure.

<u>Solve for the transformation matrix:</u>

Odd derivation of the Least Squares solution:

$$U = \square X$$

$$UX^t = \square (XX^t)$$

$$UX^t(XX^t)^{-1} = \square (XX^t)(XX^t)^{-1}$$

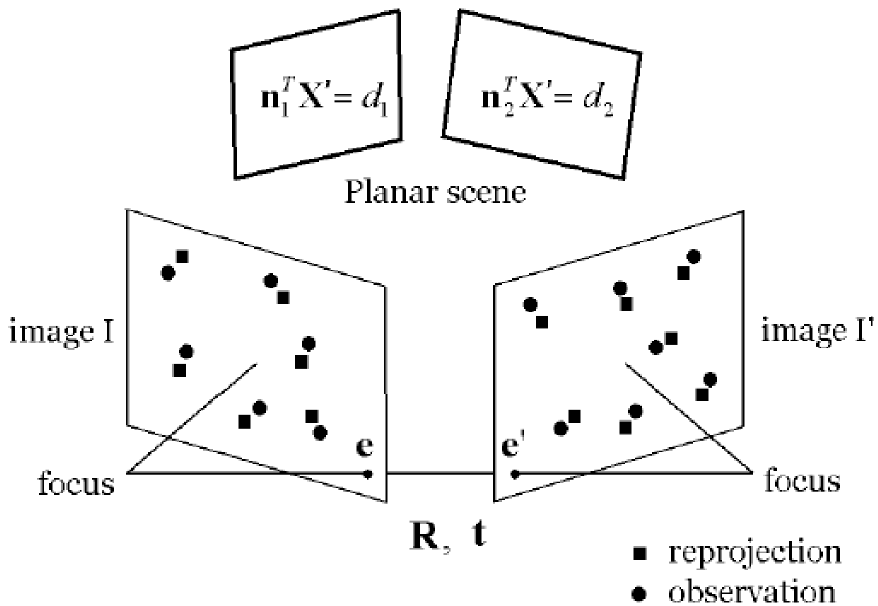
$$UX^t(XX^t)^{-1} = \square$$

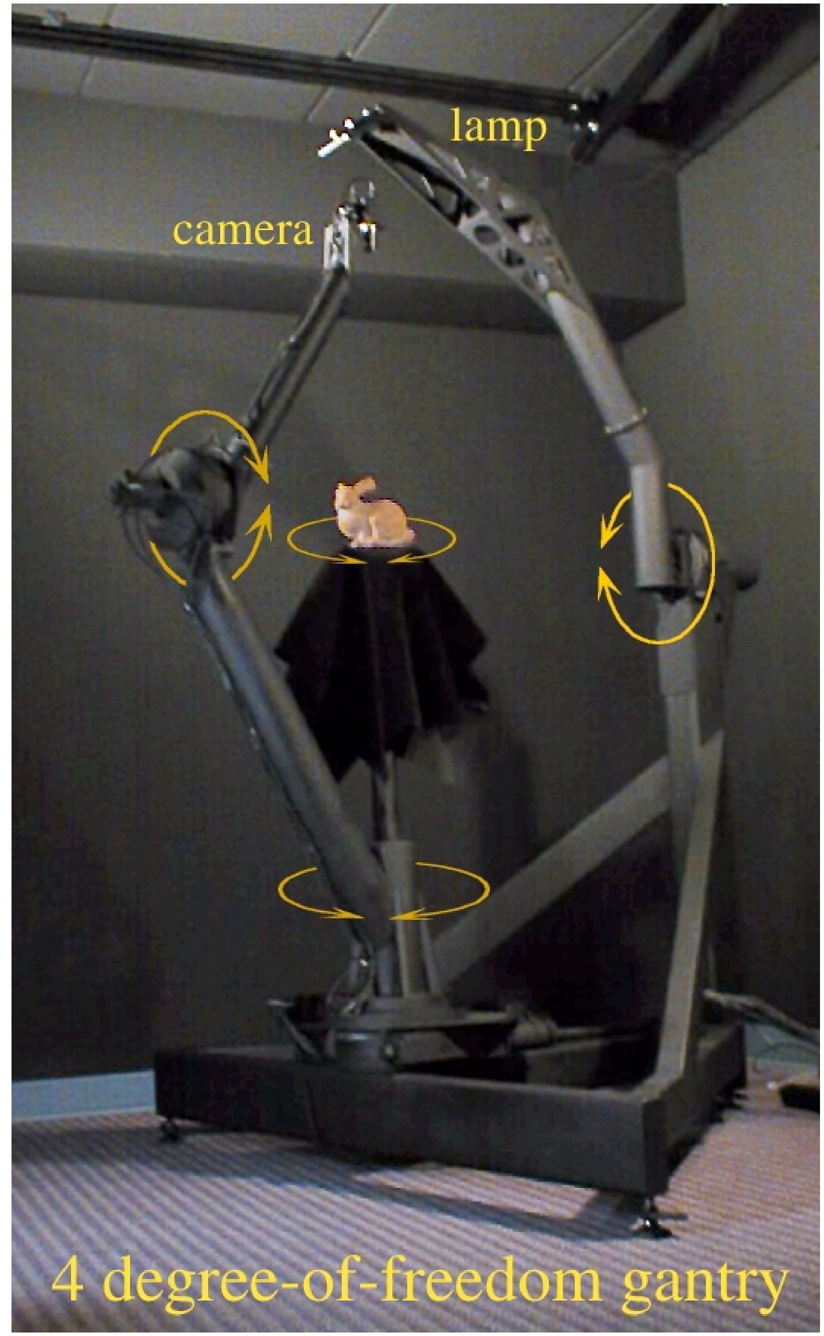
Next extract extrinsic and intrinsic parameters from $\square$

Real Calibration

- Real calibration procedures look quite different
  - Weight points by correspondence quality
  - Nonlinear optimization
  - Linearizations
  - Non-linear distortions
  - Etc.

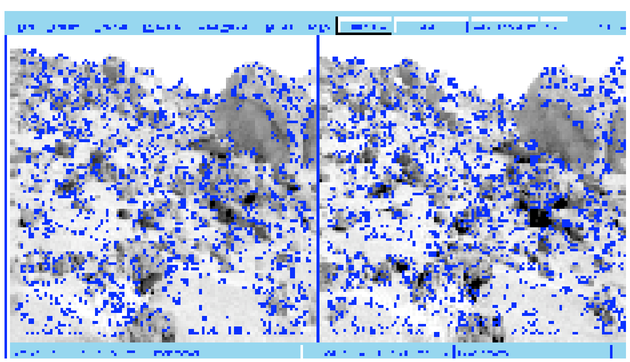
Camera Motion

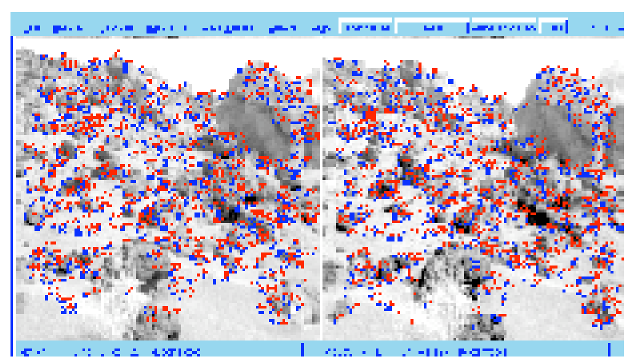




Calibration Example (Zhang, Microsoft)

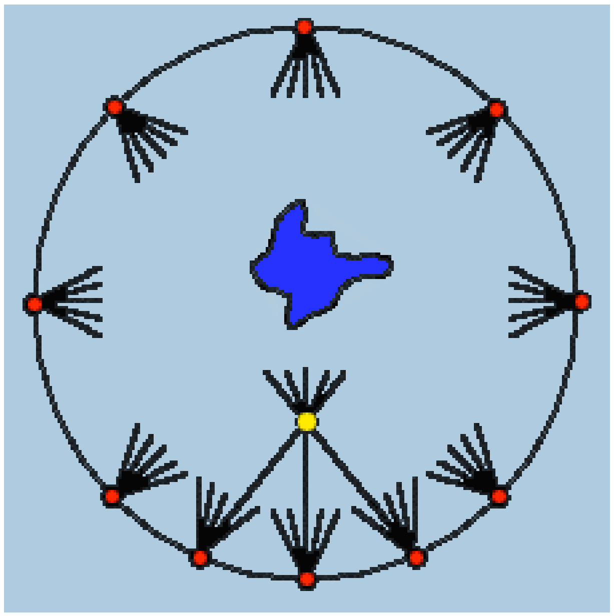
8 Point matches manually picked
Motion algorithm used to calibrate camera

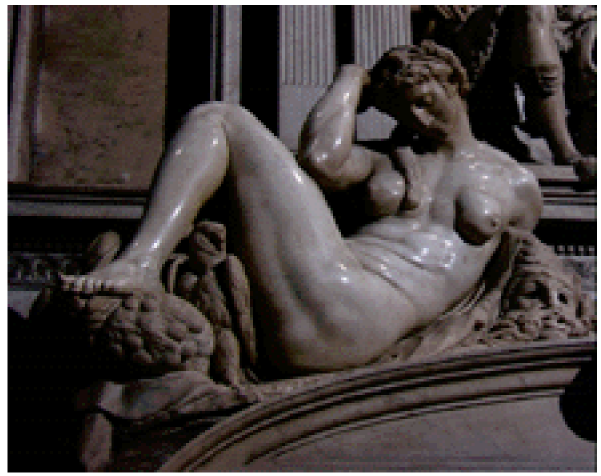


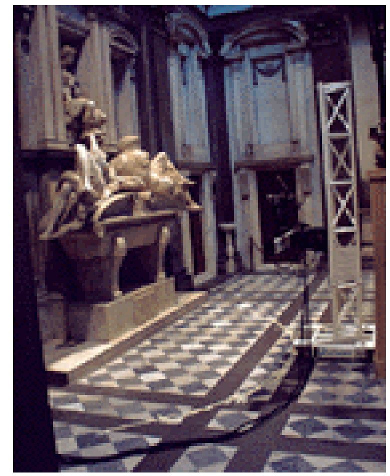


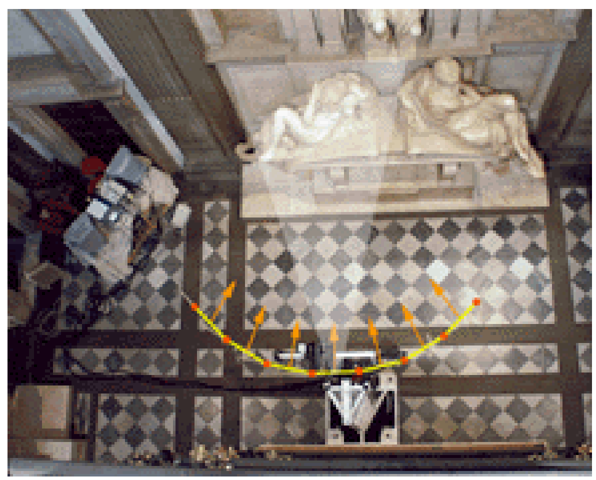
Applications

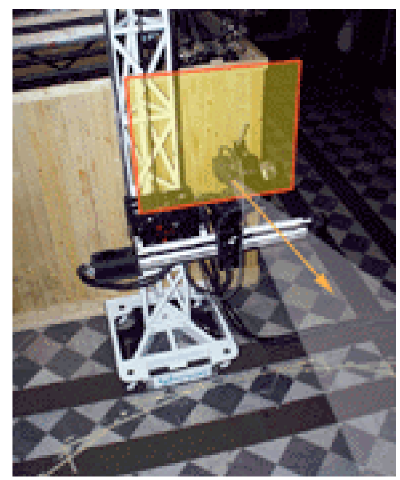
Image based rendering: Light field -- Hanrahan (Stanford)



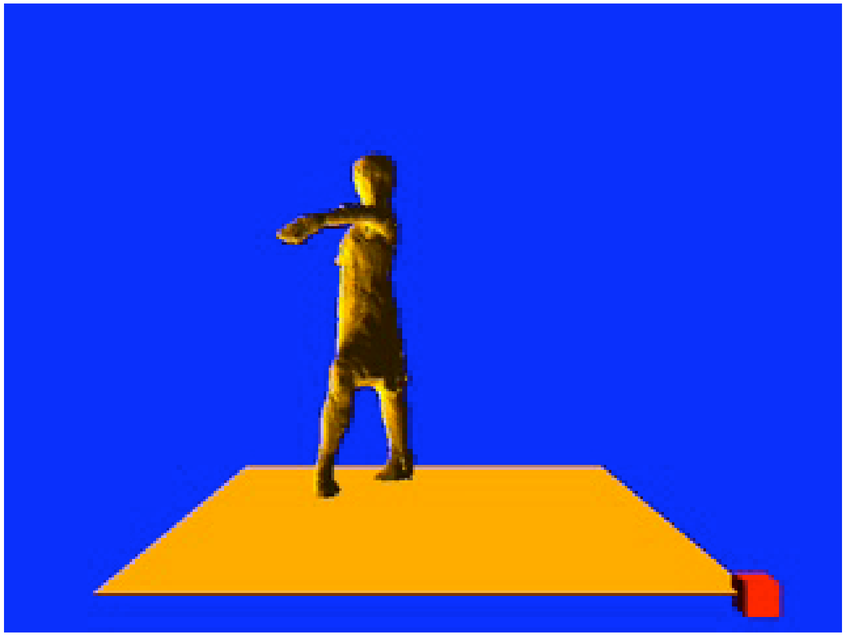




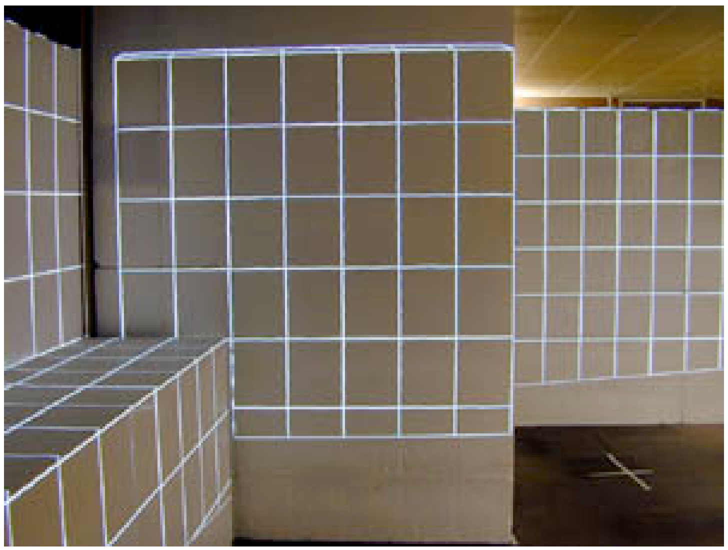


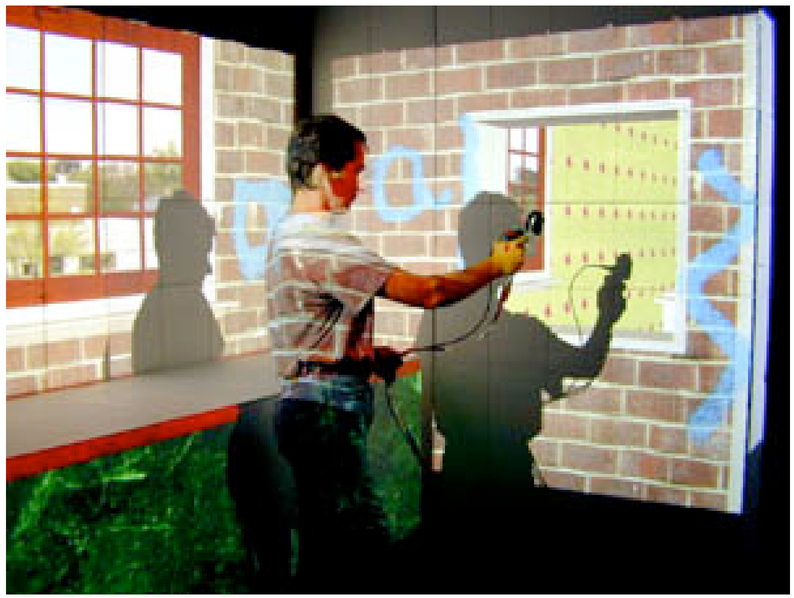


Virtualized Reality



Projector-based VR
UNC Chapel Hill





Shader Lamps

Figure 1. An enthusiastic young user (Miriam Mintzer Fuchs, age 9 years) demonstrates our easy to use system for 3D painting on movable objects. Note that the color palette on the table and the color on the spherical tip of the "paint brush" are projected.

Shape recovery without calibration
Fitzgibbons, Zisserman

Fully automatic procedure: Converting the images to 3D models is a black-box filter: Video in, VRML out.

* **We don't require that the motion be regular**: the angle between views can vary, and it doesn't have to be known. Recovery of the angle is automatic, and accuracy is about 40 millidegrees standard deviation. In golfing terms, that's an even chance of a hole in one.

* **We don't use any calibration targets**: features on the objects themselves are used to determine where the camera is, relative to the turntable. Aside from being easier, this means that there is no problem with the setup changing between calibration and acquisition, and that anyone can use the software without special equipment.

For example, this dinosaur sequence was supplied to us by the University of Hannover without any other information. (Actually, we do have the ground-truth angles so that we can make the accuracy claims above, but of course these are not used in the reconstruction).