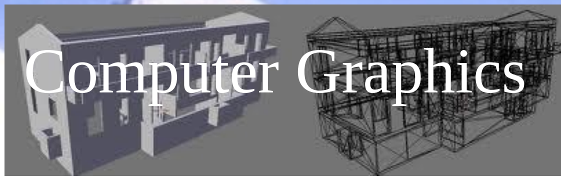
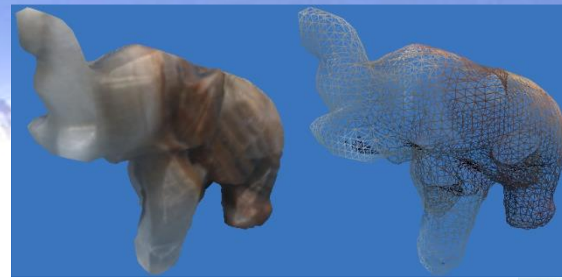
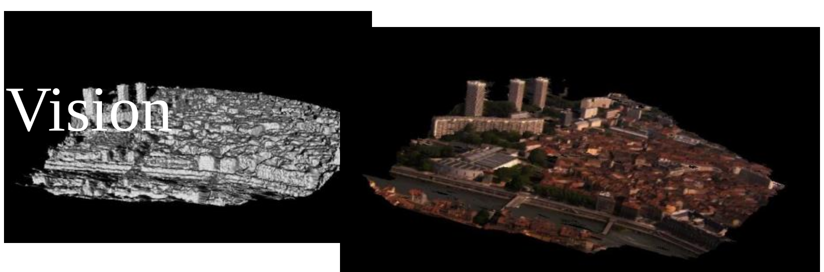
Geometric Computer Vision
Review and Outlook

What is Computer Vision?

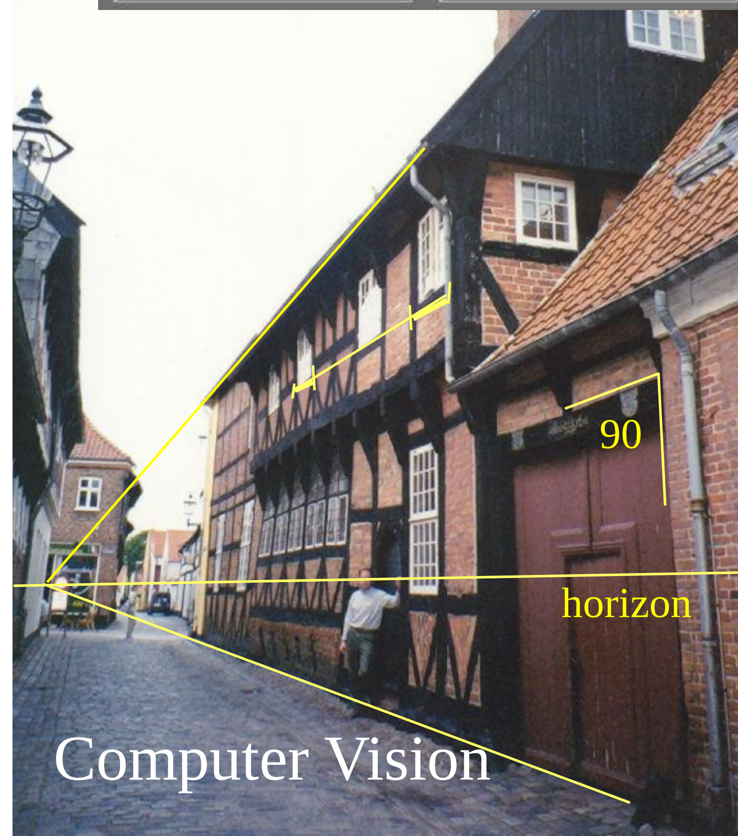




- Three Related fields
  - Image Processing: Changes 2D images into other 2D images
  - Computer Graphics: Takes 3D models, renders 2D images
  - Computer vision: Extracts scene information from 2D images and video
    - e.g. Geometry, "Where" something is in 3D,
    - Objects "What" something is"
- What information is in a 2D image?
- What information do we need for 3D analysis?

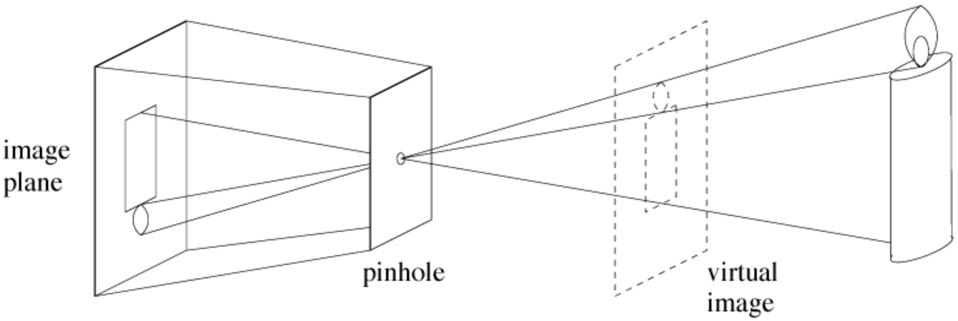


Only special cases can be solved.



The equation of projection

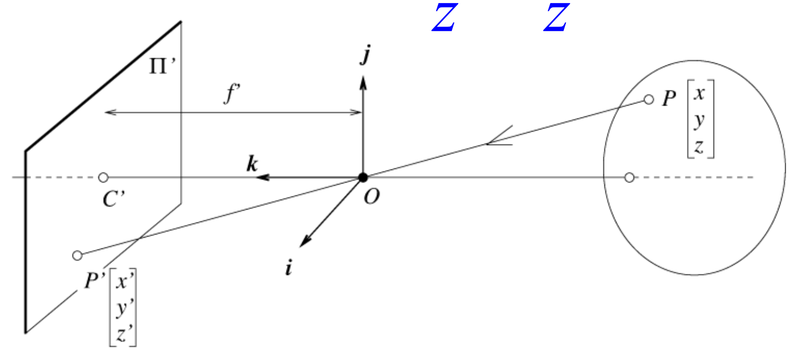
Intuitively:



Mathematically:

- Cartesian coordinates:

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$



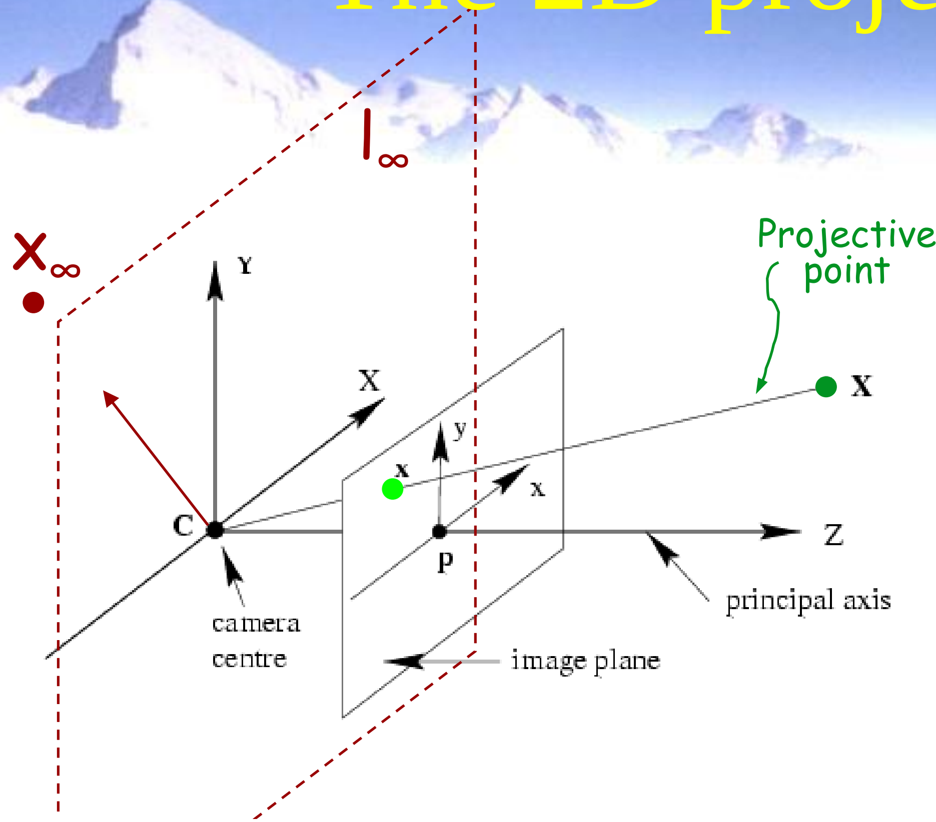
Projectively:

$\mathbf{x} = P\mathbf{X} \quad P : 3 \times 4$ Projection matrix

$\mathbf{x} = K[R \quad \mathbf{t}]\mathbf{X} \quad K : 3 \times 3$ Camera matrix (internal params)

$R, \mathbf{t}$ Rotation, translation (ext. params)

The 2D projective plane



Homogeneous coordinates

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \equiv s \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad s \neq 0$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

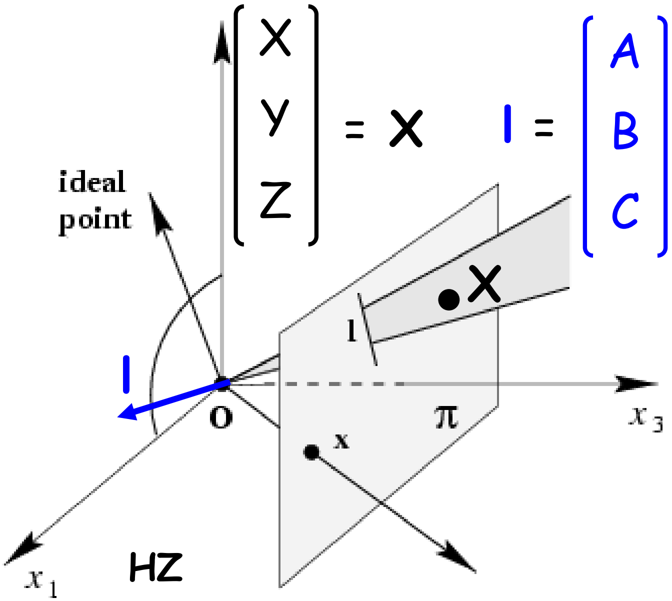
Inhomogeneous equivalent

- Perspective imaging models 2d projective space
- Each 3D ray is a point in P^2 : homogeneous coords.
- Ideal points
- P^2 is R^2 plus a "line at infinity" l_∞

$$X_\infty = \begin{bmatrix} X \\ Y \\ 0 \end{bmatrix}$$

Lines

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{x} \qquad \mathbf{l} = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$



- Projective line ~ a plane through the origin

$$\mathbf{l}^T\mathbf{x} = \mathbf{x}^T\mathbf{l} = AX + BY + CZ = 0$$

$$\mathbf{x}_\infty = \begin{bmatrix} X \\ Y \\ 0 \end{bmatrix} \qquad \mathbf{l}_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

"line at infinity"

- Ideal line ~ the plane parallel to the image

Duality: For any 2d projective property, a dual property holds when the role of points and lines are interchanged.

$$\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$$

The line joining two points

$$\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$$

The point joining two lines

Projective transformations

- Homographies, collineations, projectivities
- 3x3 nonsingular H

  maps $\mathbf{P}^2$ to $\mathbf{P}^2$
  8 degrees of freedom
  determined by 4 corresponding points

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

$$x' = Hx$$

- Transforming Lines?

subspaces preserved $\mathbf{x}^T\mathbf{l} = 0 \qquad \mathbf{x}'^T\mathbf{l}' = 0$

substitution $x^T H^T \mathbf{l}' = 0$

dual transformation $\mathbf{l}' = H^{-T}\mathbf{l}$

Planar Projective Warping

HZ

$$x_i' = Hx_i$$

$$i = 1 \ldots 4$$

A novel view rendered via four points with known structure

Geometric strata: 2d overview

Group	Transformation	Invariants	Distortion
Projective 8 DOF	$H_P = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix}$	• Cross ratio • Intersection • Tangency	
Affine 6 DOF	$H_A = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	• Parallelism • Relative dist in 1d • Line at infinity $\mathbf{l}_\infty$	
Metric 4 DOF	$H_S = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	• Relative distances • Angles • Dual conic C^*_∞	
Euclidean 3 DOF	$H_E = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	• Lengths • Areas	

2 dof $\mathbf{l}_\infty$

2 dof C^*_∞

Projective 3D space

- Points $\mathbf{X}=(x_1,x_2,x_3,x_4), \quad x_4 \neq 0$

  $(x_1,x_2,x_3,0)$

  $(x_1,x_2,x_3,x_4)\rightarrow(x_1/x_4,x_2/x_4,x_3/x_4)$

  $(X,Y,Z,1)\leftarrow(X,Y,Z)$

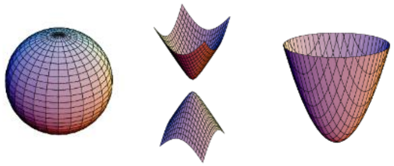
- Planes $\boldsymbol{\pi}=(\pi_1,\pi_2,\pi_3,\pi_4)$

  $\boldsymbol{\pi}^T\mathbf{X}=0$

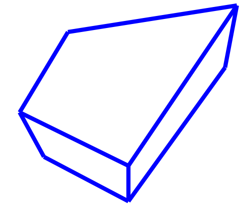
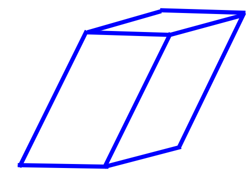
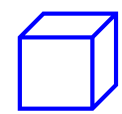
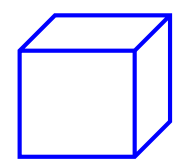
Points and planes are dual in 3d projective space.

- Lines: 5DOF, various parameterizations
- Projective transformation:
  - 4x4 nonsingular matrix H
  - Point transformation $\mathbf{X}'=H\mathbf{X}$
  - Plane transformation $\boldsymbol{\pi}'=H^{-T}\boldsymbol{\pi}$
- Quadrics: Q $\mathbf{X}^TQ\mathbf{X}=0$
  - 4x4 symmetric matrix Q
  - 9 DOF (defined by 9 points in general pose)

Dual: Q* $\boldsymbol{\pi}^TQ^*\boldsymbol{\pi}=0$



Geometric strata: 3d overview

Group	Transformation	Invariants	Distortion
Projective 15 DOF	$H_P = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix}$	• Cross ratio • Intersection • Tangency	
Affine 12 DOF	$H_A = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	• Parallelism • Relative dist in 1d • Plane at infinity $\boldsymbol{\pi}_\infty$	
Metric 7 DOF	$H_S = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	• Relative distances • Angles • Absolute conic Ω_∞	
Euclidean 6 DOF	$H_E = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	• Lengths • Areas • Volumes	

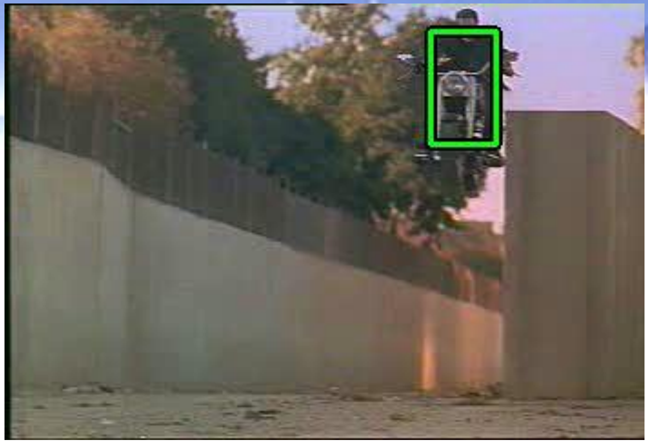
3DOF $\boldsymbol{\pi}_\infty$

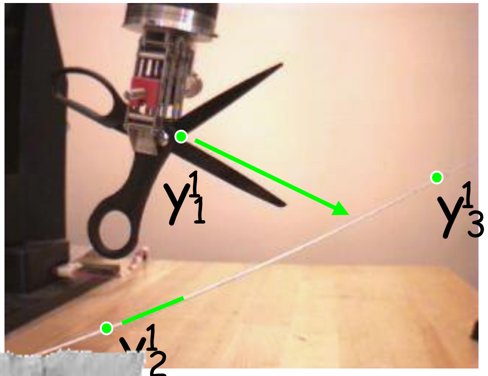
5DOF Ω_∞

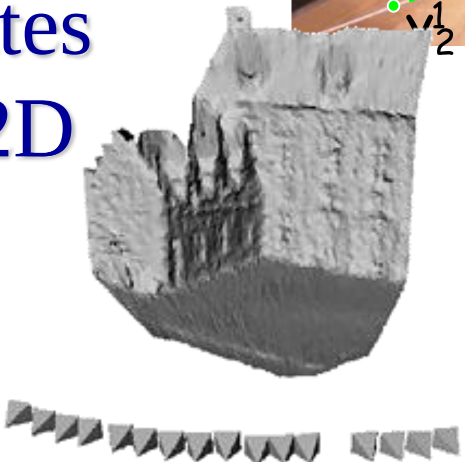
Faugeras '95

Tasks in Geometric Vision

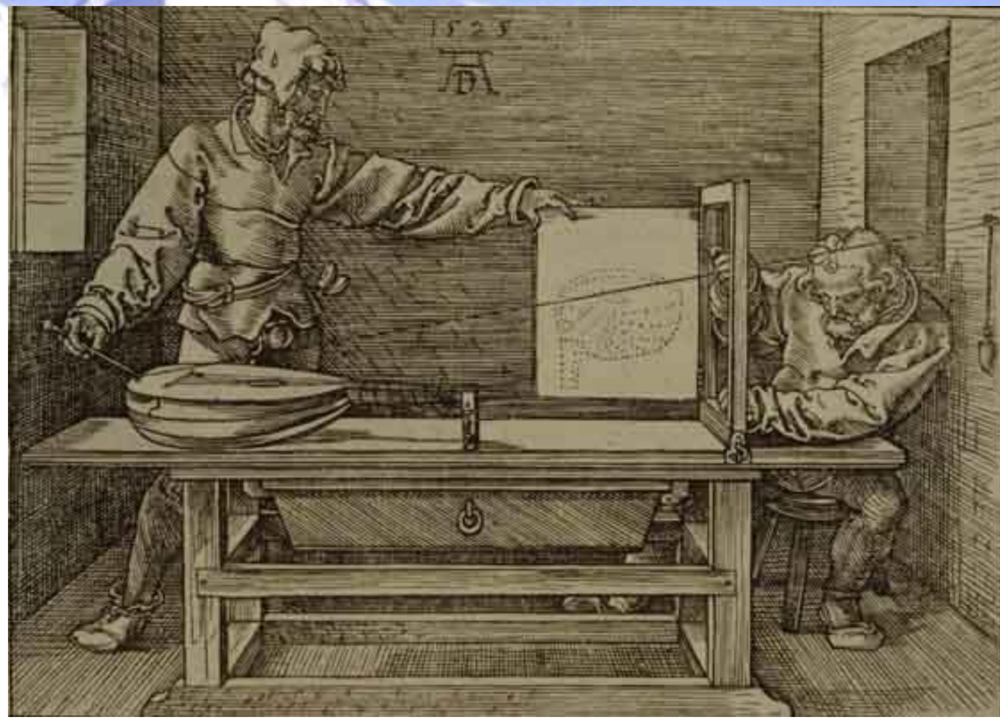
- **Tracking:** Often formulated in 2D, but tracks 3D objects/scene
- **Visual control:** Explicit 2D geometric constraints implicitly fulfills a 3D motion alignment
- **3D reconstruction**: Computes an explicit 3D model from 2D images



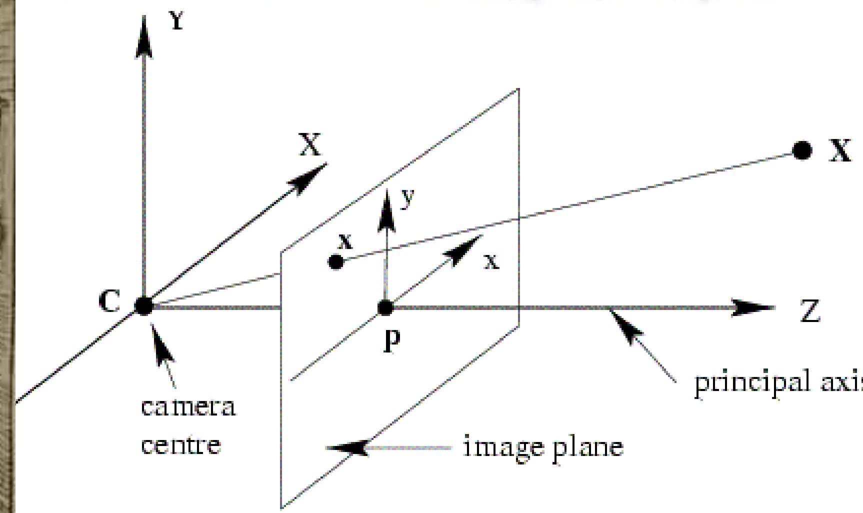




Camera model: Relates 2D – 3D



[Dürer]



$\mathbf{x} = P\mathbf{X} \quad P : 3 \times 4$ **Projection matrix**

$\mathbf{x} = K[R \quad \mathbf{t}]\mathbf{X} \quad K : 3 \times 3$ **Camera matrix (internal params)**

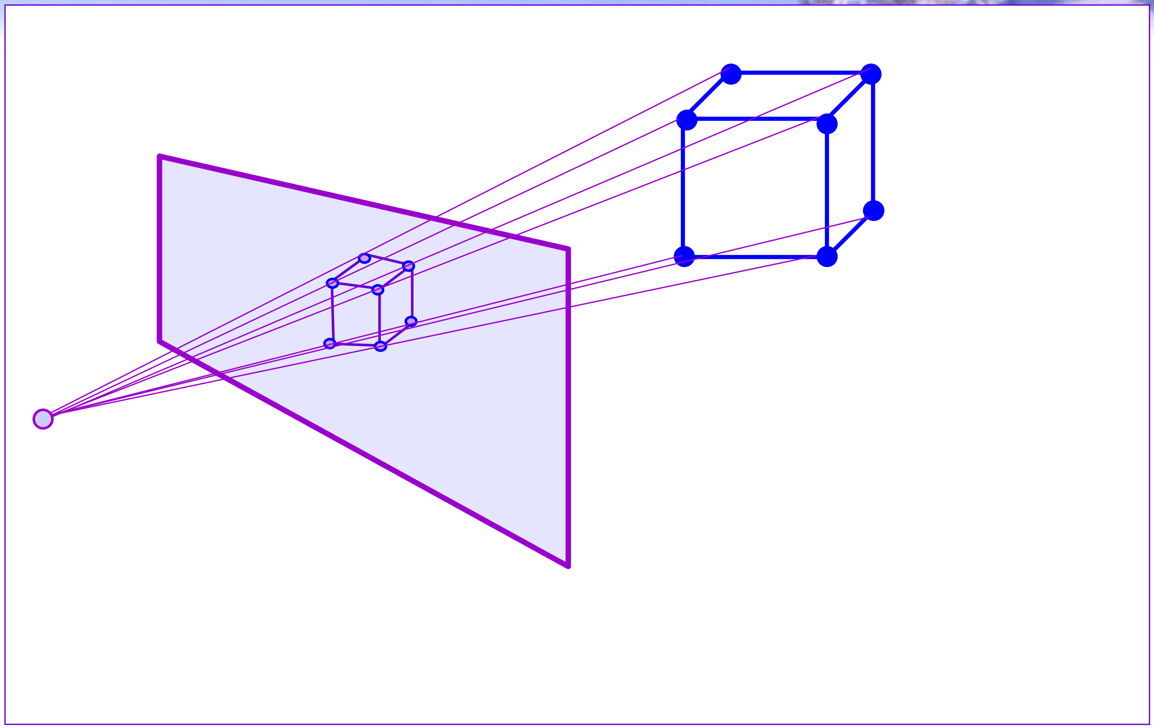
$R, \mathbf{t}$ **Rotation, translation (ext. params)**

2D-3D geometry - resection

- Projection equation

  $x_i = P_i X$

- Resection:
  - $x_i, X \longrightarrow P_i$



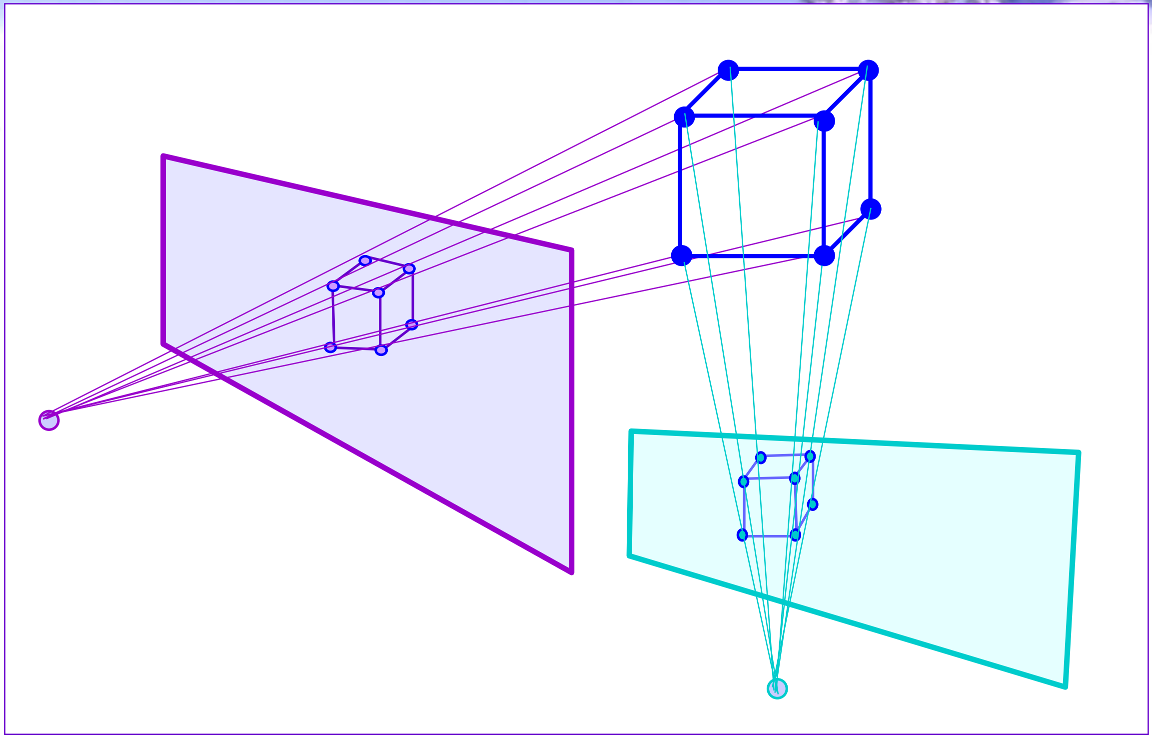
Given image points and 3D points calculate camera projection matrix.

2D-3D geometry - intersection

- Projection equation

  $x_i = P_i X$

- Intersection:
  - $x_i, P_i \longrightarrow X$



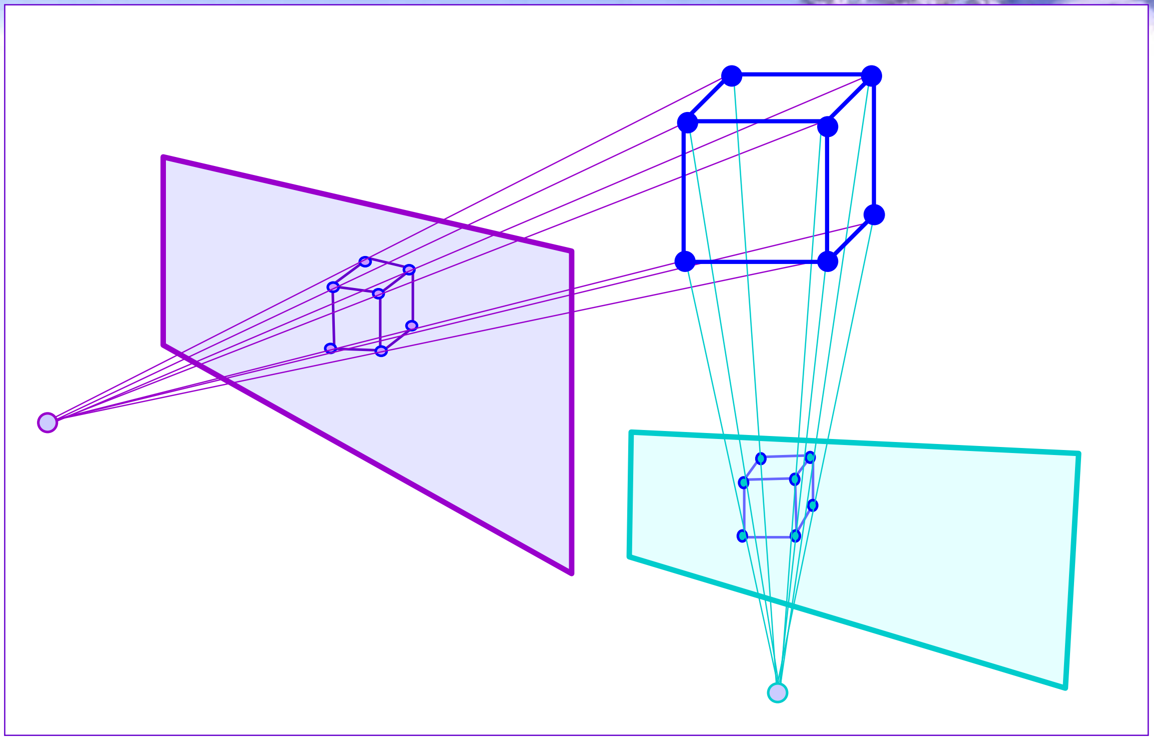
Given image points and camera projections in at least 2 views calculate the 3D points (structure)

2D-3D geometry - SFM

- Projection equation

  $x_i = P_i X$

- Structure from motion (SFM)
  - $x_i \longrightarrow P_i, X$



Given image points in at least 2 views calculate the 3D points (structure) and camera projection matrices (motion)

- Estimate projective structure
- Rectify the reconstruction to metric (autocalibration)

Reconstructing scenes

‘Small’ scenes (one, few buildings)

- SFM + multi view stereo
- man made scenes: prior on architectural elements
- interactive systems

City scenes (several streets, large area)

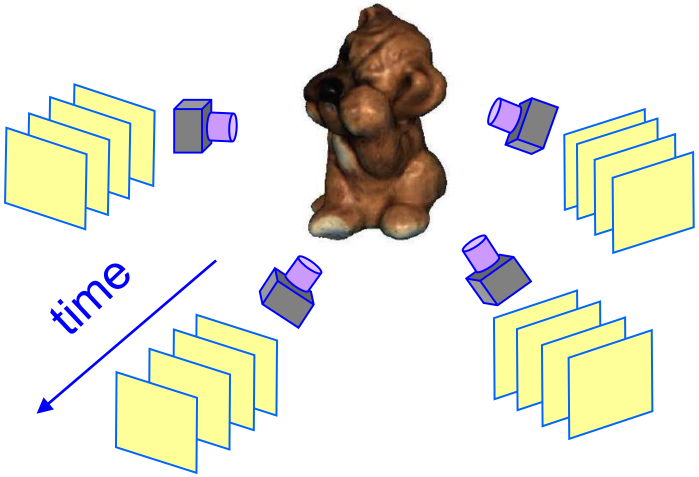
- aerial images
- ground plane, multi cameras

SFM + stereo [+ GPS]

depth map fusions

Large scale (city) modeling

Modeling dynamic scenes



Modeling (large scale) scenes

[Adam Rachmielowski]

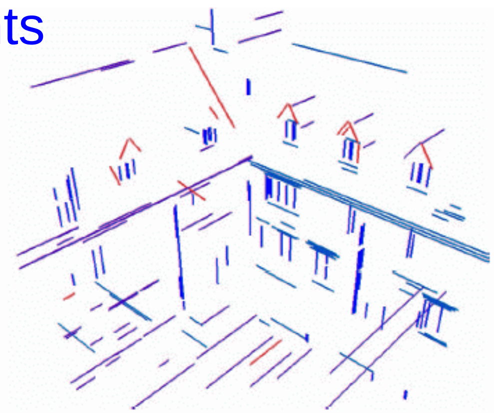
SFM + stereo

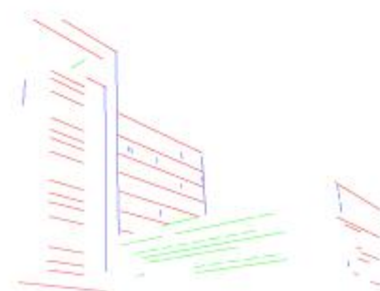
Man-made environments :

- straight edges
- family of lines
- vanishing points

[Dellaert et al 3DPVT06]

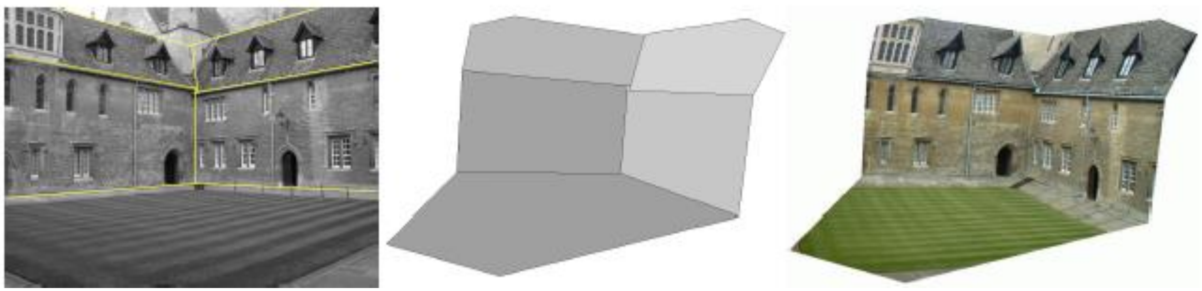
[Zisserman, Werner ECCV02]

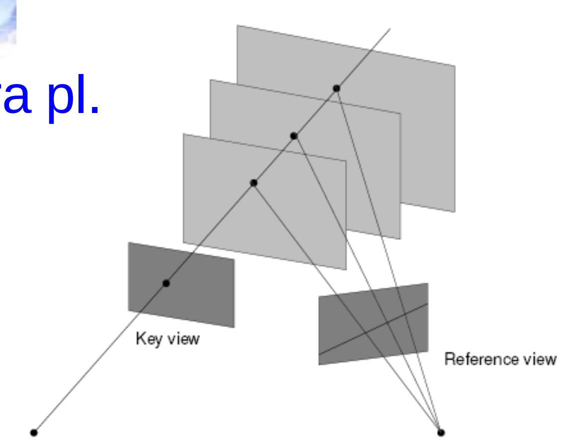


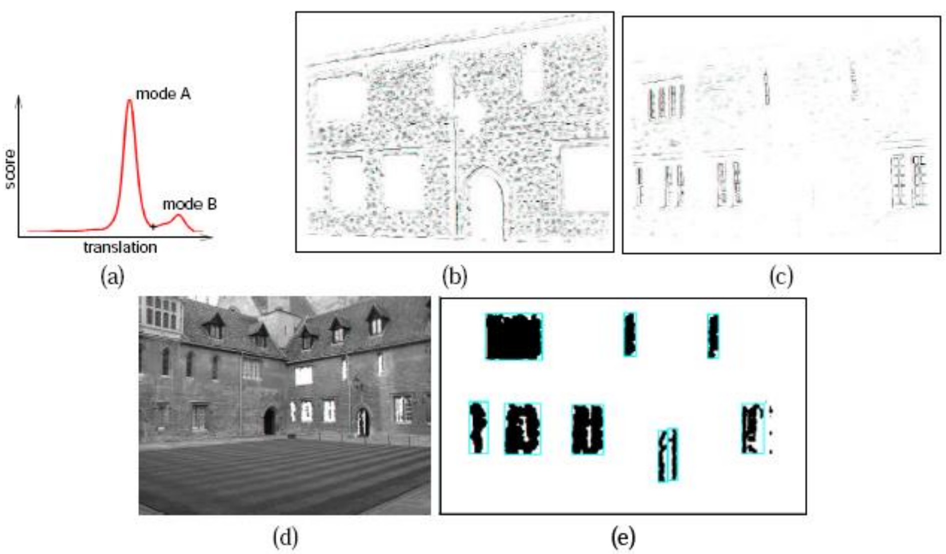


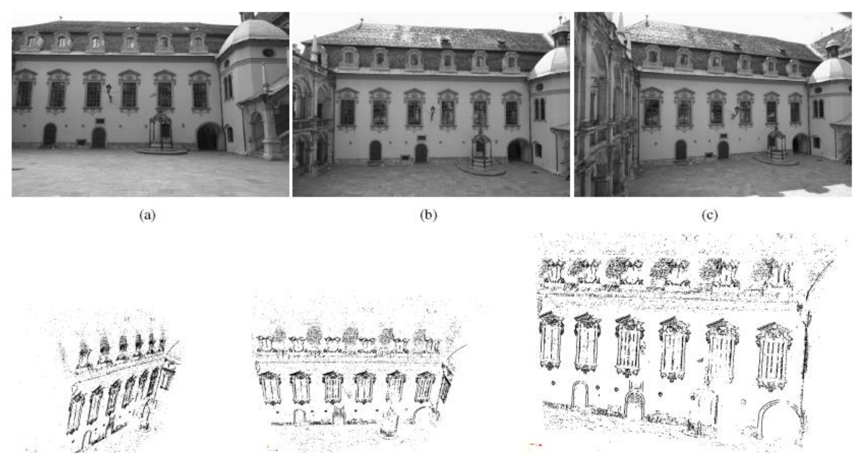
SFM + stereo

- dominant planes
- plane sweep – homog between 3D pl. and camera pl.







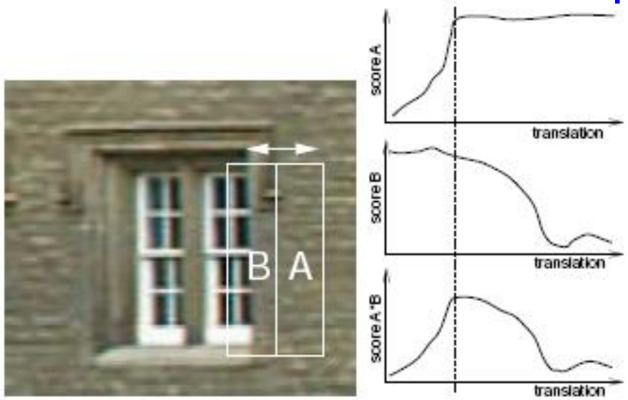


[Zisserman, Werner ECCV02]

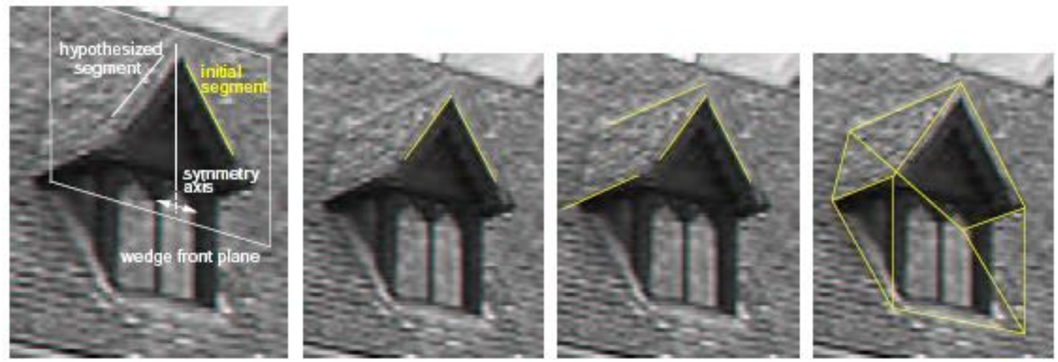
[Bischof et al 3DPVT06]

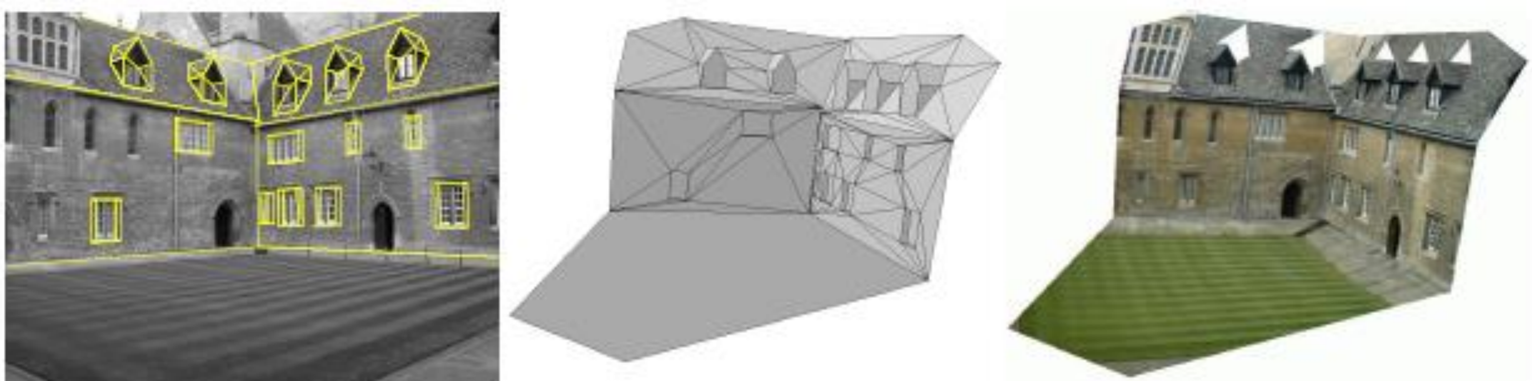
SFM + stereo

- refinement – architectural primitives



(a) (b)

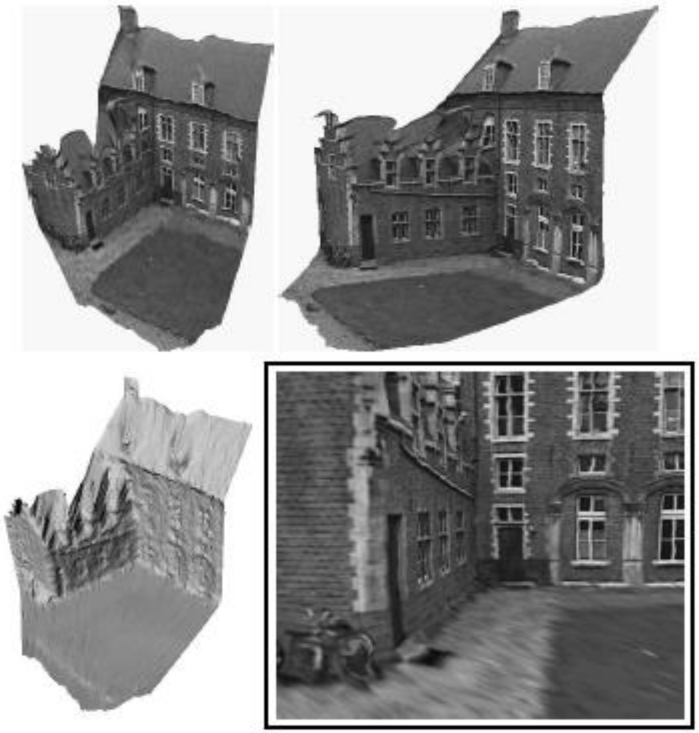




[Zisserman, Werner ECCV02 …]

SFM+stereo

- Refinement – dense stereo



ARC 3D Webservice

A Family of Web Tools for Remote 3D Reconstruction

www.arc3d.be

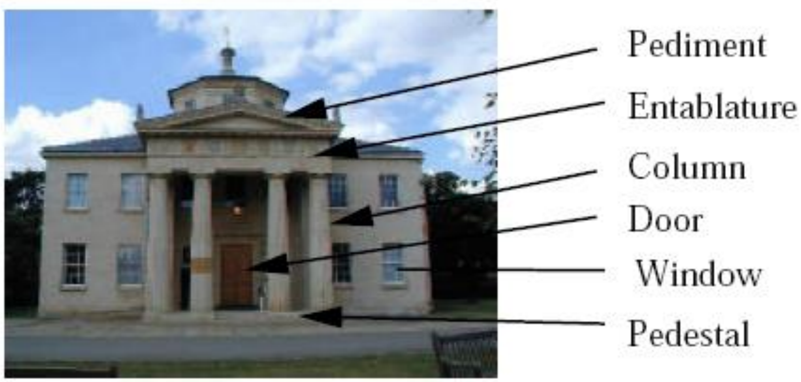
[**Pollefeys, Van Gool 98,00,01**]

Façade – first system

Based on SFM
(points, lines, stereo)
Some manual modeling
View dependent texture

[Debevec, Taylor et al. Siggraph 96]

Priors on architectural primitives



$$\Pr(\mathbf{M}\boldsymbol{\theta}|\mathbf{DI}) \ \alpha \ \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I}) \Pr(\mathbf{M}\boldsymbol{\theta}\mathbf{I})$$

$$= \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}_L\boldsymbol{\theta}_S\boldsymbol{\theta}_T\mathbf{I}) \Pr(\boldsymbol{\theta}_T|\boldsymbol{\theta}_L\mathbf{MI}) \Pr(\boldsymbol{\theta}_S|\boldsymbol{\theta}_L\mathbf{MI}) \Pr(\boldsymbol{\theta}_L|\mathbf{MI})$$

prior

θ – parameters for architectural priors

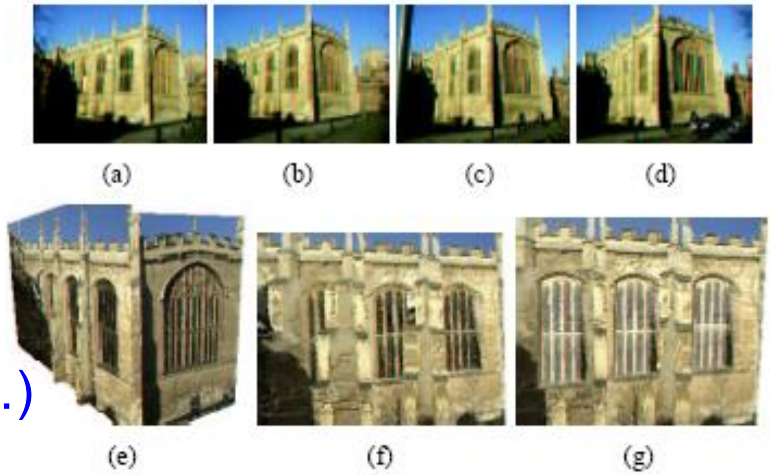
type, shape, texture

M – model

D – data (images)

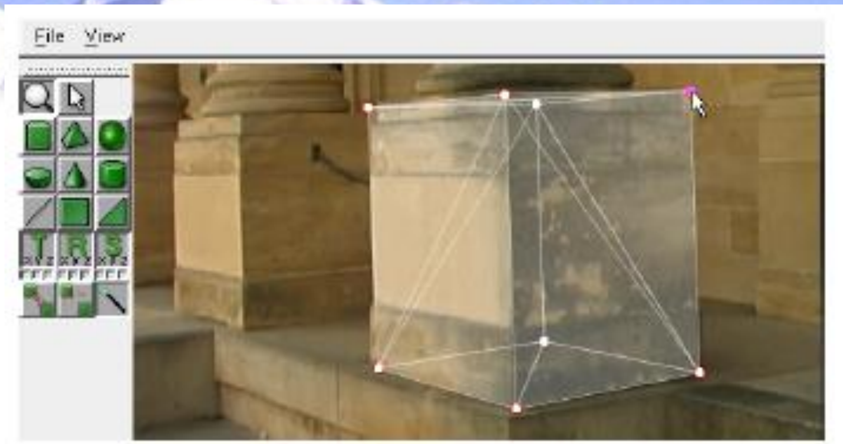
I – reconstructed structures (planes, lines …)

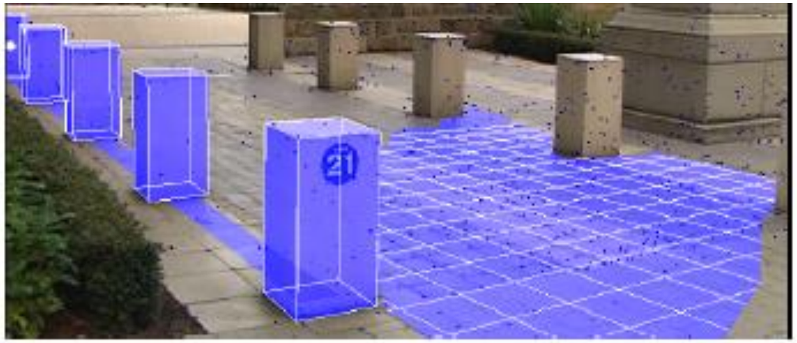
[**Cipolla, Torr, … ICCV01**]



Occluded windows

Interactive systems





Video, sparse 3D points, user input

$$\Pr(\mathcal{M}|\mathrm{DI}) \propto \Pr(\mathrm{D}|\mathcal{M}\mathrm{I})\Pr(\mathcal{M}|\mathrm{I}).$$

M – model primitives

D- data

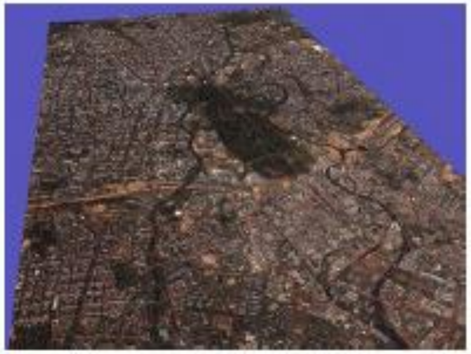
I – reconstructed geometry

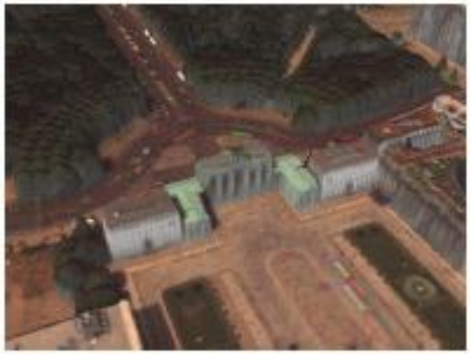
Solved with graph cut

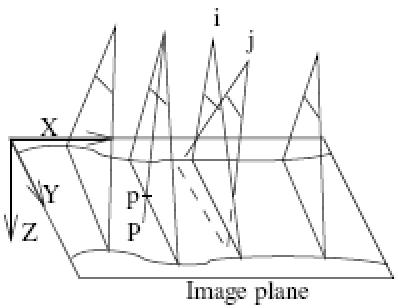
[Torr et al. Eurogr.06, Siggraph07]

VideoTrace

City modeling – aerial images







(c) 2D Image Creation

[Heiko Hirschmuller et al - DLR]

Airborne pushbroom camera

Semi-global stereo matching
(based on mutual information)

City modeling – ground plane

Camera cluster

car + GPS

Video: Cannot do frame-frame correspondences

Calibrated cameras – relative pose
GPS – car position - 3D tracking

→ 2D feature tracker
SFM

↓

3D points
Dense stereo+fusion
Texture

→ 3D MODEL

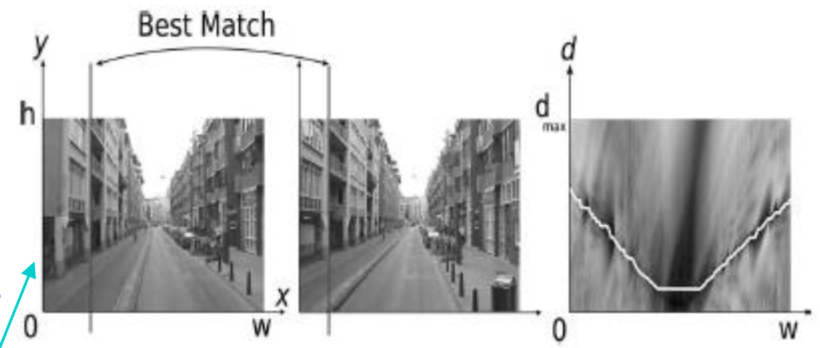
[Nister, Pollefeys et al 3DPVT06, ICCV07]

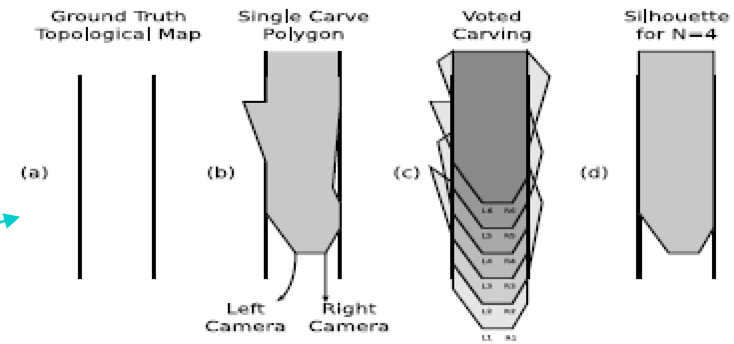
[Cornelis, Van Gool CVPR06…]

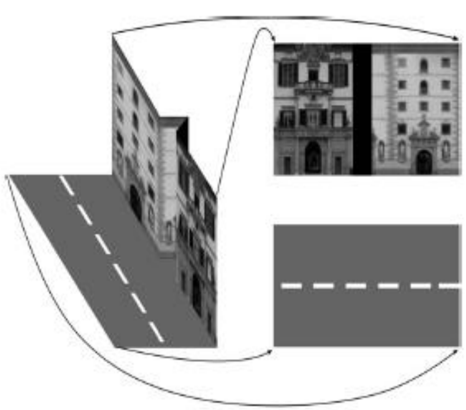
City modeling - example

[Cornelis, Van Gool CVPR06…]

1. feature matching = tracking
2. SFM – camera pose + sparse 3D points
3. Façade reconstruction
   – rectification of the stereo images
   - vertical line correlation
4. Topological map generation
   - orthogonal proj. in the horiz. plane
   - voting based carving
5. Texture generation
   - each line segment – column in texture space



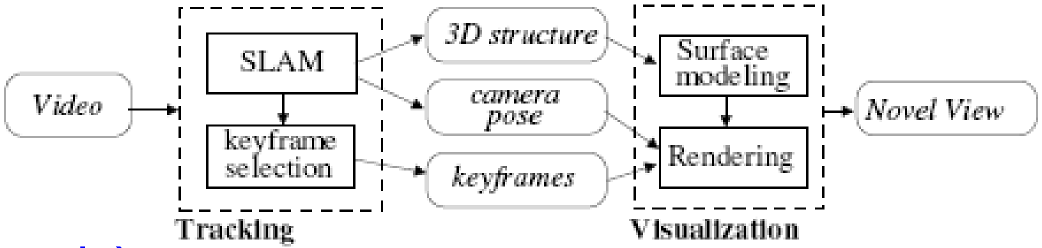




VIDEO

On-line scene modeling : Adam's project

On-line modeling from video
Model not perfect but enough for scene visualization
Application predictive display

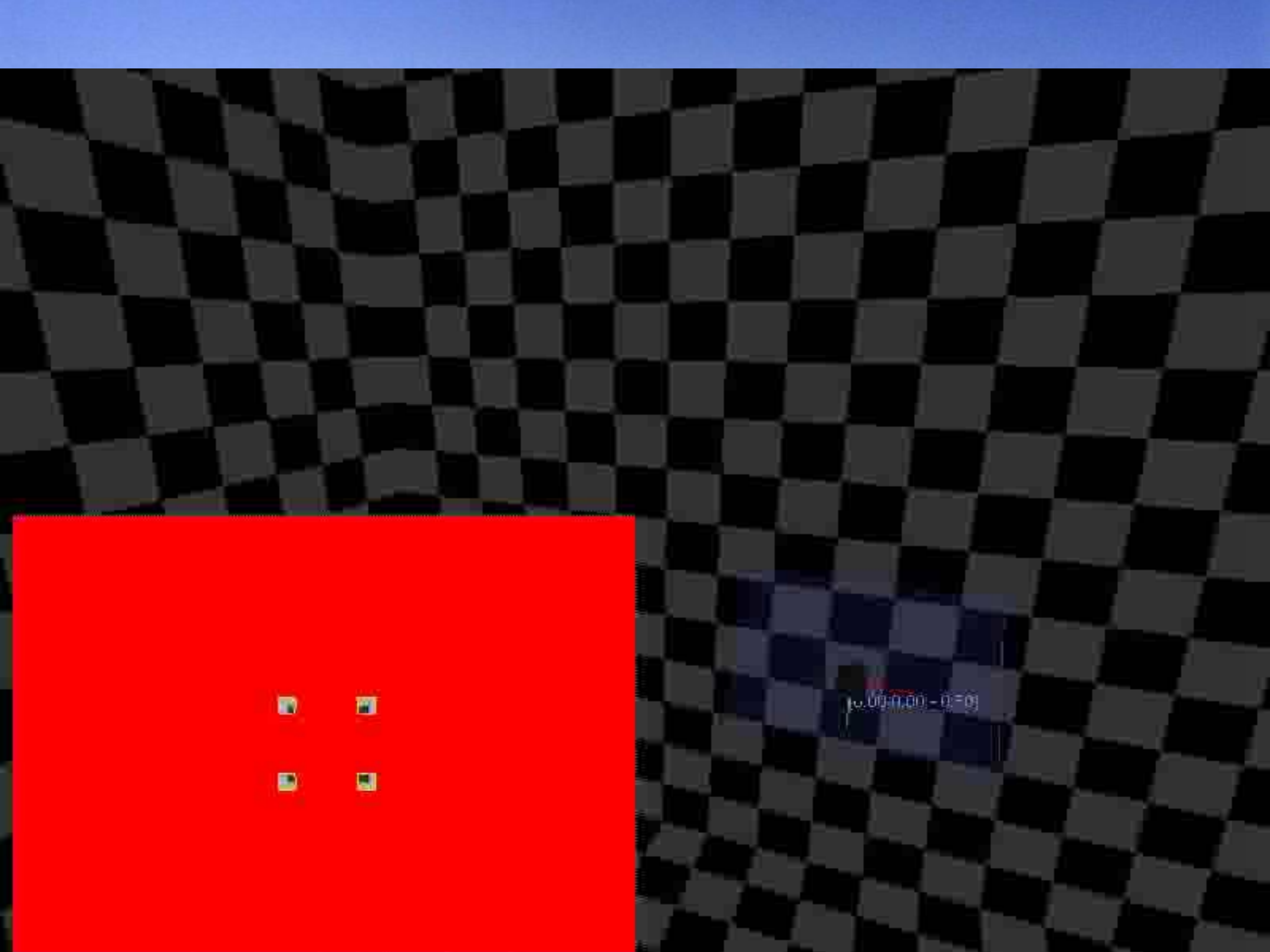


Tracking and Modeling
- New image
- Detect fast corners (similar to Harris)
- SLAM (mono SLAM [Davison ICCV03])
  - Estimate camera pose
  - Update visible structure
- Partial bundle adjustment – update all points
- Save image if keyframe (new view – for texture)

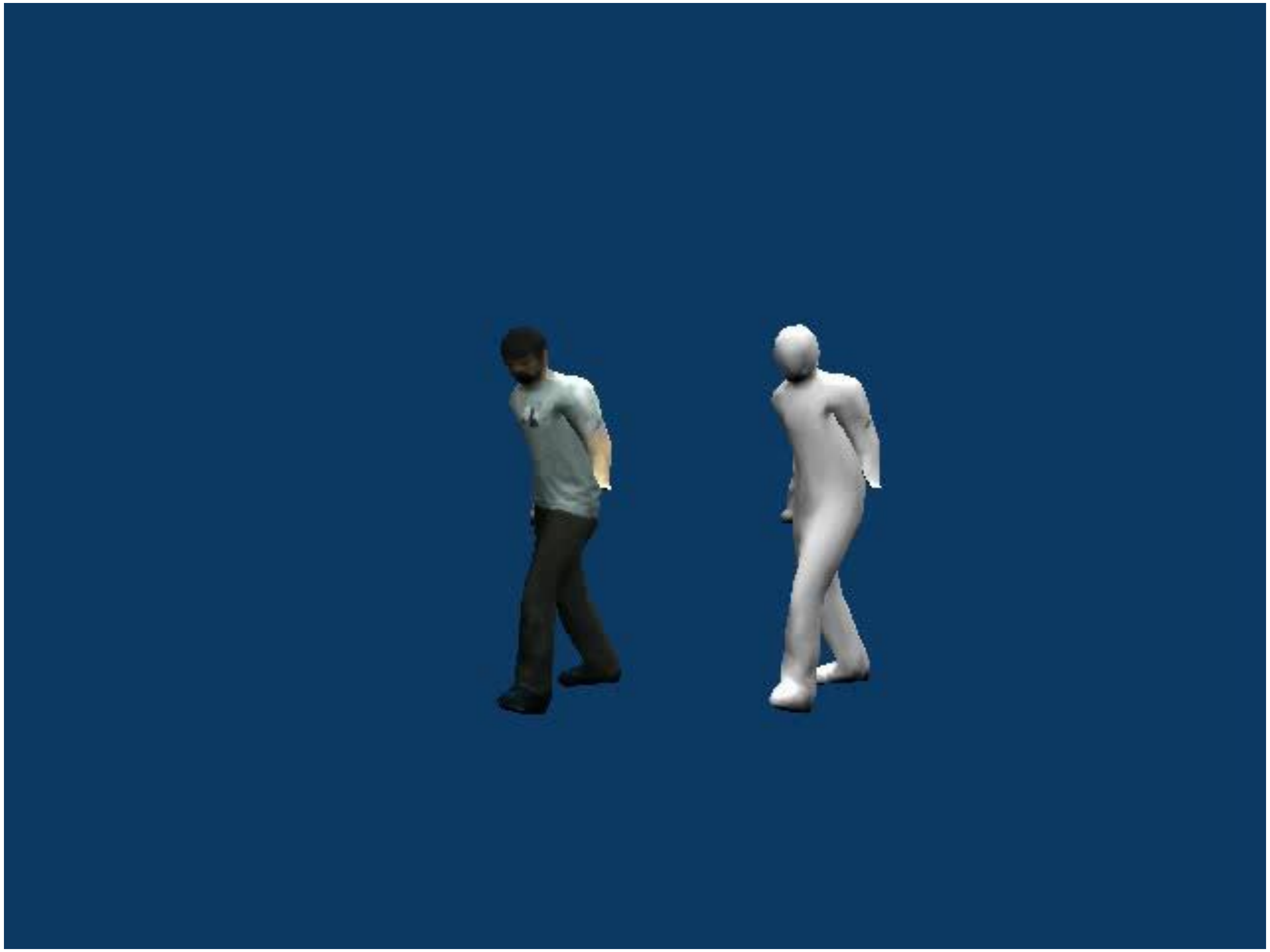
Visualization
- New visual pose
  - Compute closet view
  - Triangulate
  - Project images from closest views onto surface

SLAM
Camera pose
3D structure
Noise model
Extended Kalman Filter

Model refinement

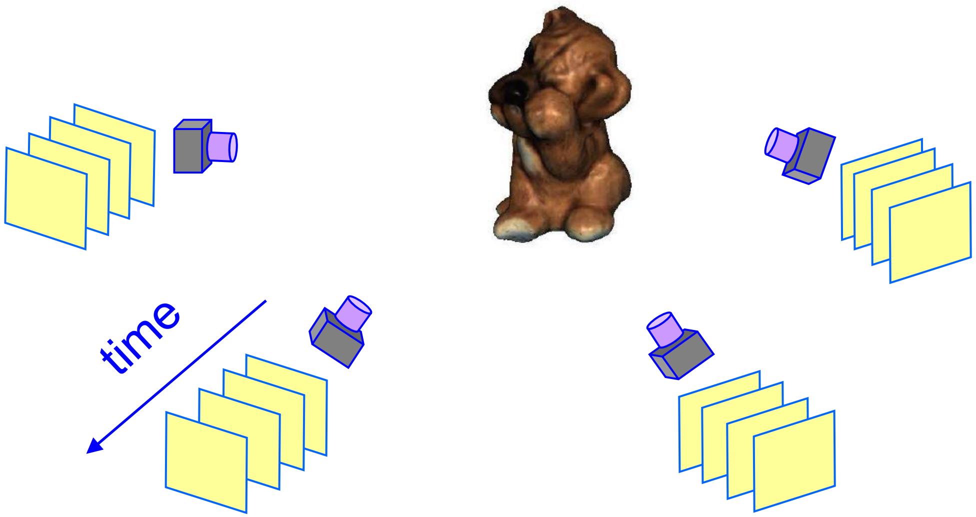


Modeling dynamic scenes



[Neil Birkbeck]

Multi-camera systems



Several cameras mutually registered (precalibrated)

Video sequence in each camera

Moving object

Techniques

- Naïve : reconstruct shape every frame
- Integrate stereo and image motion cues
- Extend stereo in temporal domain
- Estimate scene flow in 3D from optic flow and stereo

Representations :

- Disparity/depth
- Voxels / level sets
- Deformable mesh – hard to keep time consistency

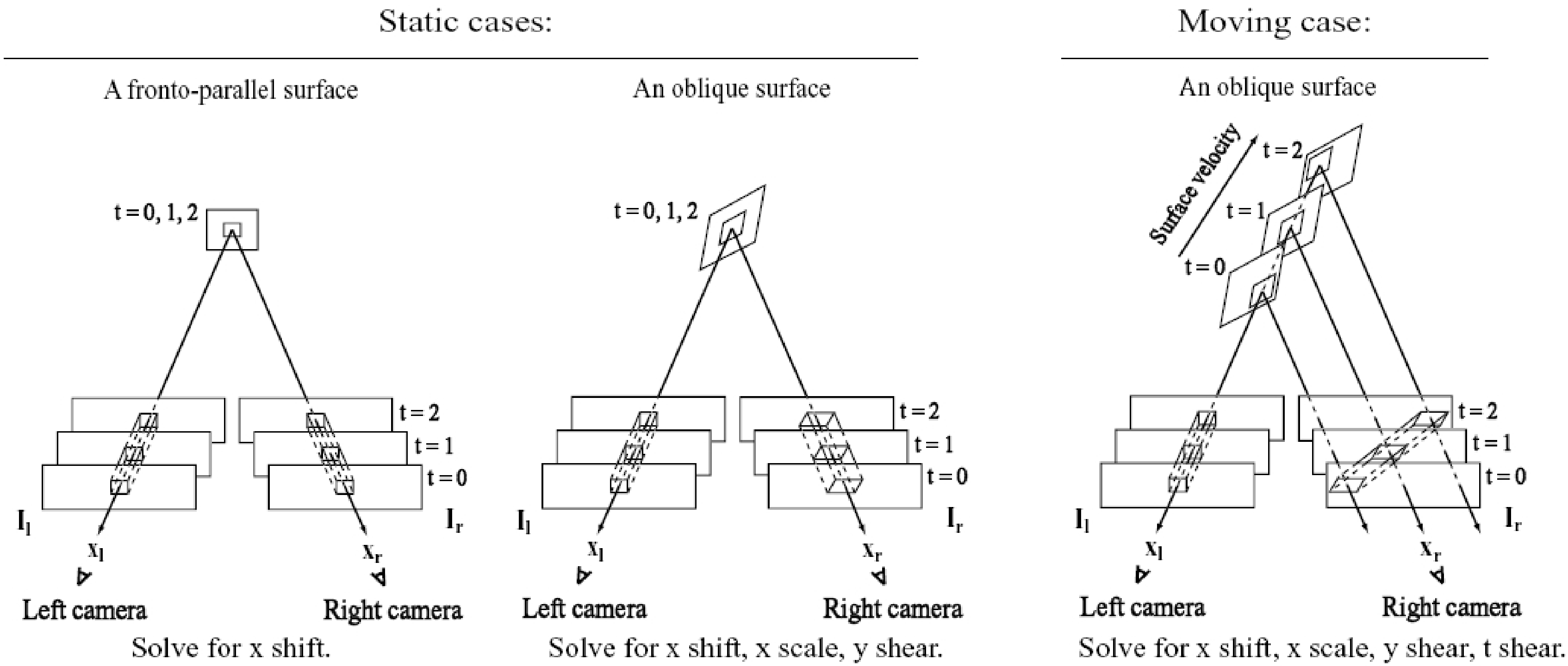
Knowledge:

- Camera positions
- Scene correspondences (structured light)

Spacetime stereo

[Zhang, Curless, Seitz: Spacetime stereo, CVPR 2003]

Extends stereo in time domain: assumes intra-frame correspondences



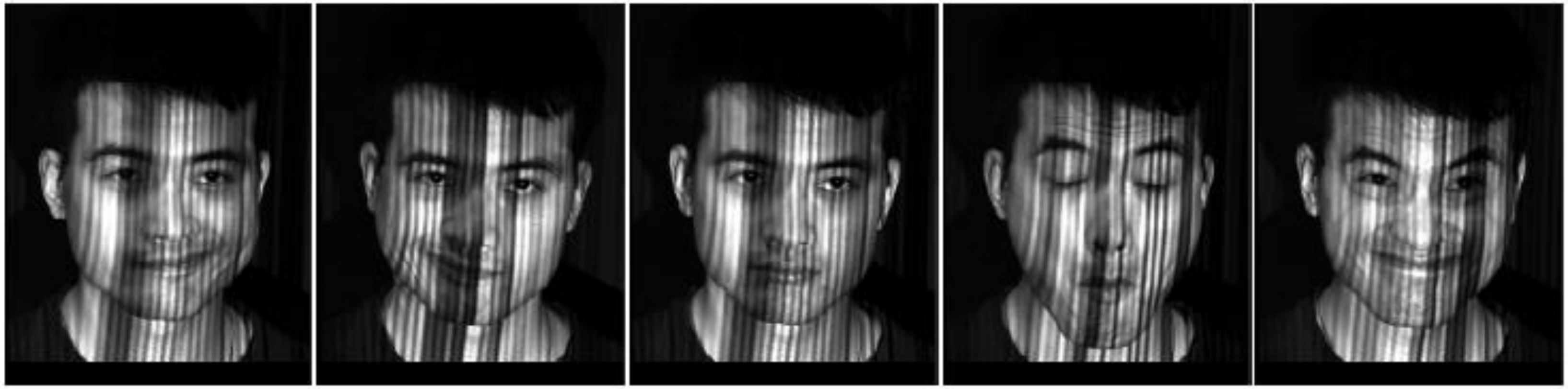
Static scene: disparity

$$d(x,y,t) \approx d_0 + d_{x_0}(x - x_0) + d_{y_0}(y - y_0)$$

Dynamic scene:

$$d(x,y,t) \approx d_0 + d_{x_0}(x - x_0) + d_{y_0}(y - y_0) + \boxed{d_{t_0}(t - t_0)}$$

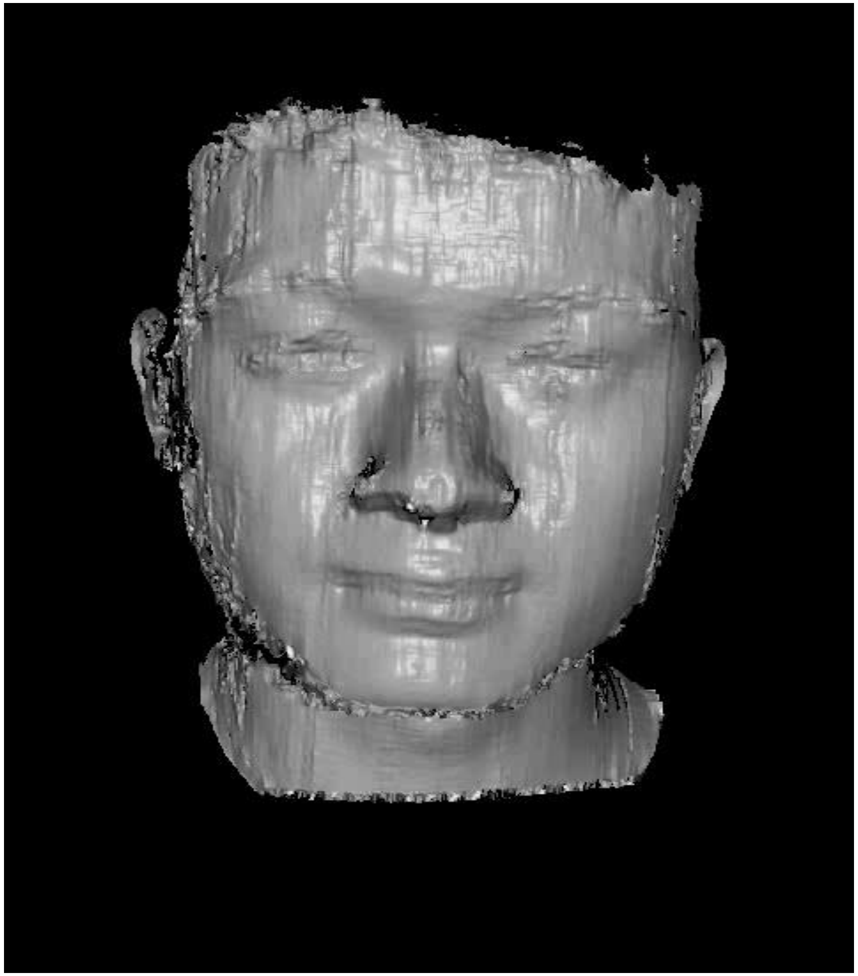
Spacetime stereo: Results



Input: 400 stereo pairs (5 left camera imagest shown here)

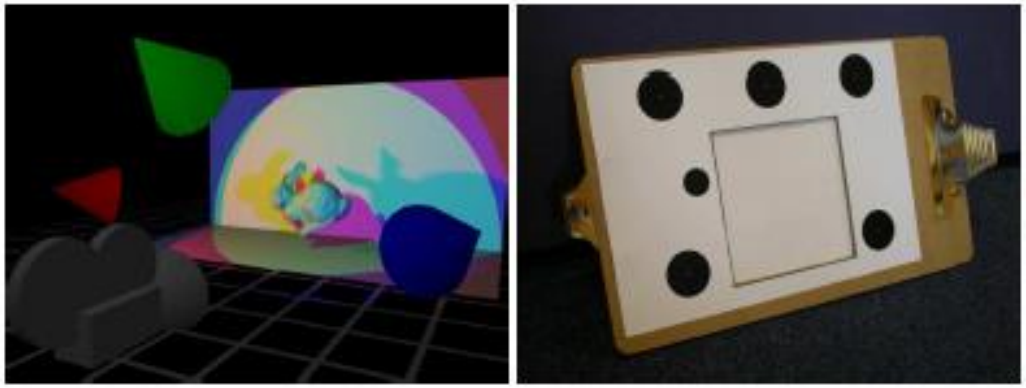
Spacetime stereo reconstruction with 9x5x5 window

Spacetime stereo:video



Spacetime photometric stereo

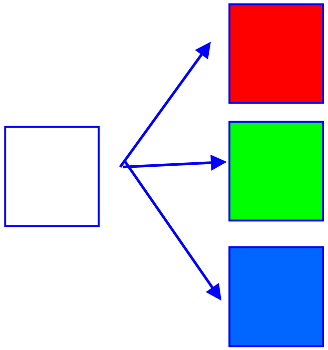
[Hernandez et al. ICCV 2007]



One color camera

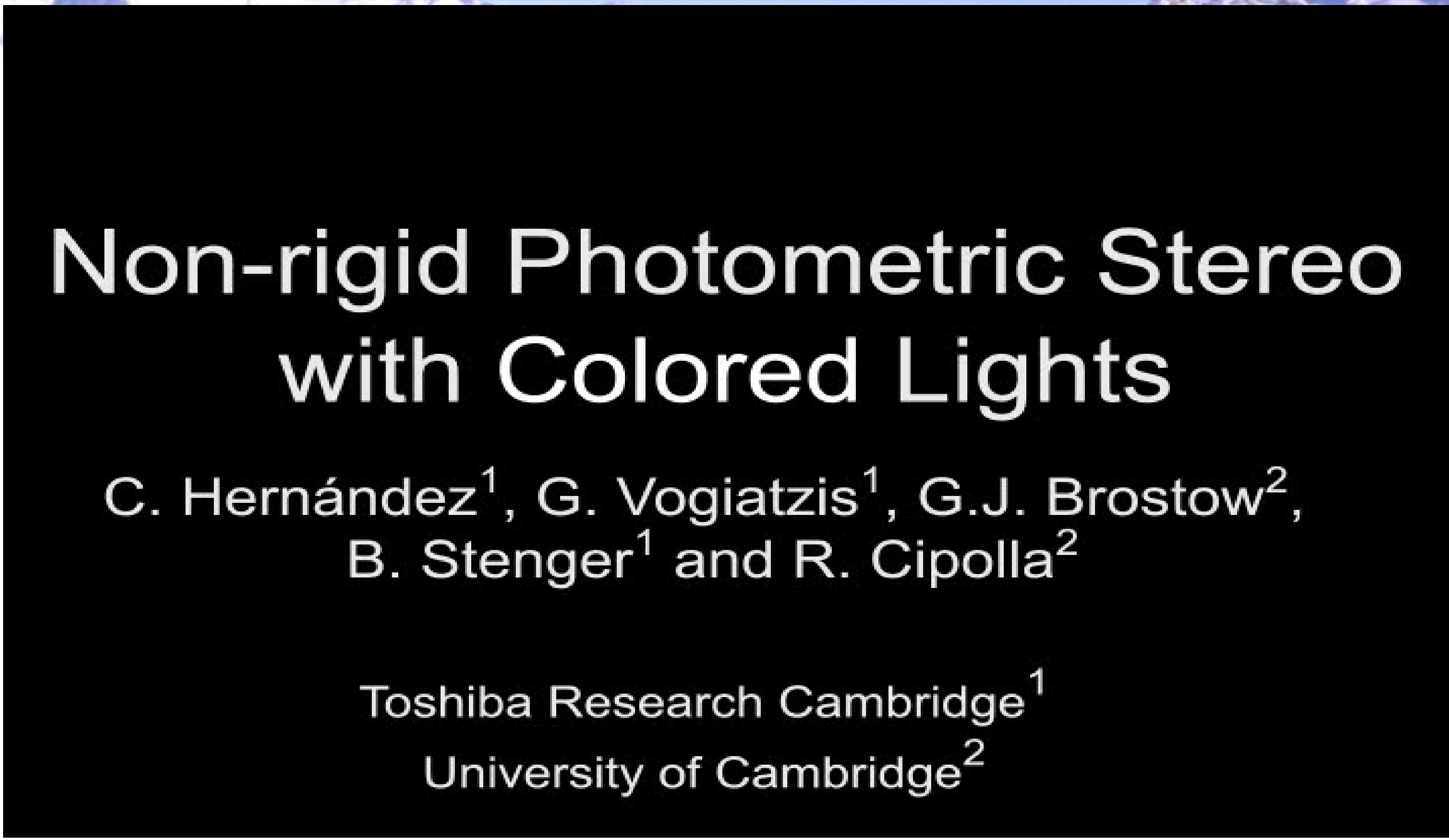
projectors – 3 different positions

Calibrated w.r.t. camera



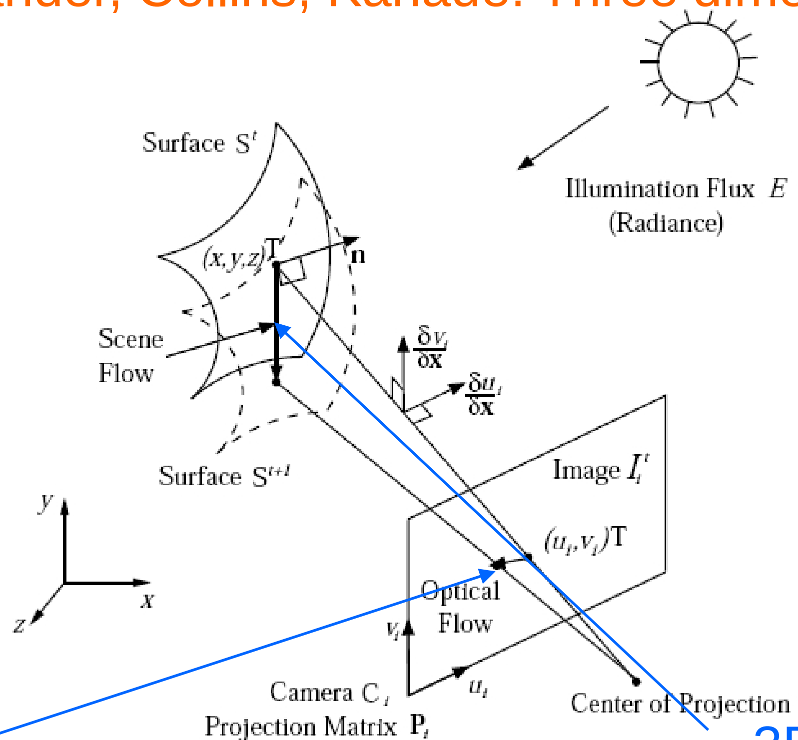
Each channel (R,G,B) – one colored light pose
Photometric stereo

Spacetime PS - Results



3. Scene flow

[Vedula, Baker, Rander, Collins, Kanade: Three dimensional scene flow, ICCV 99]



2D Optic flow

$$\frac{d\mathbf{u}}{dt}: \quad \nabla I_i \frac{d\mathbf{u}_i}{dt} + \frac{\partial I_i}{\partial t}$$

3D Scene flow

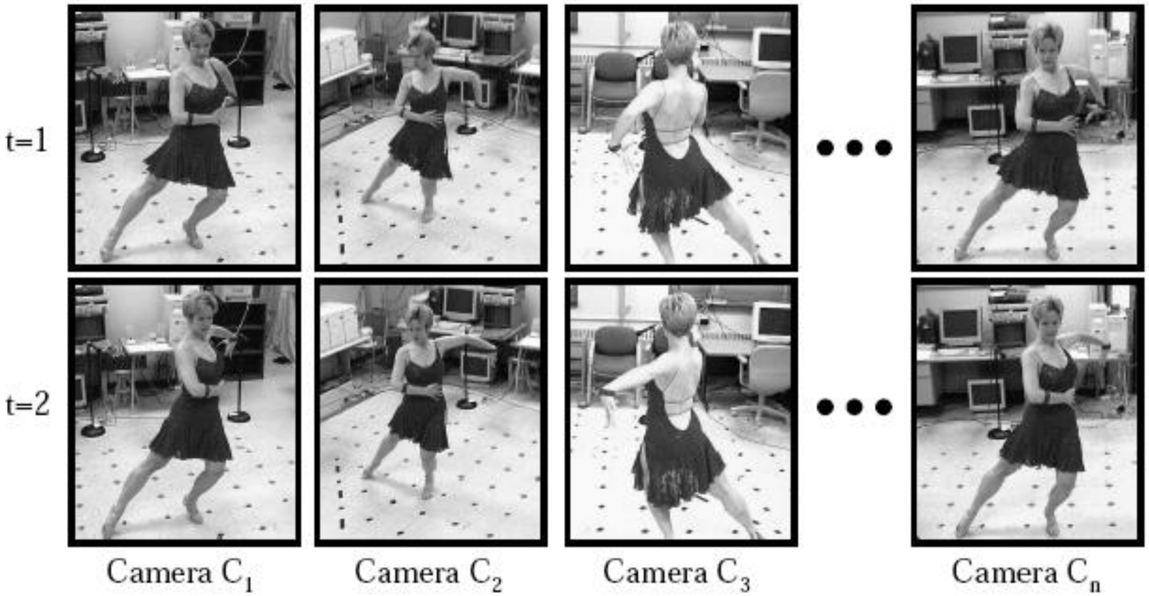
$$\mathbf{x} = \mathbf{x}(\mathbf{u}_i(t); t)$$

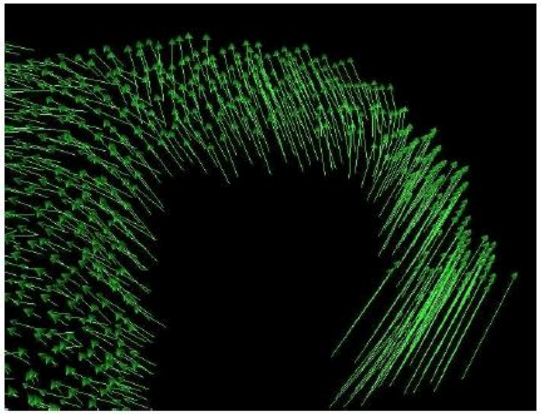
$$\frac{d\mathbf{x}}{dt} = \underbrace{\frac{\partial \mathbf{x}}{\partial \mathbf{u}_i} \frac{d\mathbf{u}_i}{dt}} + \underbrace{\left.\frac{\partial \mathbf{x}}{\partial t}\right|_{u_i}}$$

Scene flow on tangent plane

Motion of x along a ray

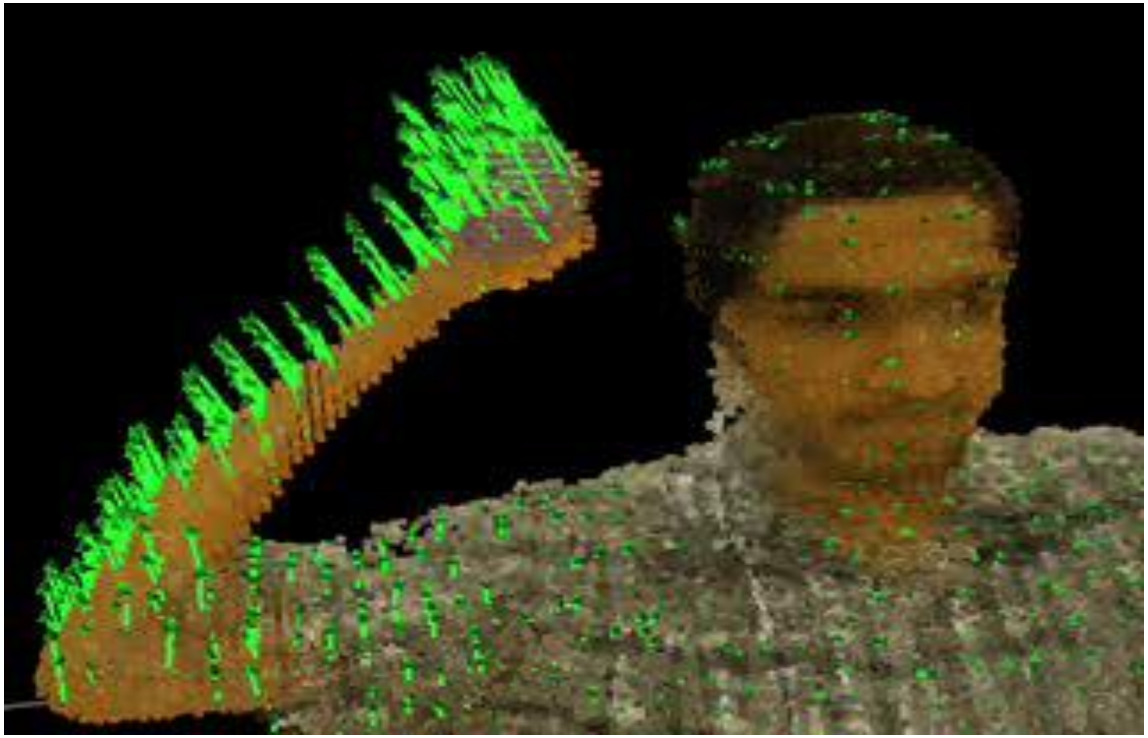
Scene flow: results





[Vedula, et al. ICCV 99]

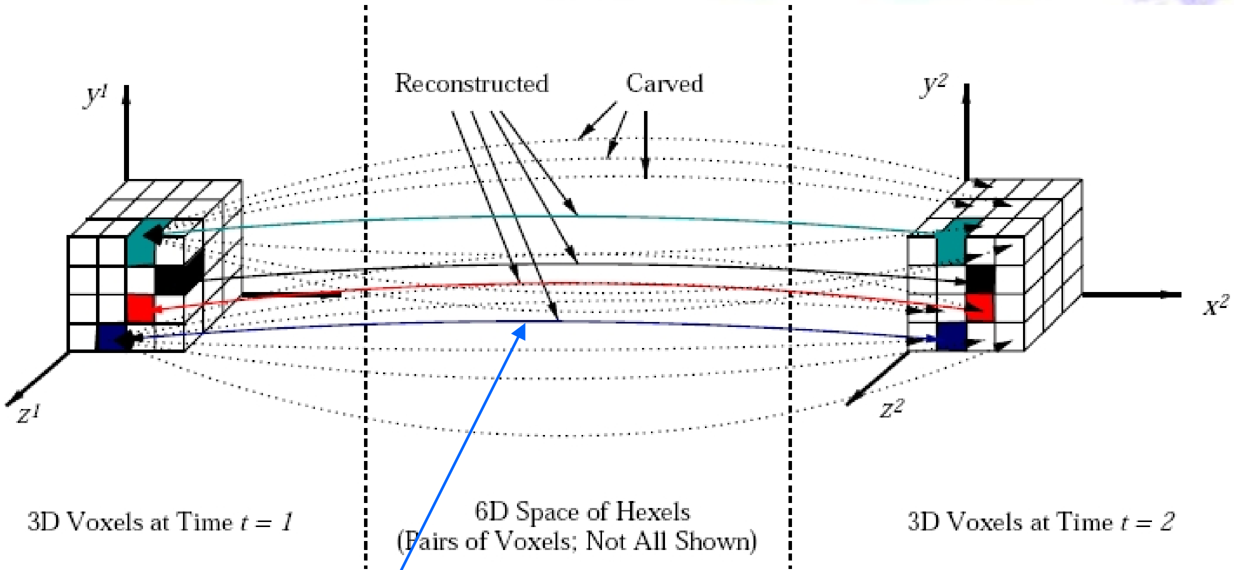
Scene flow: video



[Vedula, et al. ICCV 99]

4. Carving in 6D

[Vedula, Baker, Seitz, Kanade: Shape and motion carving in 6D]



Hexel:

$$(x_1, y_1, z_1, \Delta x, \Delta y, \Delta z)$$

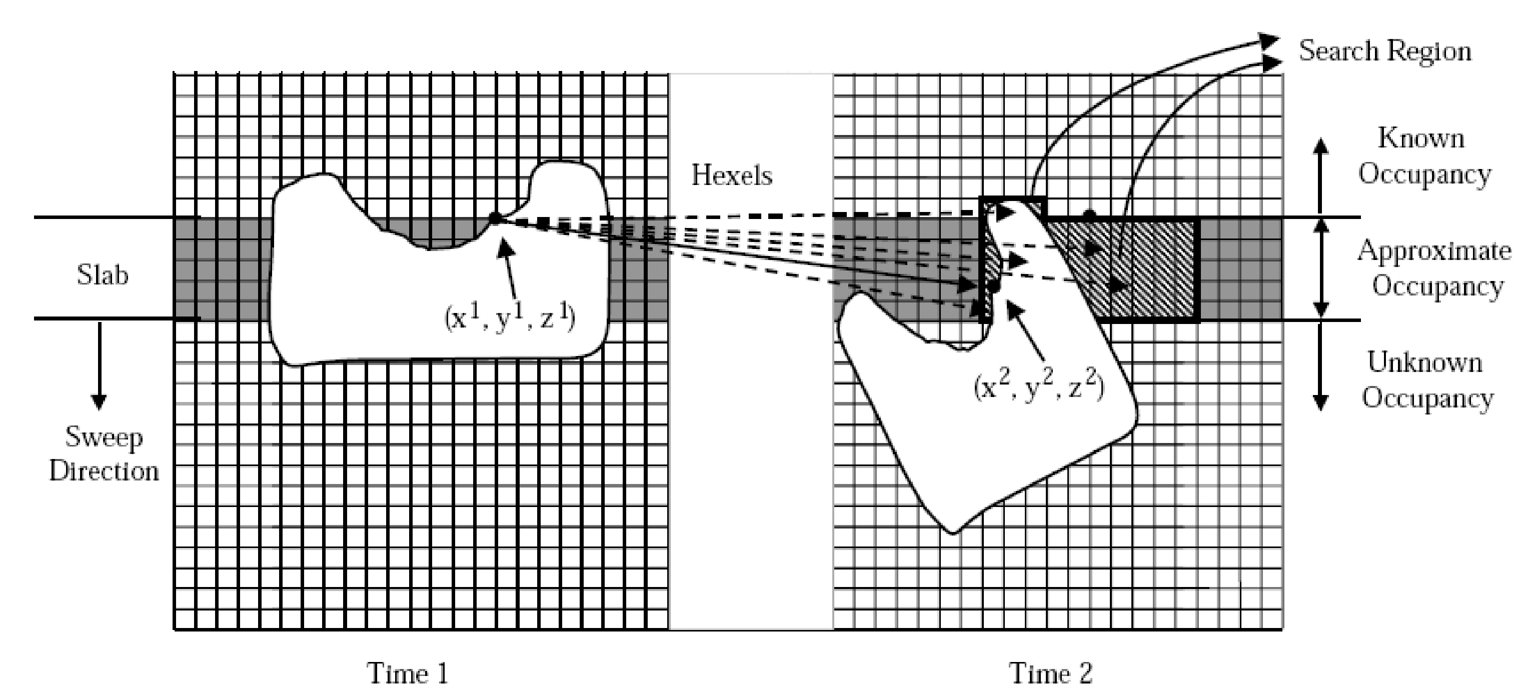
$$(x_2, y_2, z_2) = (x_1, y_1, z_1) + (\Delta x, \Delta y, \Delta z)$$

6D photo-consistency:

$$S^t = \sum_i I_i^t(P_i(\mathbf{x}^t)); \quad SS^t = \sum_i \left(I_i^t(P_i(\mathbf{x}^t))\right)^2$$

$$\frac{SS^1 + SS^2 - (S^1 + S^2) * (S^1 + S^2)}{n^1 + n^2}$$

6D slab sweeping

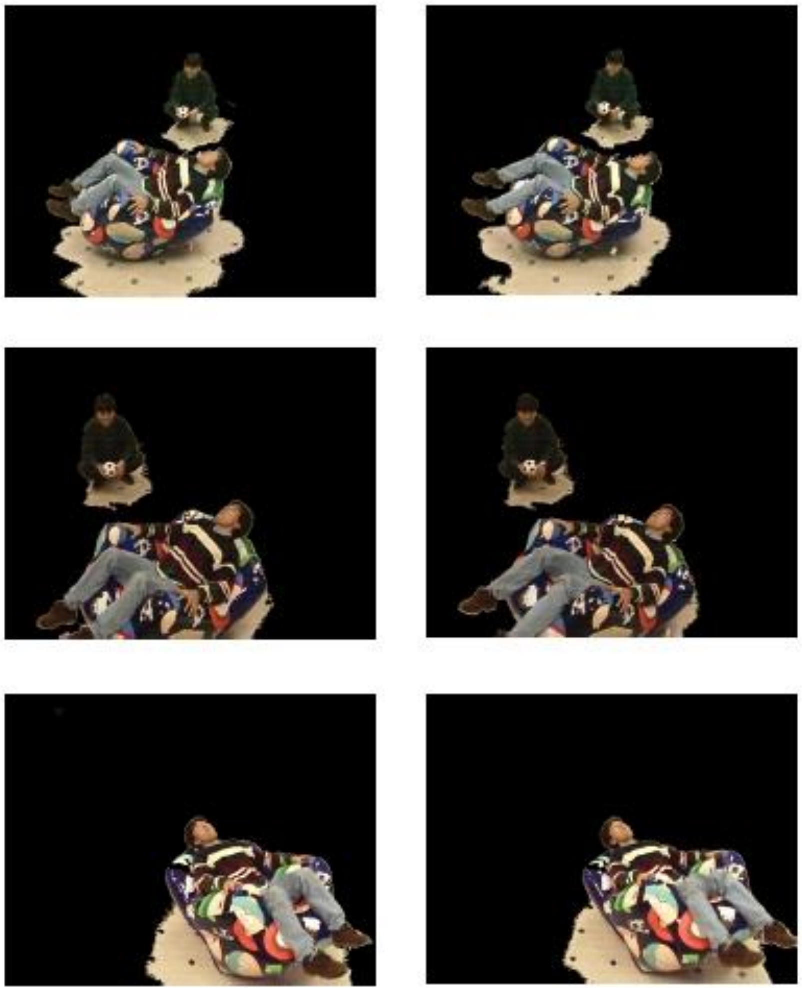


Slab = thickened plane (thikness = upper bound on the flow magnitude)

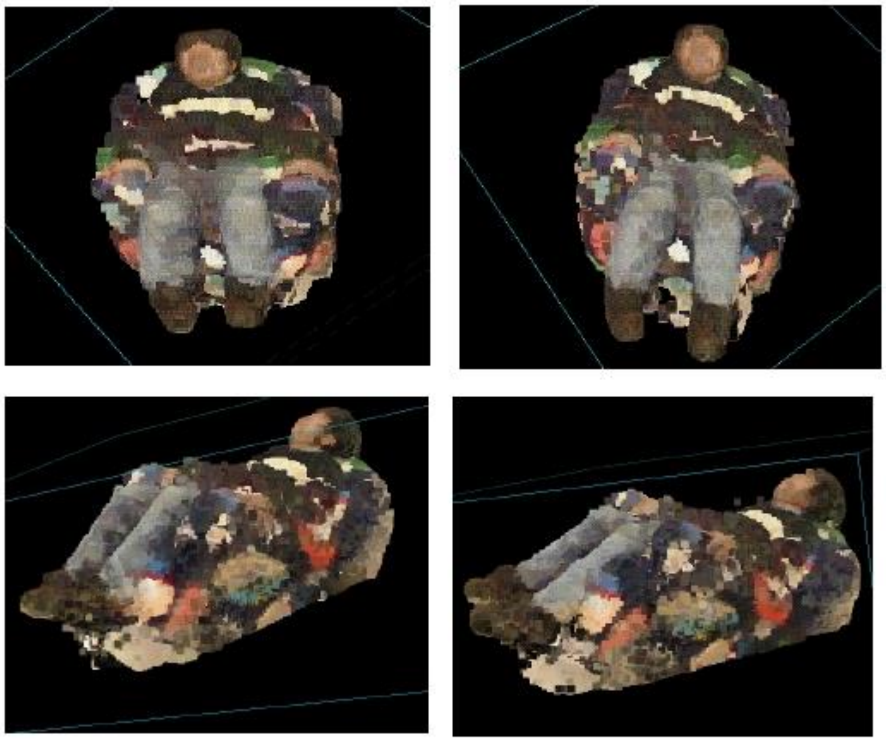
- compute visibility for x^1
- determine search region
- compute all hexel photo-consistency
- carving hexels
- update visibility

(Problem: visibility below the top layer in the slab before carving)

Carving in 6D: results



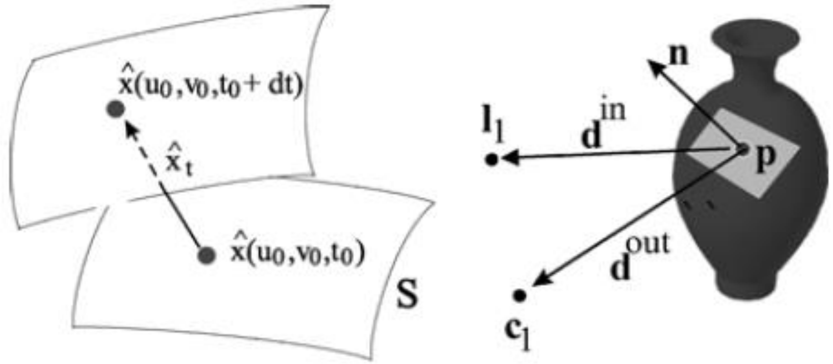
Time 1 Time 2



Time 1 Time 2

7. Surfel sampling

[Carceroni, Kutulakos: Multi-view scene capture by surfel samplig, ICCV01]



Surfel: dynamic surface element

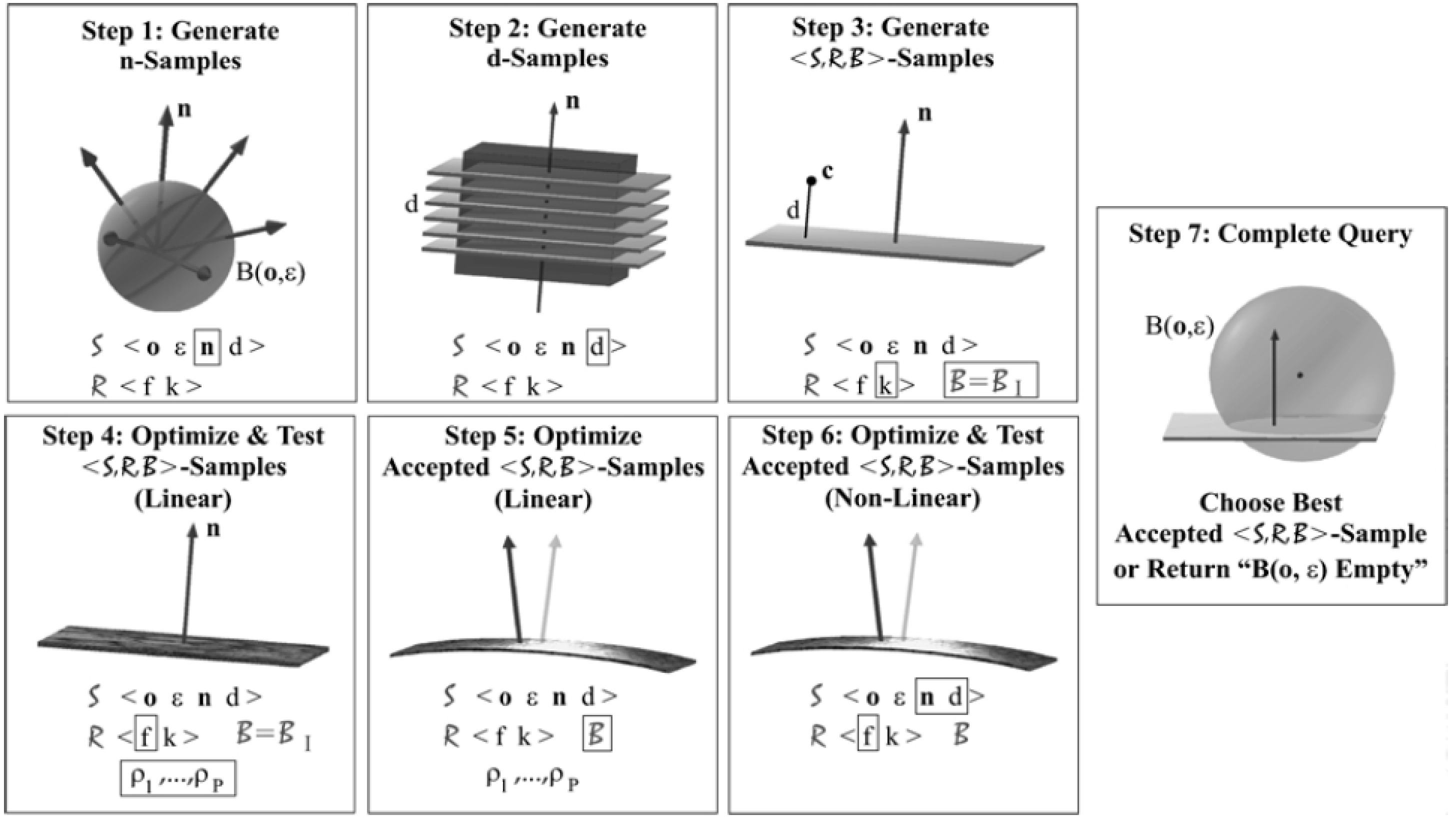
- shape component : center, normal, curvature
- motion component:
- reflectance component: Phong parameters

$$S = \langle \mathbf{x}_0, \mathbf{n}_0, \mathbf{k} \rangle$$

$$M = \langle \tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_{ut}, \tilde{\mathbf{x}}_{vt} \rangle$$

$$R = \langle f, k, \{\rho_1, \ldots, \rho_P\} \rangle$$

Reconstruction algorithm



$$E(S,R)=\sum_i \sum_j v_{c_i}(\mathbf{p}_i)\left[I_i^{pred}(\mathbf{p}_i)-I_i(\mathbf{p}_i)\right]$$

visibility

$$\langle S^*, R^* \rangle = \arg\min_S (\min_R E(S,R))$$

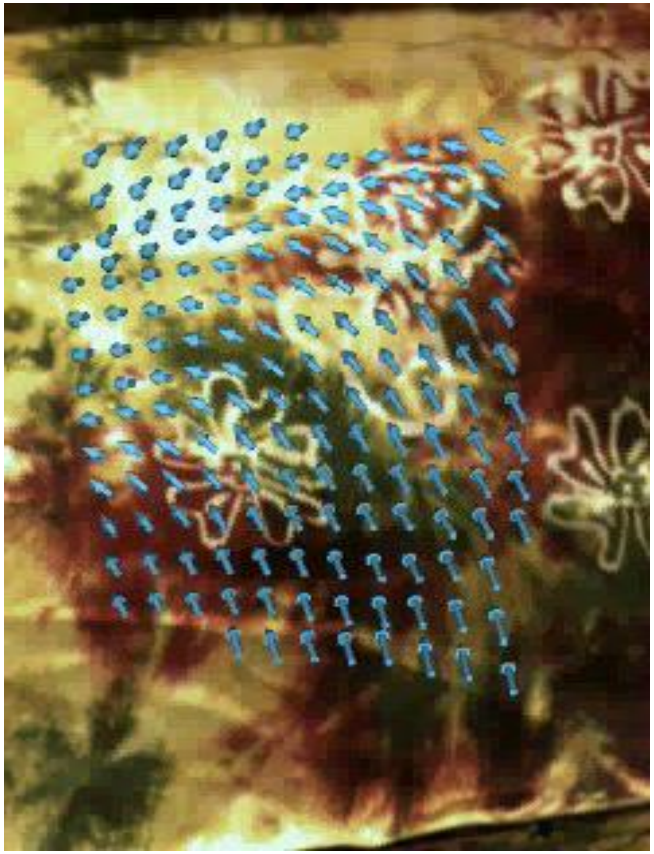
$$I_i^{pred}(\mathbf{p}_i)=\sum_l r(\mathbf{p},\mathbf{n},\mathbf{c}_i-\mathbf{p},\mathbf{l}_l-\mathbf{p})L_l(\mathbf{p})$$

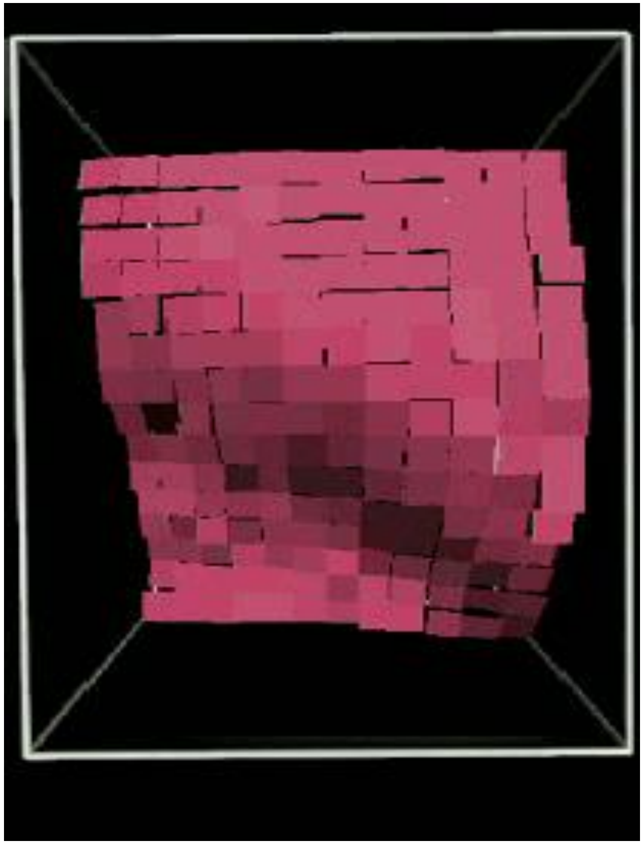
Phong reflectance shadow

c_i – camera i

$\mathbf{l}_l$- light l

Surfel sampling : results

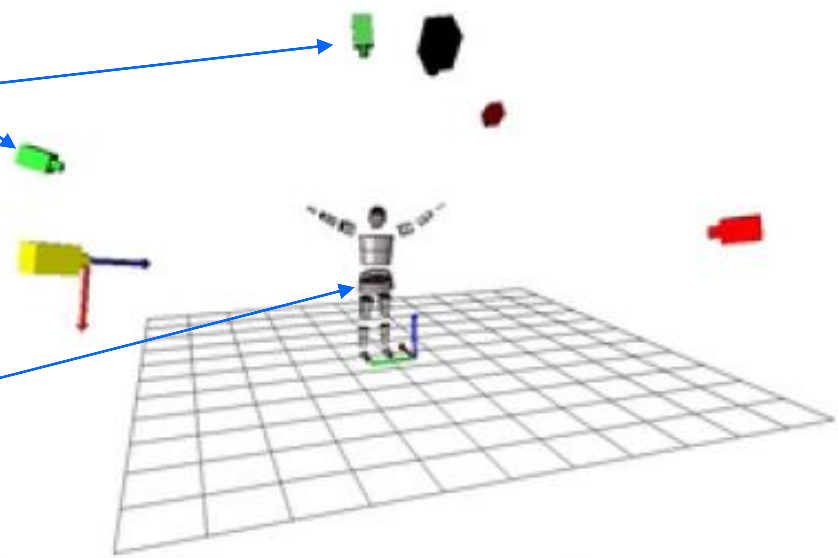




Modeling humans in motion

Multiple calibrated cameras

Human in motion



Goal: 3D model of the human

Instantaneous model that can be viewed from different poses (‘Matrix’) and inserted in an artificial scene (tele-conferences)

GRIMAGE platform- INRIA Grenoble

Our goal: 3D animated human model

- capture model deformations and appearance change in motion
- animated in a video game

[Neil Birkbeck]

Articulated model

Model based approach

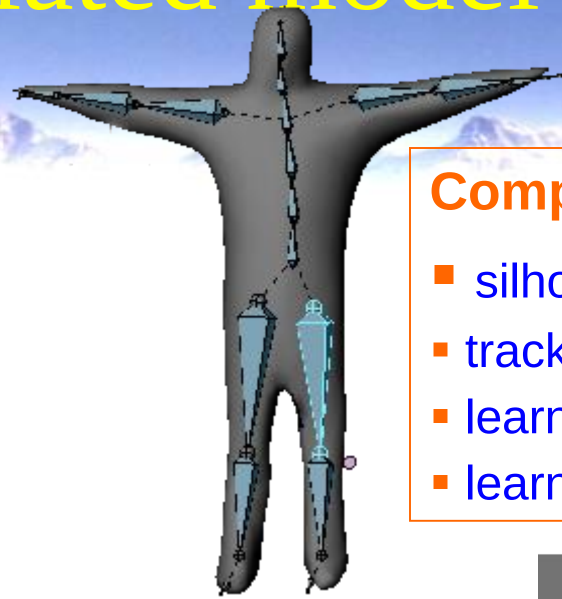
<u>**Geometric Model**</u>

Skeleton + skinned mesh
(bone weights)

50+ DOF (CMU mocap data)

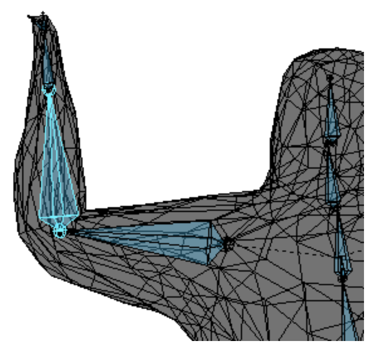
<u>**Tracking**</u>

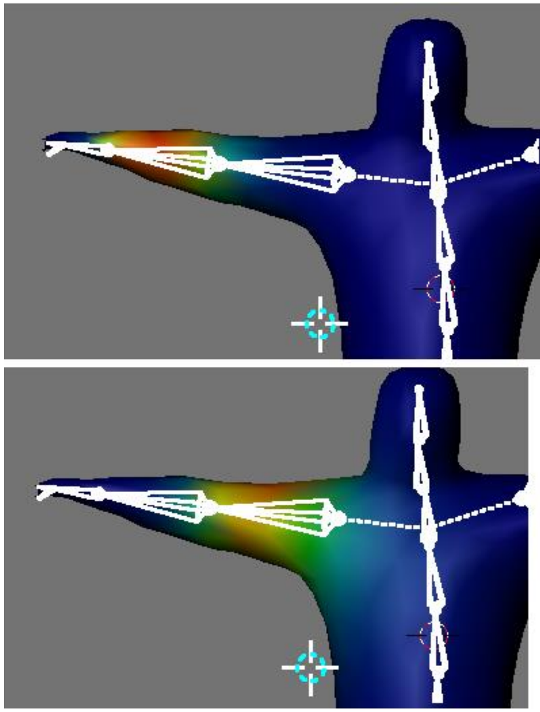
- visual hull – bone weights by diffusion
- refine mesh/weights



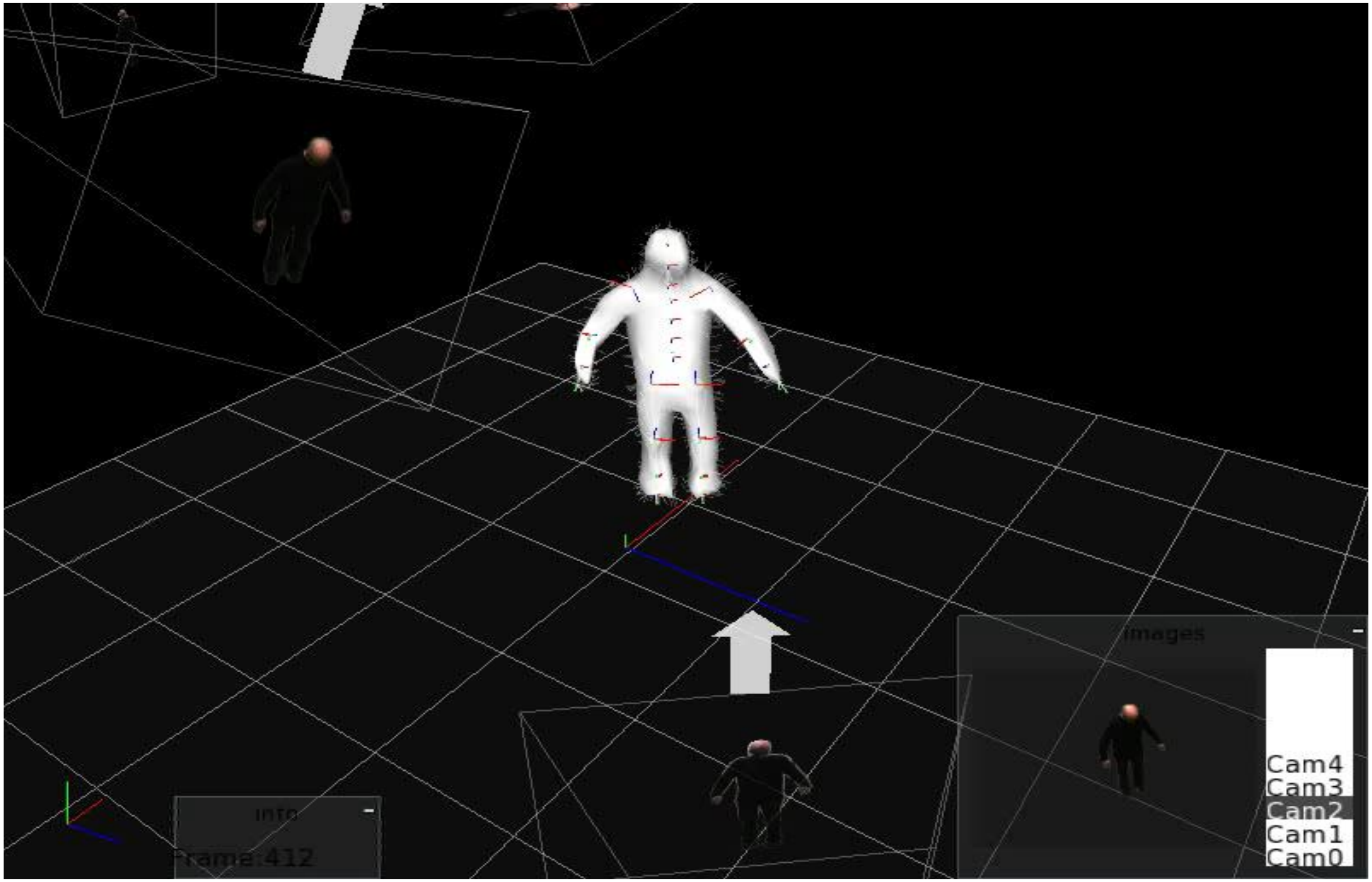
Components

- silhouette extraction
- tracking the course model
- learn deformations
- learn appearance change



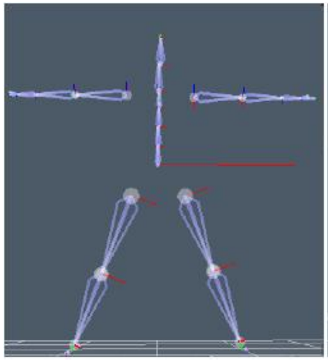
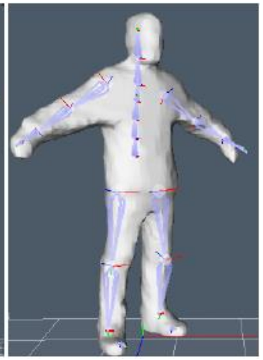


Neil- tracking results

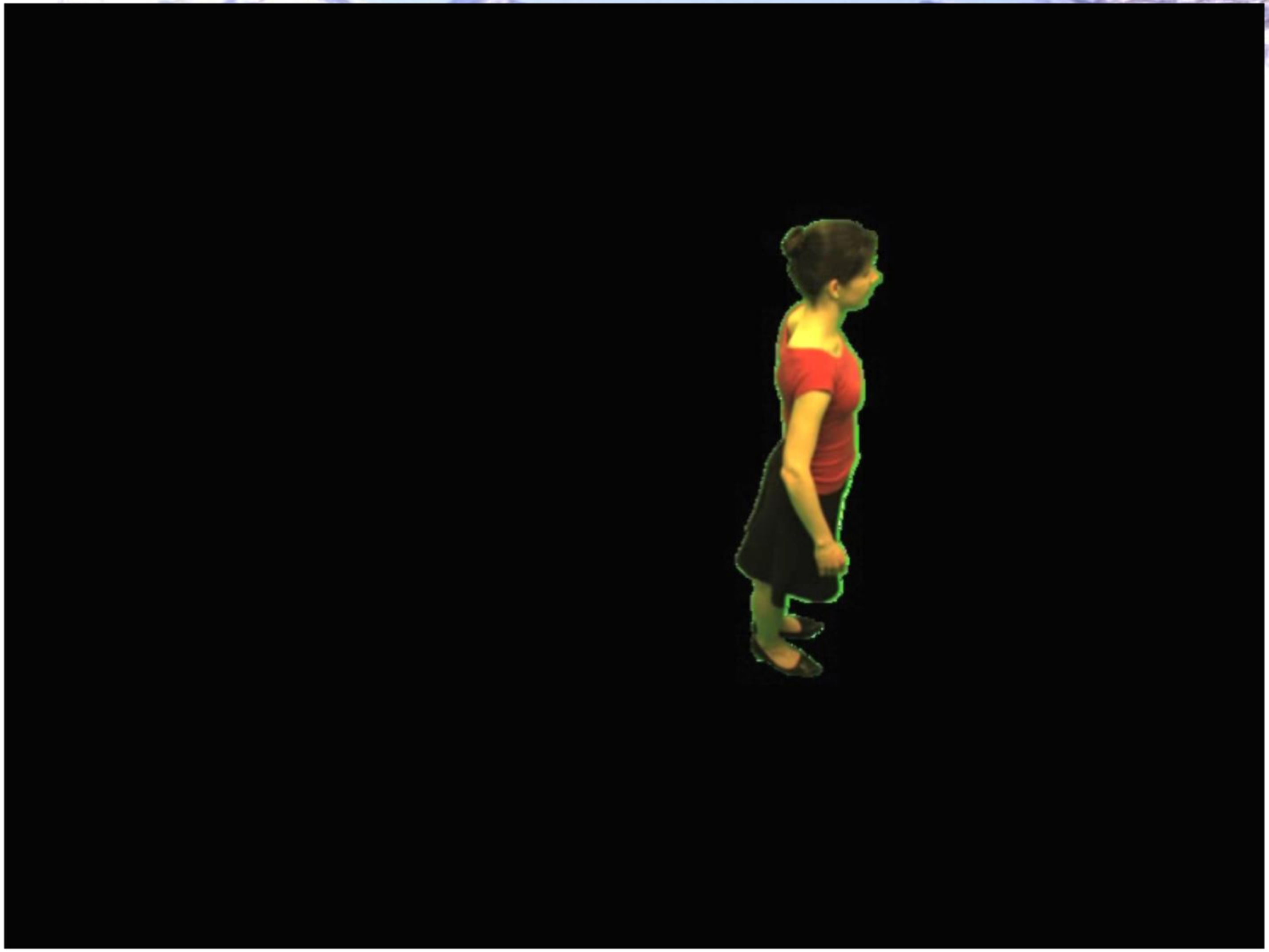


Beyond 3D
Non-rigid and articulated motion

- Humans ubiquitous in graphics applications
- A practical, realistic model requires
  - Skeleton
  - Geometry (manually modeled, laser scanned)
    - Physical simulation for clothes, muscle
  - Texture/appearance (from images)
  - Animation (mocap, simulation, artist)

Beyond 3D: Non-rigid and articulated motion



PhD work of Neil Birkbeck, Best thesis prize winner

Computer Vision

Questions?