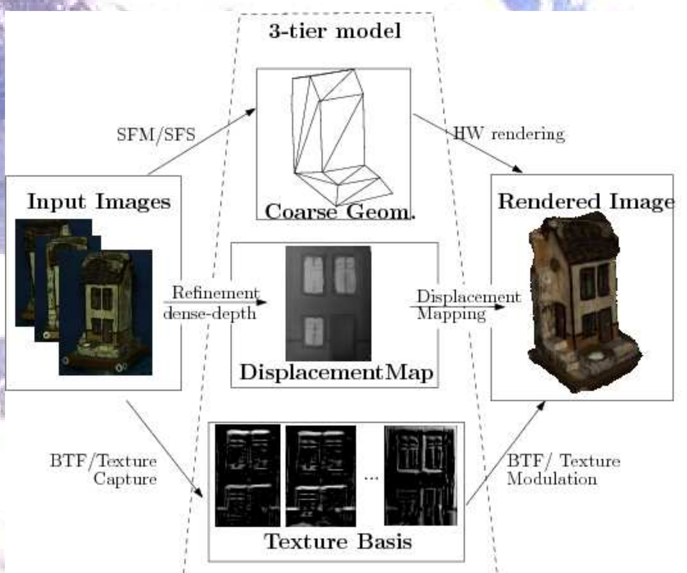
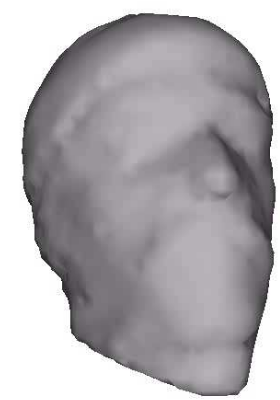
Computing models of 3D geometry and appearance for photo-realistic rendering

Martin Jagersand,





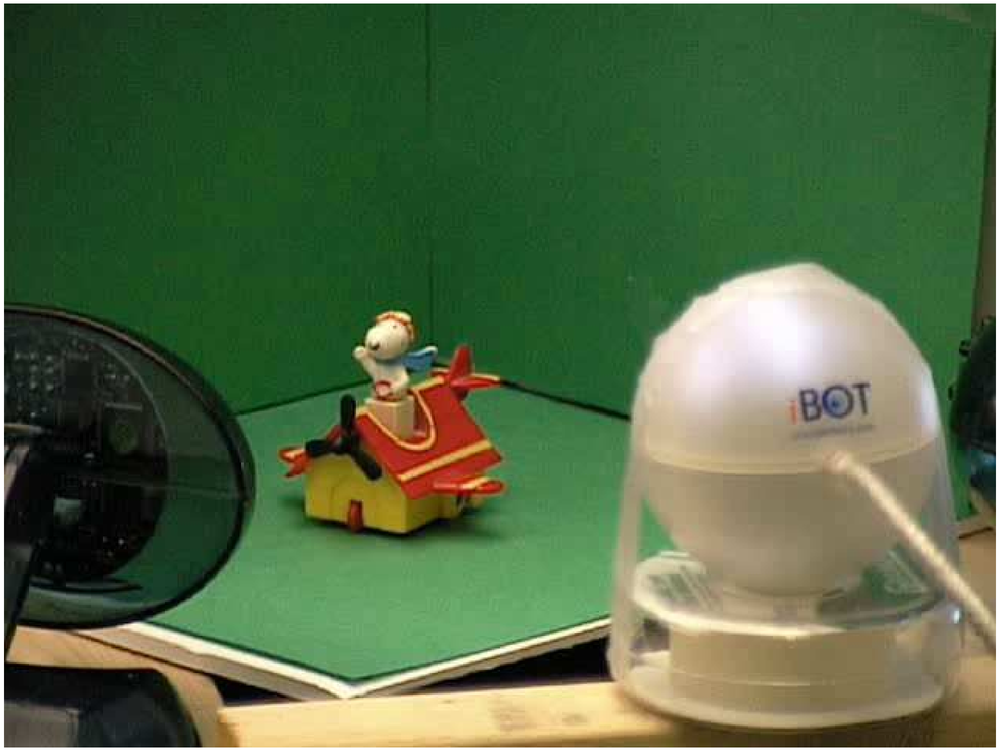
video

Low budget 3D from video example
Capture objects

- Inexpensive
- Quick and convenient for the user
- Integrates with existing SW e.g. Blender, Maya

Low budget 3D from video

- Inexpensive

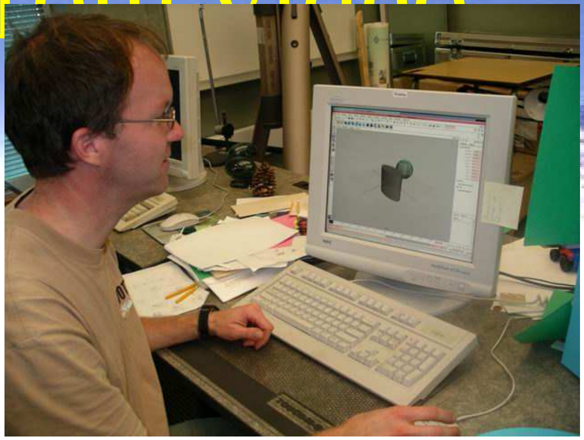


$100: Webcams, Digital Cams

$100,000 Laser scanners etc.

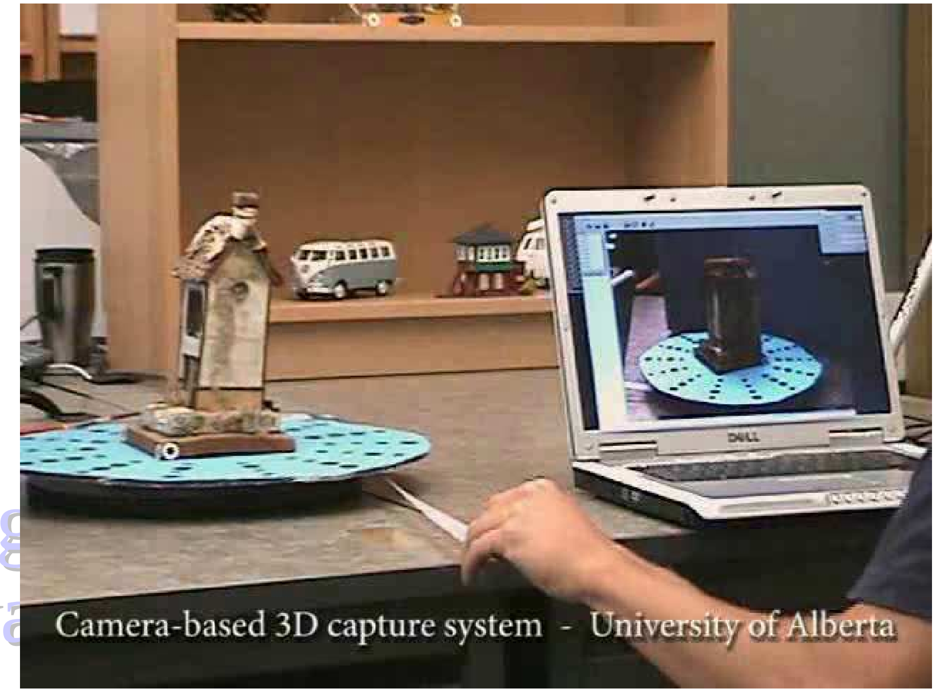
Martin Jagersand
U of Alberta

Low budget 3D from video



Modeling geom primitives into scenes: >>Hours

- Inexpensive
- Quick and convenient for the user
- Integrates with existing SW e.g. Blender, Maya



Capturing 3D from 2D video: minutes

Martin Jagersand
U of Alberta

Low budget 3D from video

- Inexpensive

- Quick and convenient for the user

- Integrates with existing SW e.g. Blender, Maya

Martin Jagersand
U of Alberta

Application Case Study
Modeling Inuit Artifacts

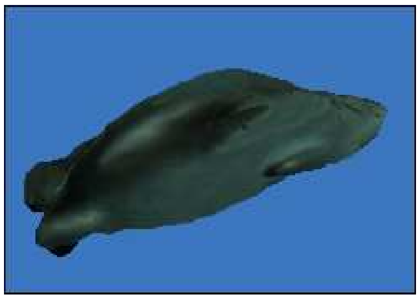
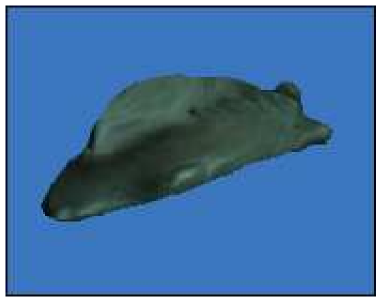
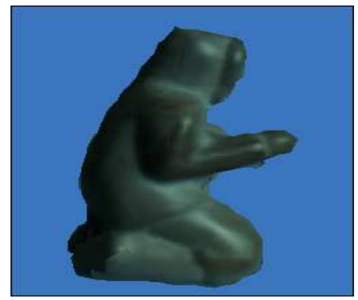
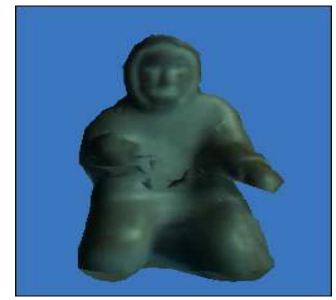
- New acquisition at the UofA: A group of 8 sculptures depicting Inuit seal hunt
- Acquired from sculptor by Hudson Bay Company

Martin Jagersand
U of Alberta

Application Case Study
Modeling Inuit Artifacts

Results:

1. A collection of 3D models of each component

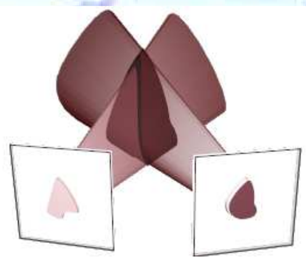





2. Assembly of the individual models into animations and Internet web study material.



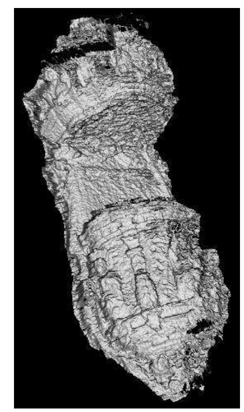

Martin Jagersand
U of Alberta

Preliminaries: Capturing **Macro** geometry:



- Shape From Silhouette
  - Works for objects
  - Robust
  - Visual hull not true object surface

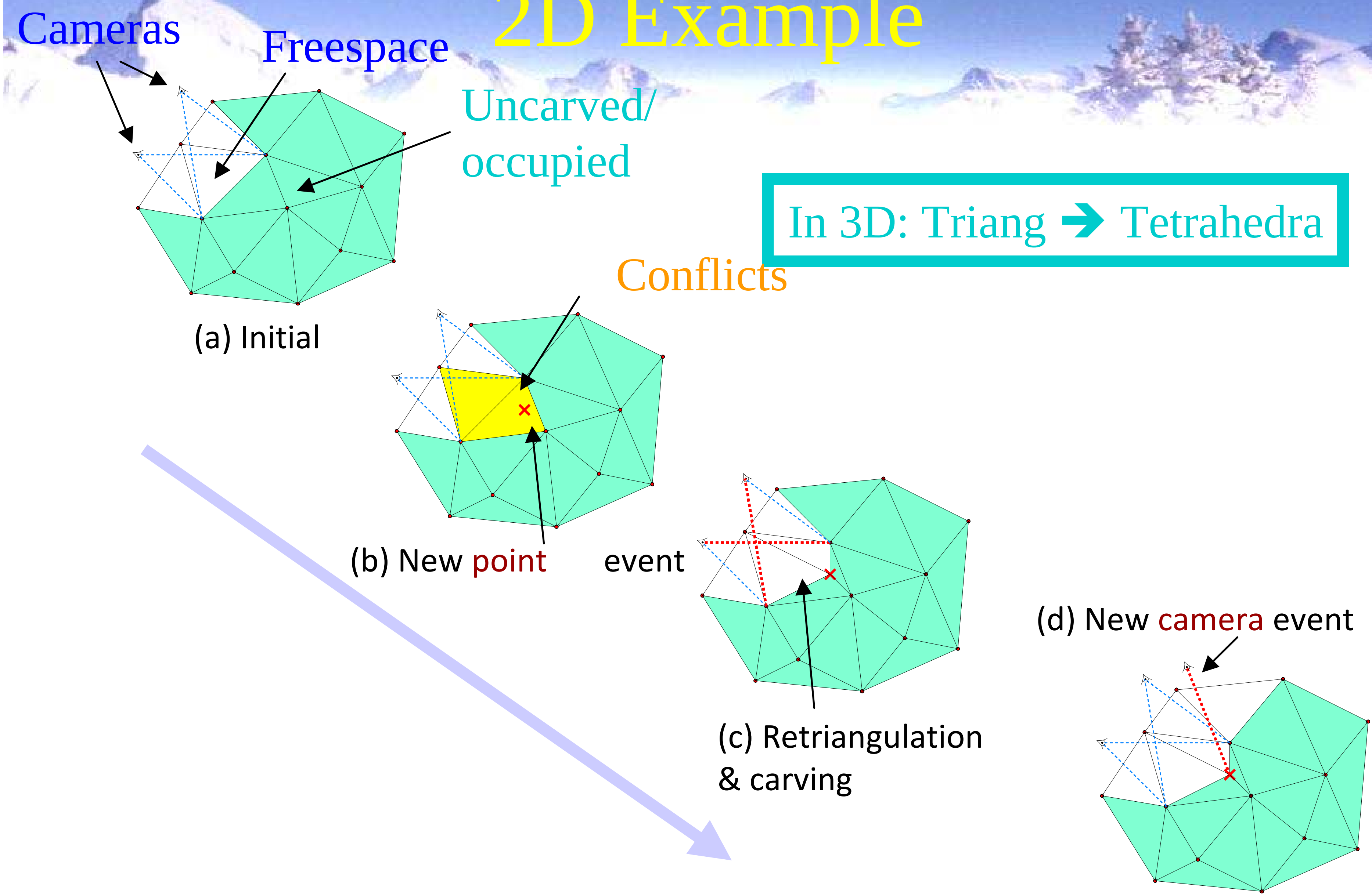
- Structure From Motion
  - Works for Scenes
  - Typically sparse
  - Sometimes fragile (no salient points in scene)
- Space carving
  - Use free space constraints



- Global Dense Refinement
  - Variational PDE
  - Use as second refinement step

Incremental Free-Space Carving 2D Example

Martin Jagersand
U of Alberta

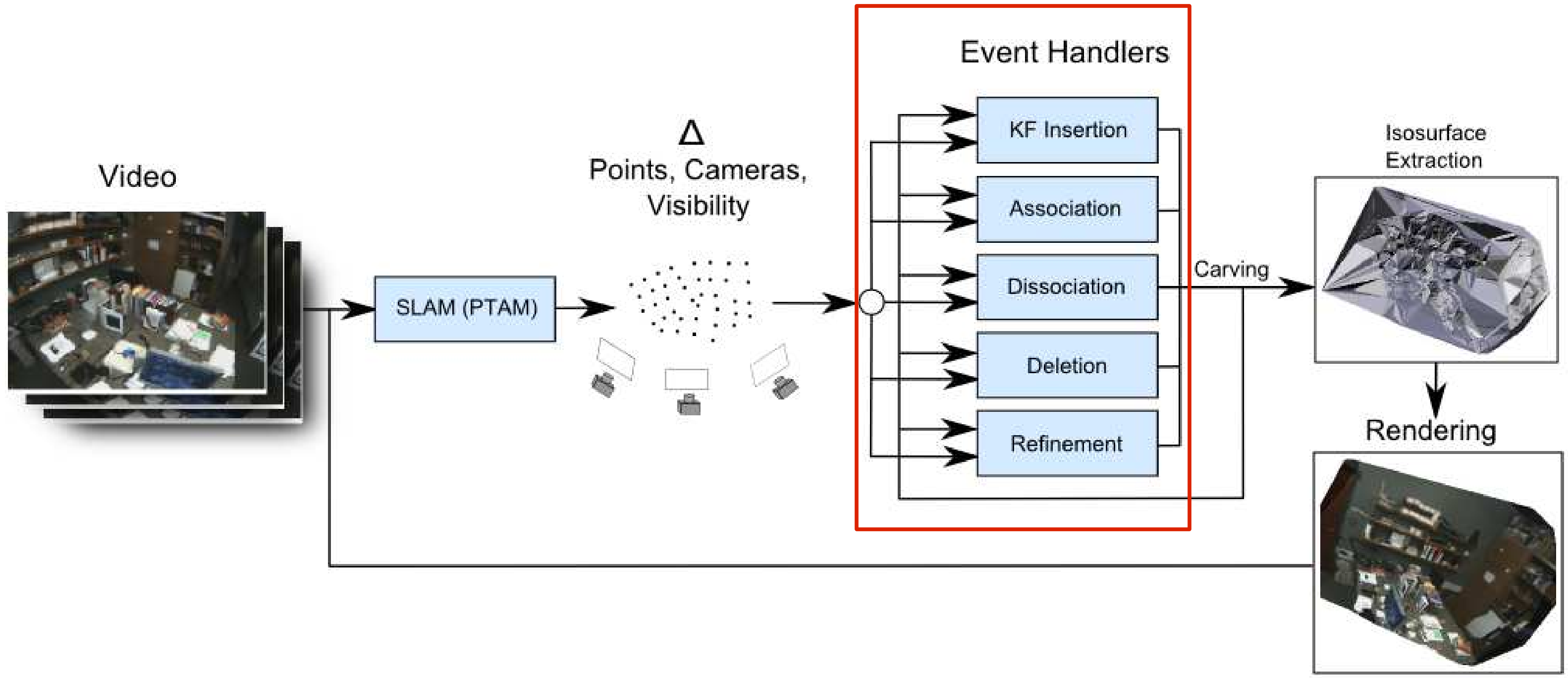


In 3D: Triang ➔ Tetrahedra

Martin Jagersand
U of Alberta

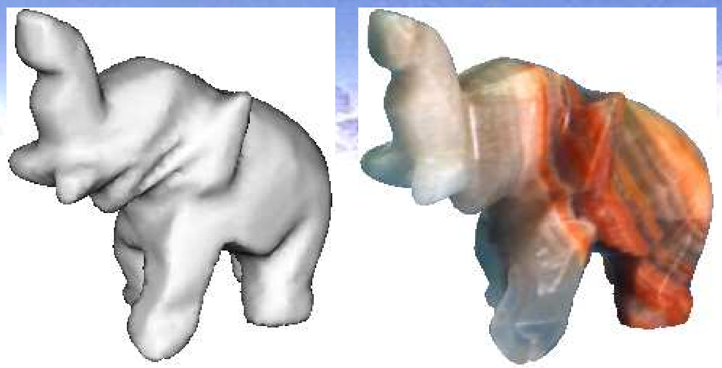
3D Modeling System

- Online, incremental handling of new-information events
  - Inputs continuously change online
  - Different types of changes trigger tailored processing

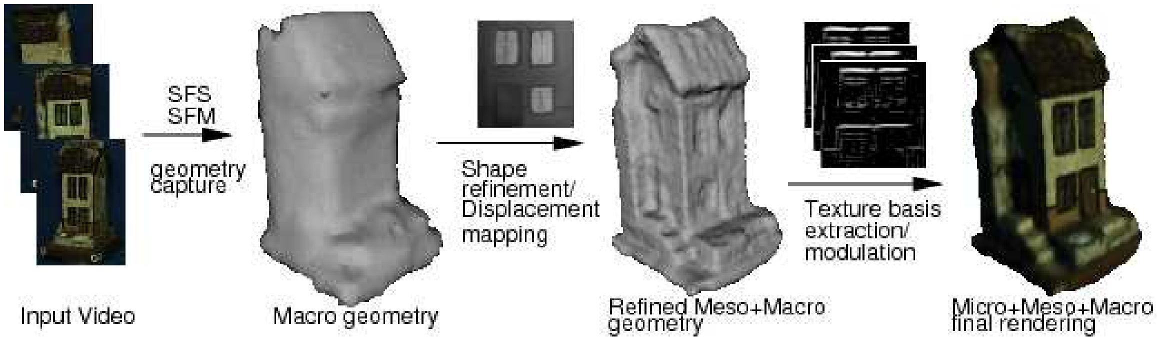


Martin Jagersand
U of Alberta

3-tier Macro, Meso, Micro model

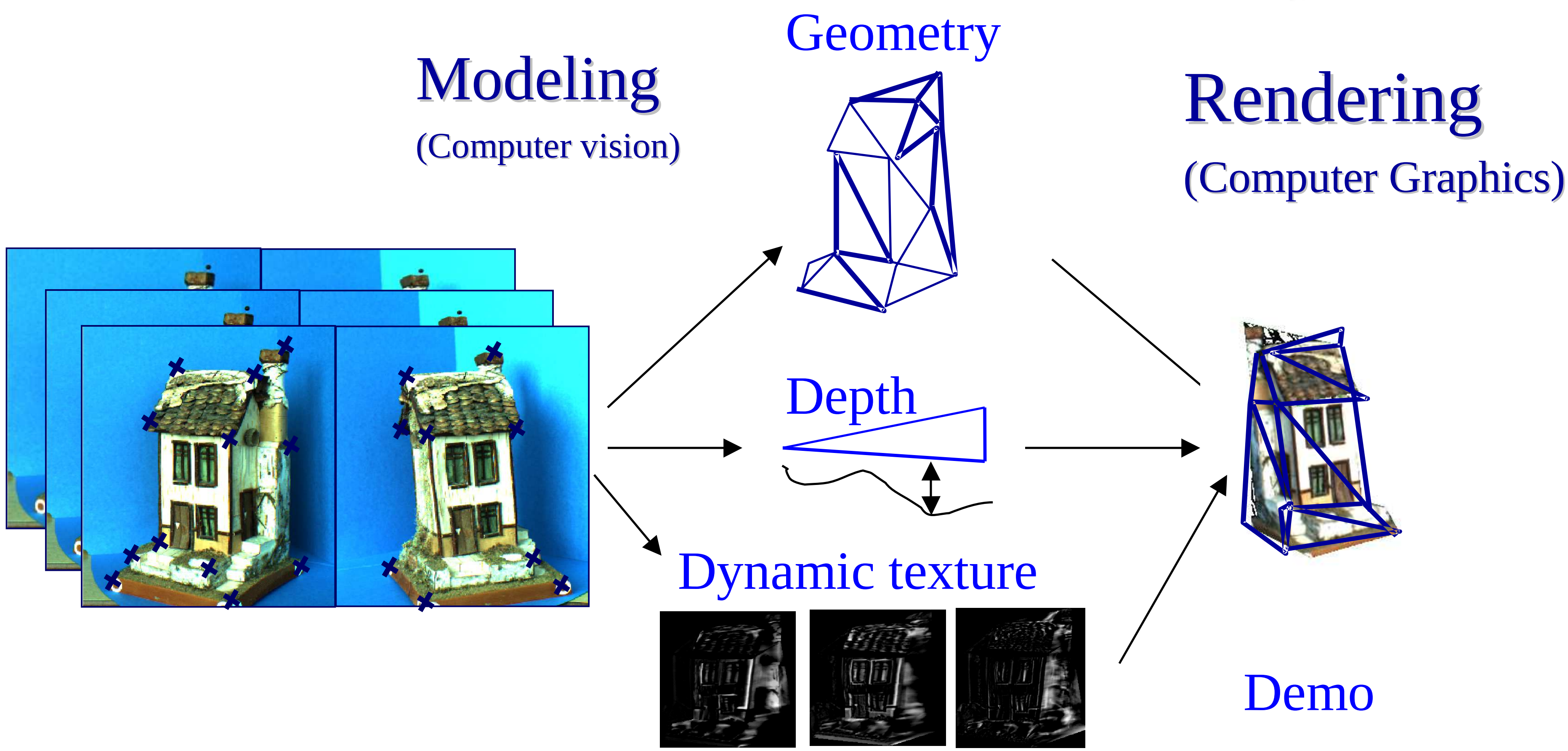


- Multi-Tiered Models:
  - Commonly:
    - Two tiers: 3D Geometry and appearance (* texture mapping)
    - Used in graphics applications, recovered in Vision applications
  - Three-Tier
    - Macro scale: describes scene geometry (triangulated mesh)
    - Meso scale: fine scale geometric detail (displacement map)
    - Micro: fine scale geometry and reflectance (Texture basis)
  - Captured by sequential refinement



Martin Jagersand
U of Alberta

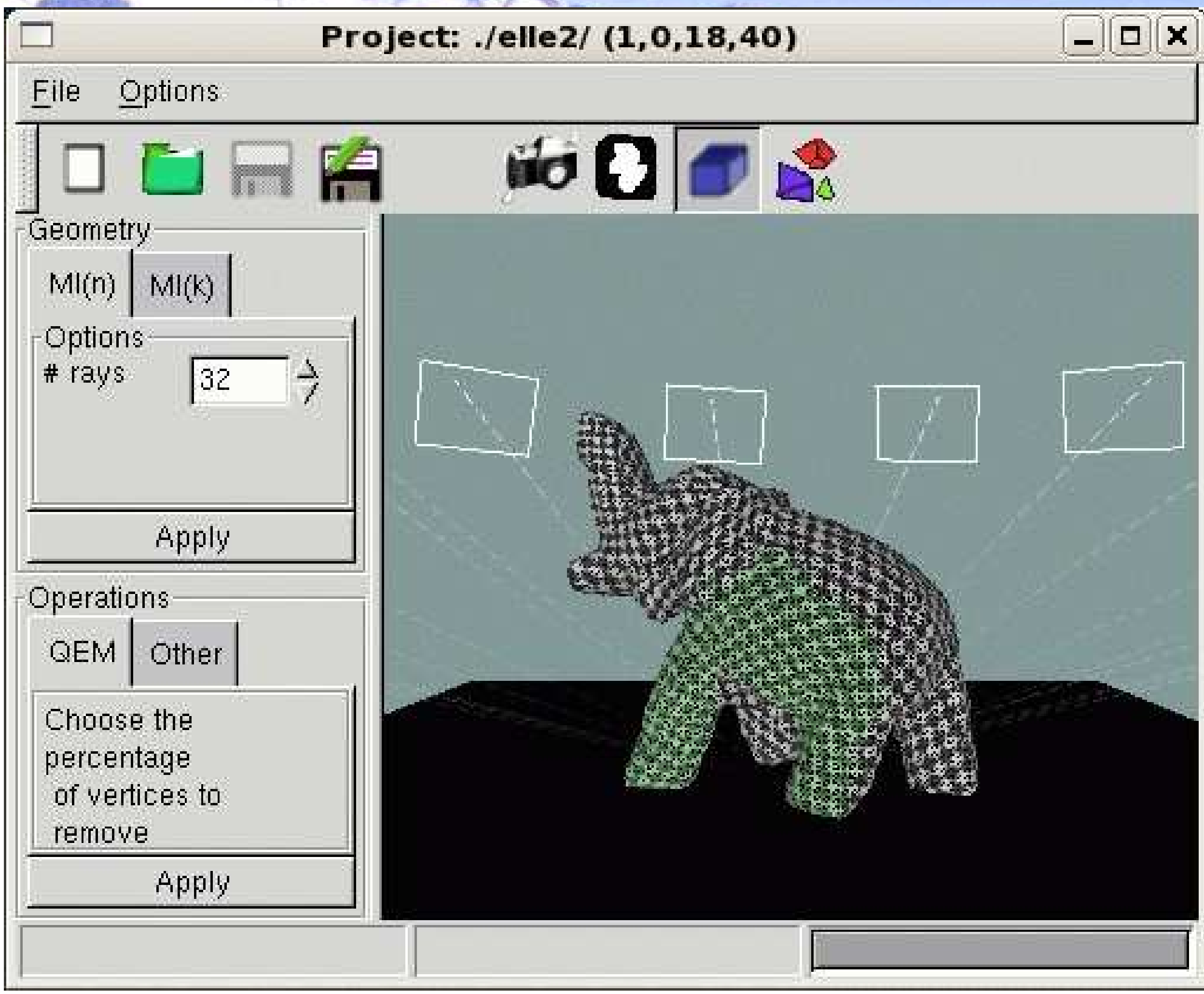
Geometry alone does not solve modeling! Need: Multi-Scale Model

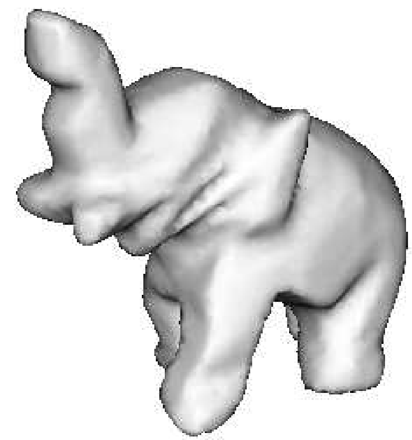


Multi-Scale model: **Macro** geometry, **Meso** depth, **Micro** texture

Martin Jagersand
U of Alberta

Capgui Demo





http://webdocs.cs.ualberta.ca/~vis/ibmr/

Martin Jagersand
U of Alberta

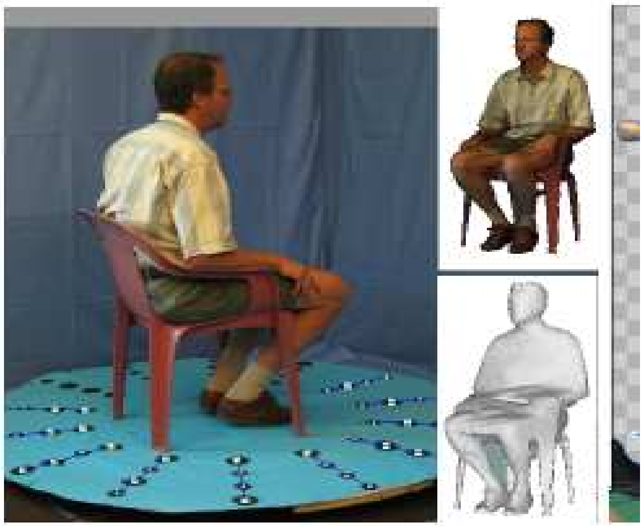
Capgui demo

Martin Jagersand
U of Alberta

Step 1 Calibration

Compute camera – object pose

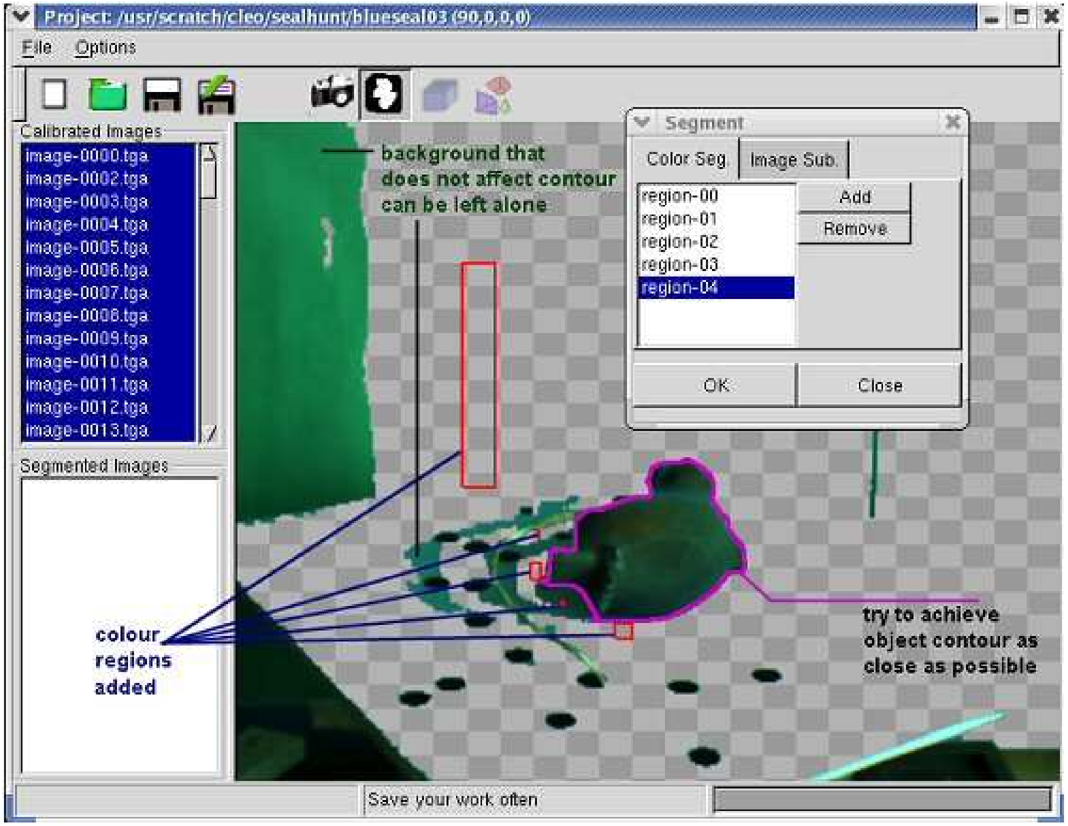
- Flat radial calib pattern
  - Rotating all-around capture
- Each row a unique code
- Crossratio projective invariant



Martin Jagersand
U of Alberta

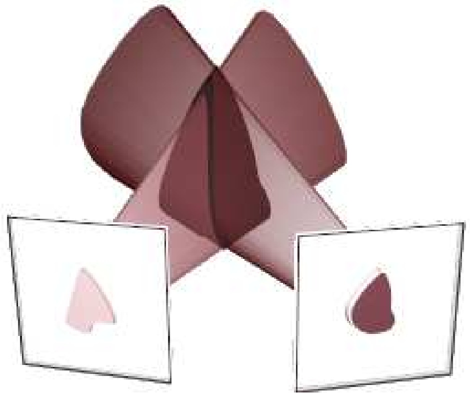
Step 2 Segmentation

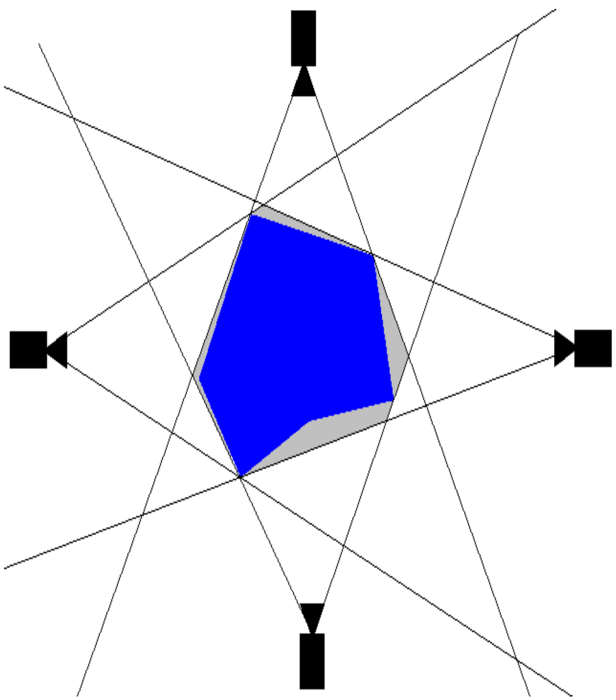
- Background removal
- PCA model
  – see tracking lectures
- One or more pixel samples



Martin Jagersand
U of Alberta

Step 3: Shape From Silhouette

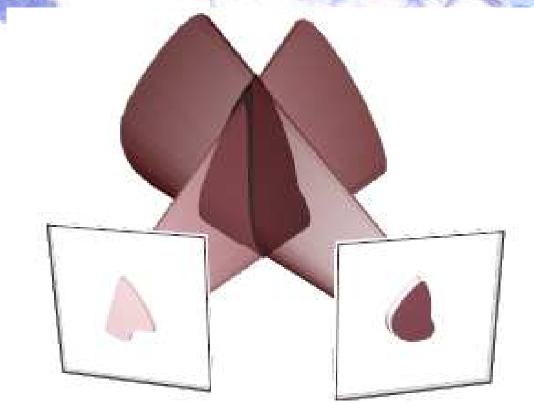


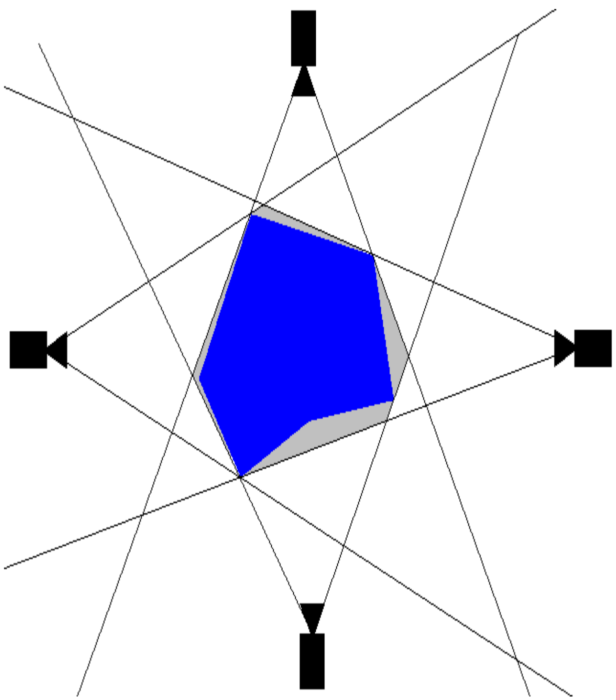


- With multiple views of the same object, we can intersect the *generalized cones* generated by each image, to build a volume which is guaranteed to contain the object.
- The limiting smallest volume obtainable in this way is known as the *visual hull* of the object.
- Use 8 - 60 images

Martin Jagersand
U of Alberta

Step 3: Shape From Silhouette



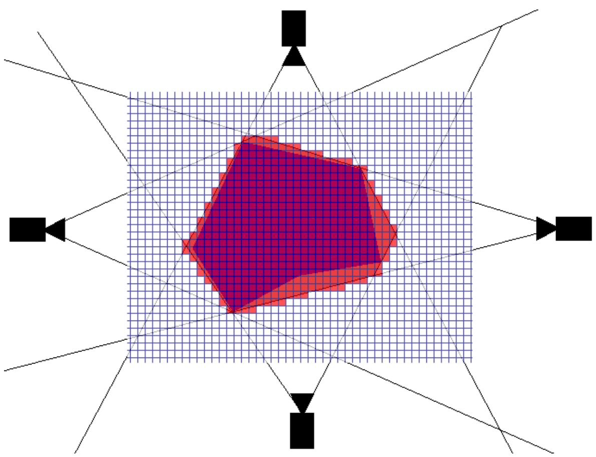


- With multiple views of the same object, we can intersect the *generalized cones* generated by each image, to build a volume which is guaranteed to contain the object.
- The limiting smallest volume obtainable in this way is known as the *visual hull* of the object.

Martin Jagersand
U of Alberta

SFS Methods

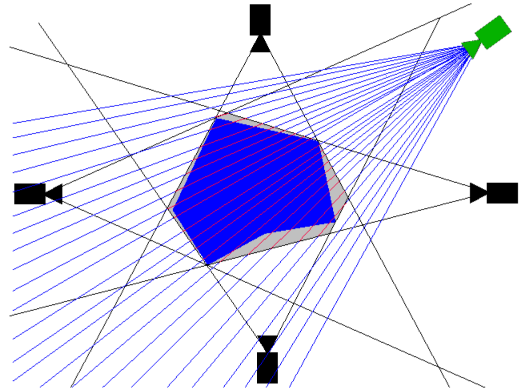
Voxel-based



- Inaccurate

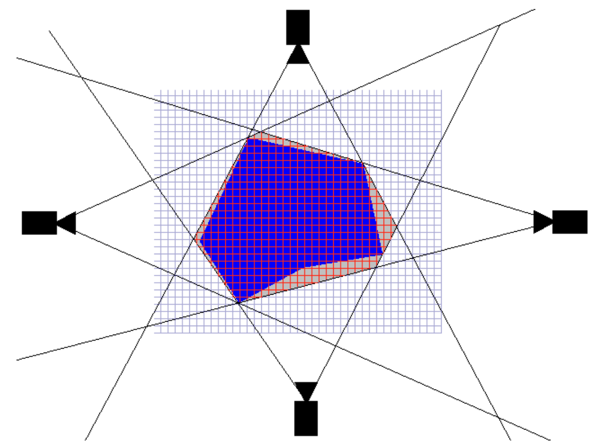
+ Triangulate w. marching cubes

Image ray based



+ Accurate

Axis-aligned



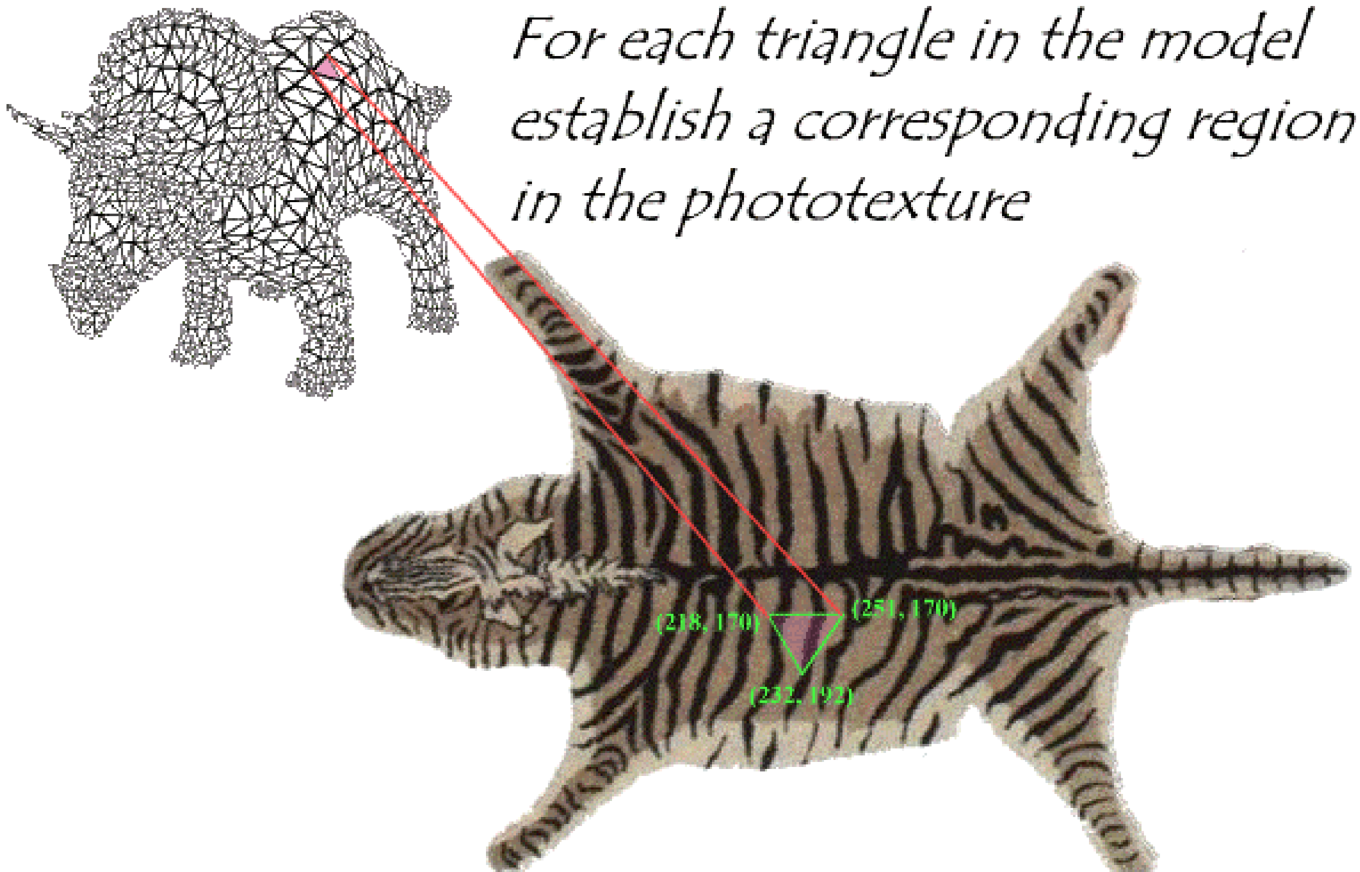
Moderatly accurate

+ Fast

+ Marching intersections

Step 4 Photo-textures and texture mapping

Martin Jagersand
U of Alberta



For each triangle in the model establish a corresponding region in the phototexture

During rasterization interpolate the coordinate indices into the texture map

Simple models can be manually flattened

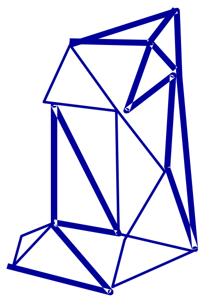
Martin Jagersand
U of Alberta

3D geometry and texture warp map between views and texture images

View

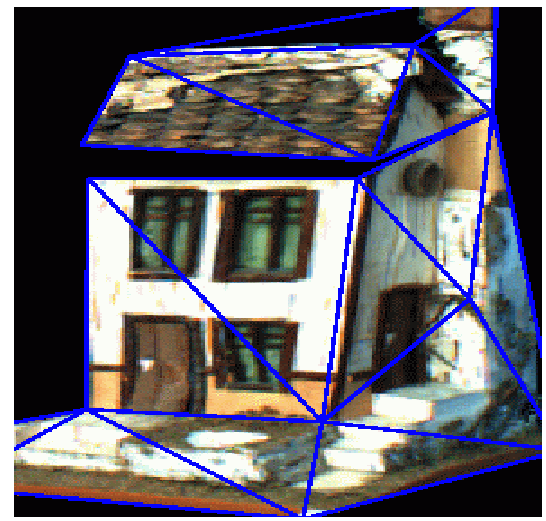
I1

Re-projected geometry



Texture coordinate warp

Texture

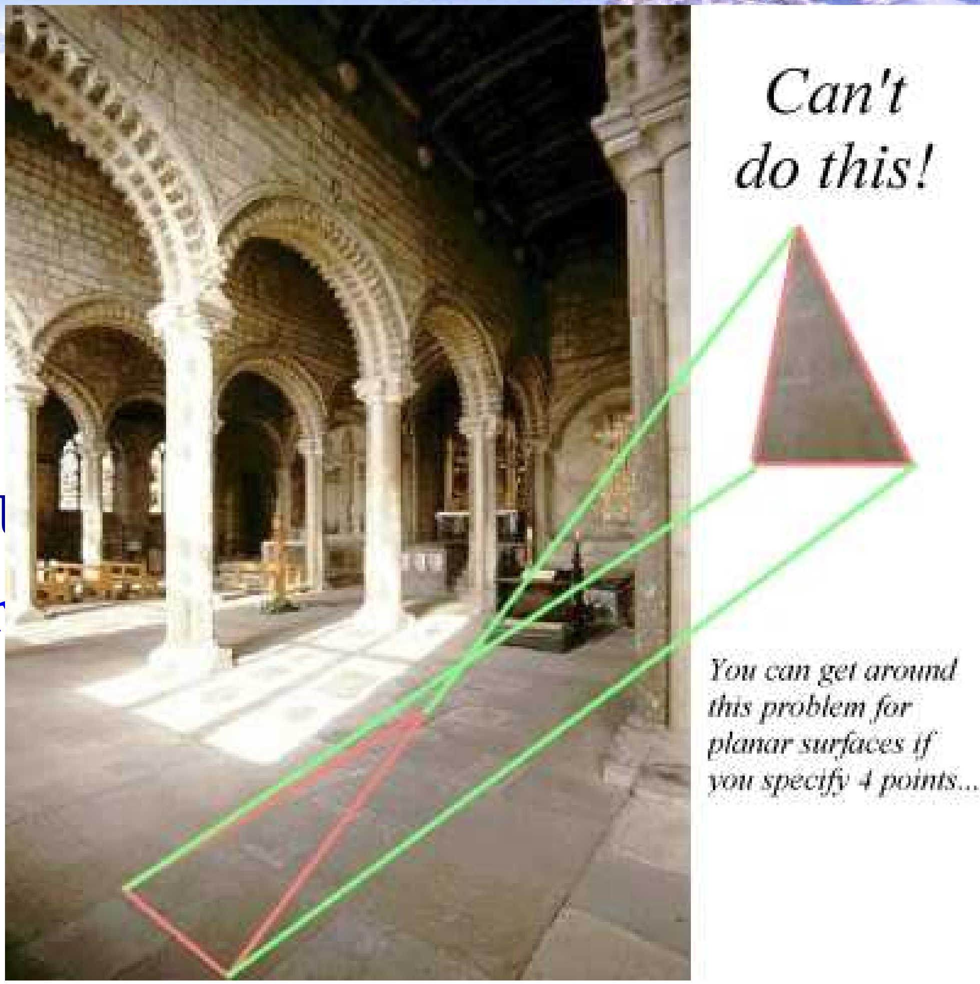


t

Martin Jagersand
U of Alberta

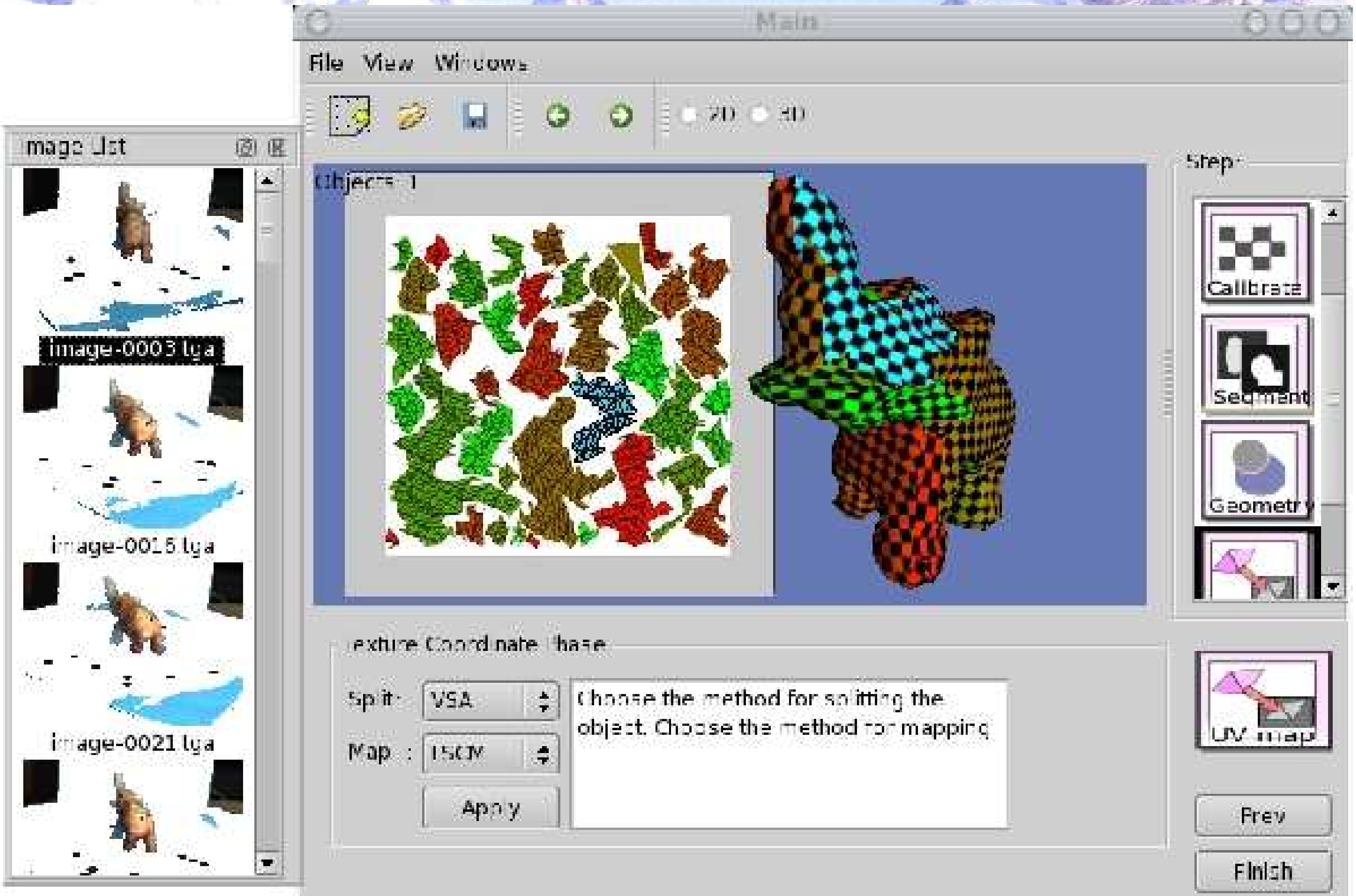
Texture Mapping Difficulties

- Tedious to specify texture coordinates for every triangle
- Acquiring textures is su
  - Texture image can't have pr
  - Seamless tiling
  - Non-repeating textures



Martin Jagersand
U of Alberta

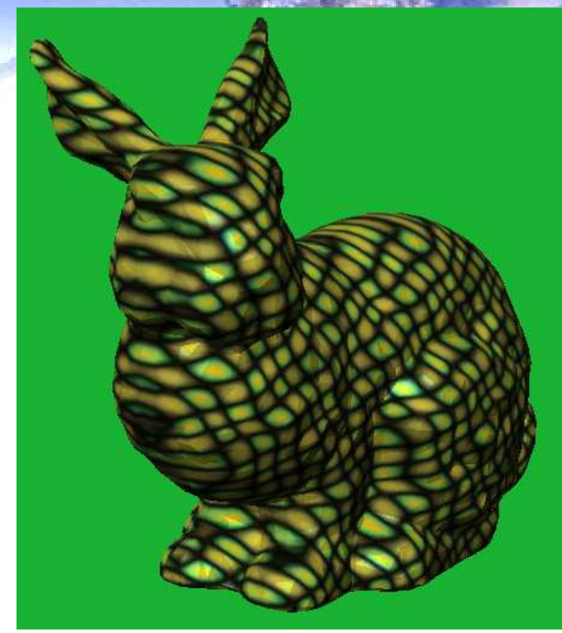
Complex, detailed models: Automatic texture split and flatten

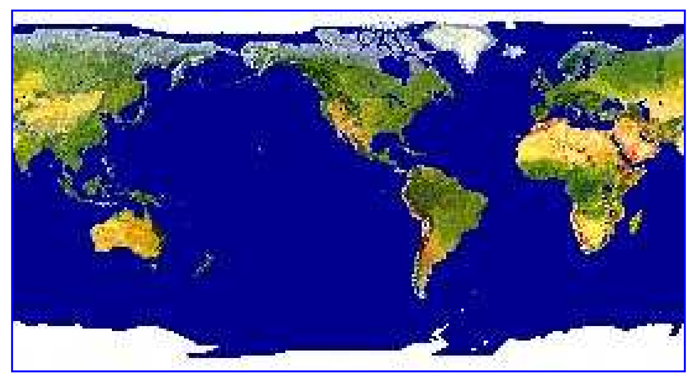


Martin Jagersand
U of Alberta

Common Texture Coordinate Mappings

- Orthogonal
- Cylindrical
- Spherical
- Perspective Projection
- Texture Chart

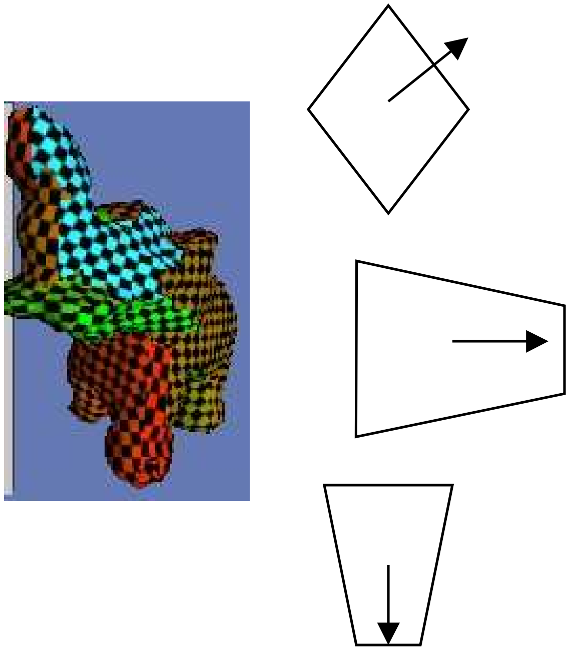




Advanced texture splitting and mapping

Martin Jagersand
U of Alberta

- Floating Planes (Mathieu Desbrun, INRIA, CalTech)

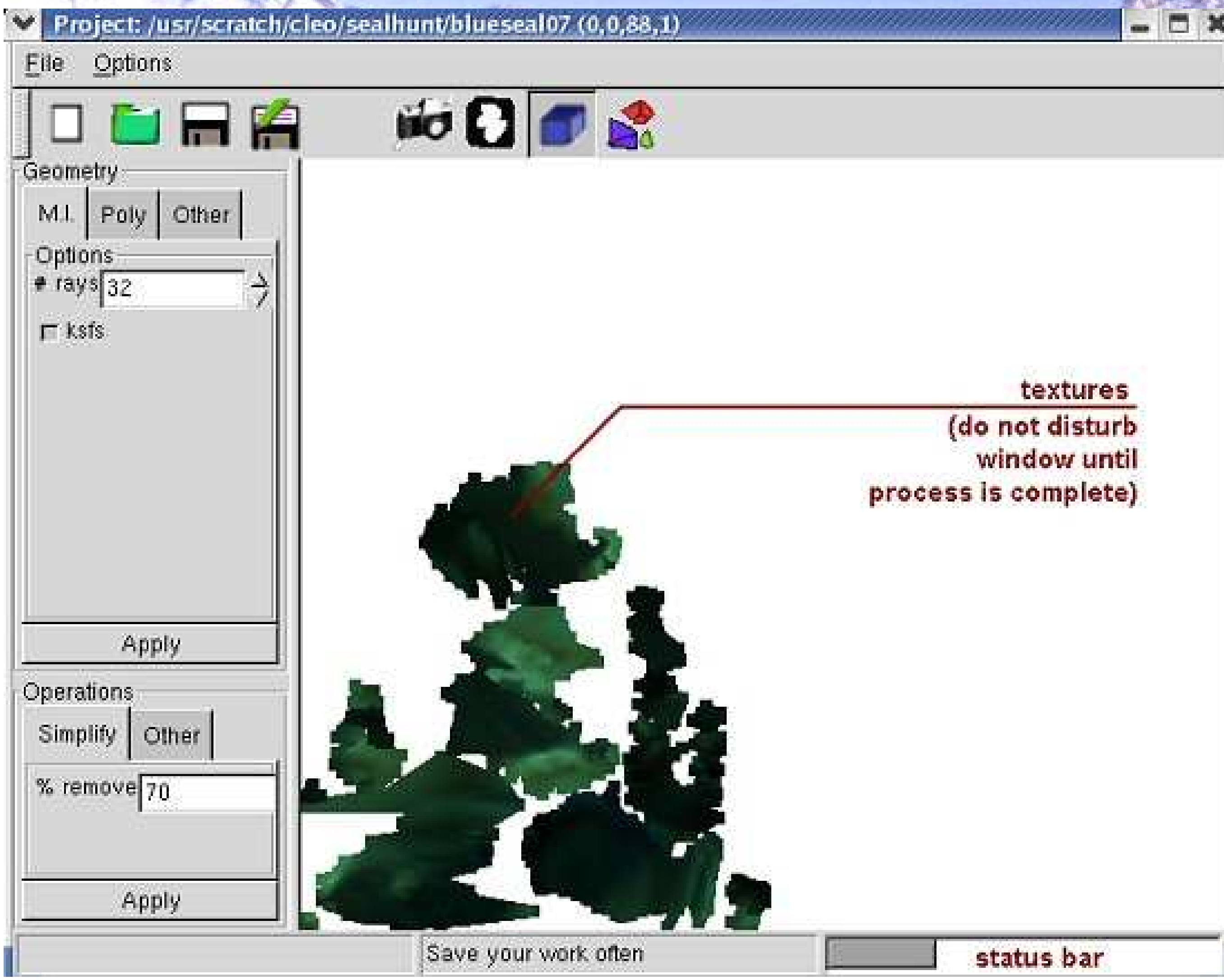


- LCSM: Least Squares Conformal Mapping

Martin Jagersand
U of Alberta

Step 6 Texture basis computation

•



https://webdocs.cs.ualberta.ca/~vis/ibmr/capgui/guide.html

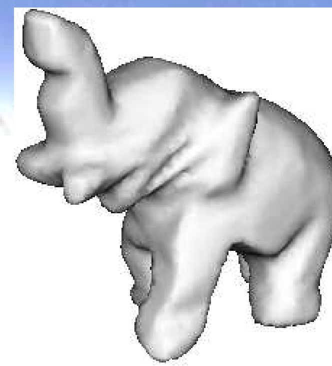
Martin Jagersand
U of Alberta

Dyntext theory

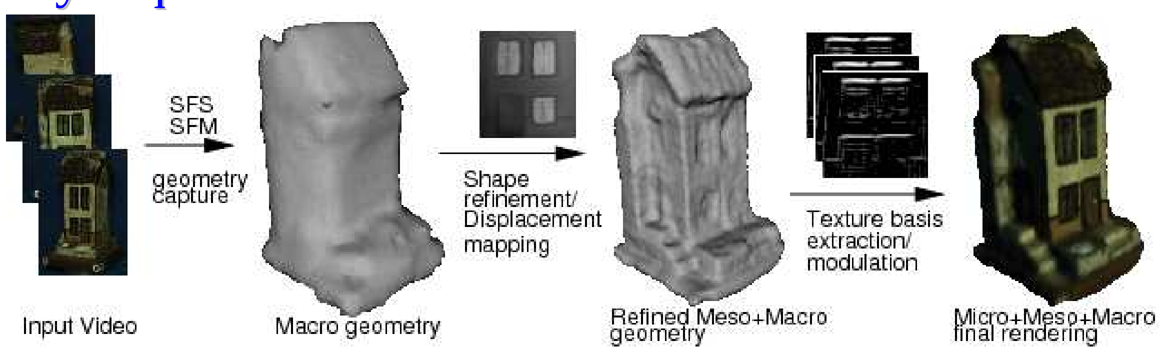
- Dyntext slides start here

Martin Jagersand
U of Alberta

3-tier Macro, Meso, Micro model

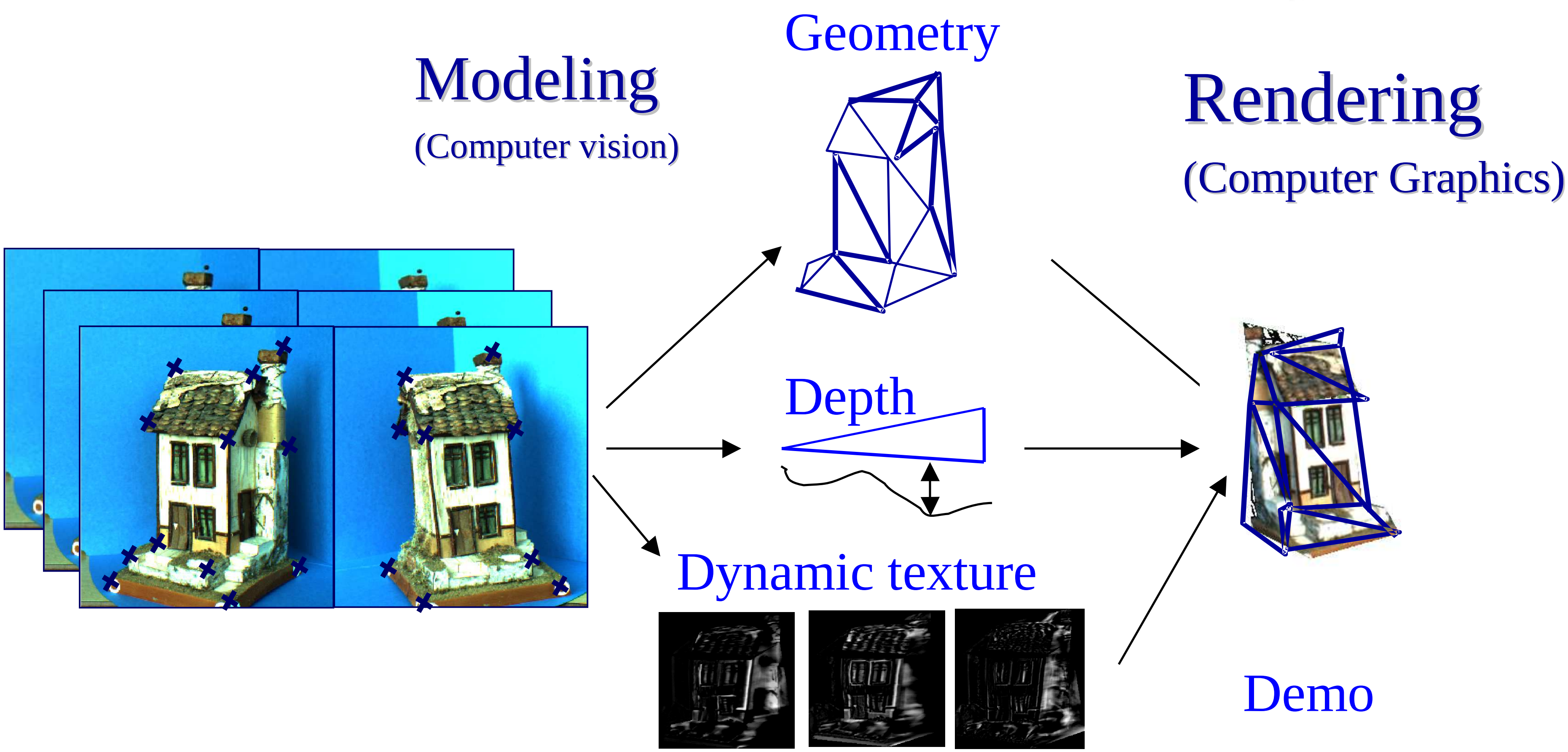


- Multi-Tiered Models:
  - Commonly:
    - Two tiers: 3D Geometry and appearance (* texture mapping)
    - Used in graphics applications, recovered in Vision applications
  - Three-Tier
    - Macro scale: describes scene geometry (triangulated mesh)
    - Meso scale: fine scale geometric detail (displacement map)
    - Micro: fine scale geometry and reflectance (Texture basis)
  - Captured by sequential refinement



Martin Jagersand
U of Alberta

Geometry alone does not solve modeling! Need: Multi-Scale Model



Multi-Scale model: **Macro** geometry, **Meso** depth, **Micro** texture

Martin Jagersand
U of Alberta

Three scales map naturally to CPU and GPU hardware layers

Key issue: Efficient memory access and processing

Speedup

1. Macro: Conventional geometry processing

10x

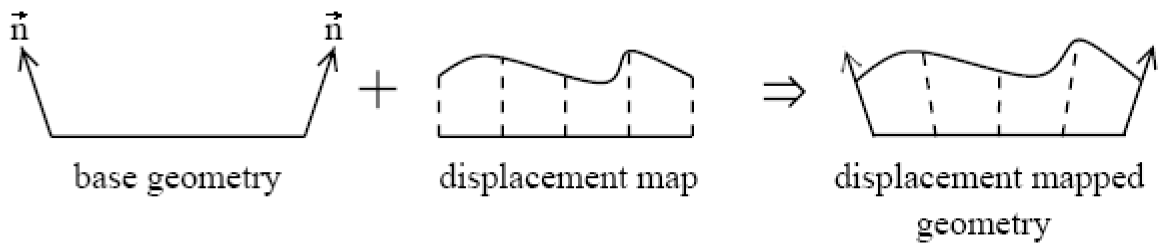
2. Meso: Pixel shader
   – Fixed code, variable data access

10x

3. Micro: Shader or Register comb.
   – Fixed code, fixed data access

Martin Jagersand
U of Alberta

2. Meso Structure: Depth with respect to a plane



Flat texture

Displacement mapped

Computing **Meso** structure: Variational shape and reflectance

Martin Jagersand
U of Alberta

Per-point cost function

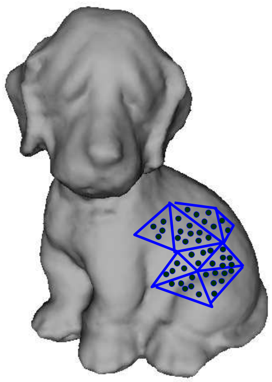
$$\Phi(\mathbf{X},\mathbf{n}) = \sum_i h(\mathbf{X}, P_i)\left\| I_i(P_i(\mathbf{X})) - R(\mathbf{X},\mathbf{n},\mathbf{L}_i)\right\|$$

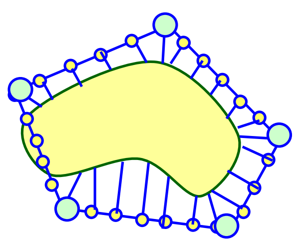
Visibility+sampling

reflectance

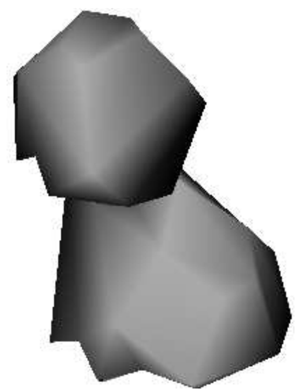
$$\frac{\partial S}{\partial t} = \left(2\Phi k - \langle \nabla\Phi, \mathbf{n}\rangle\right)\mathbf{n}$$

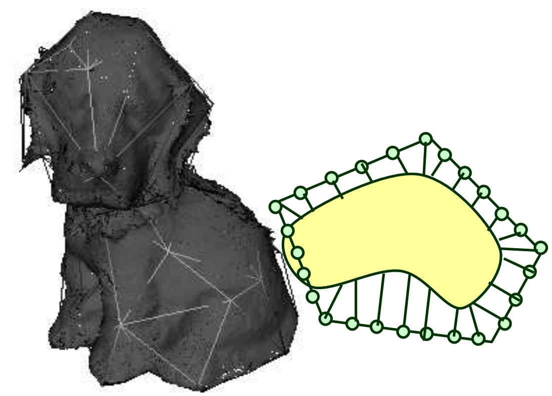
Deformable mesh





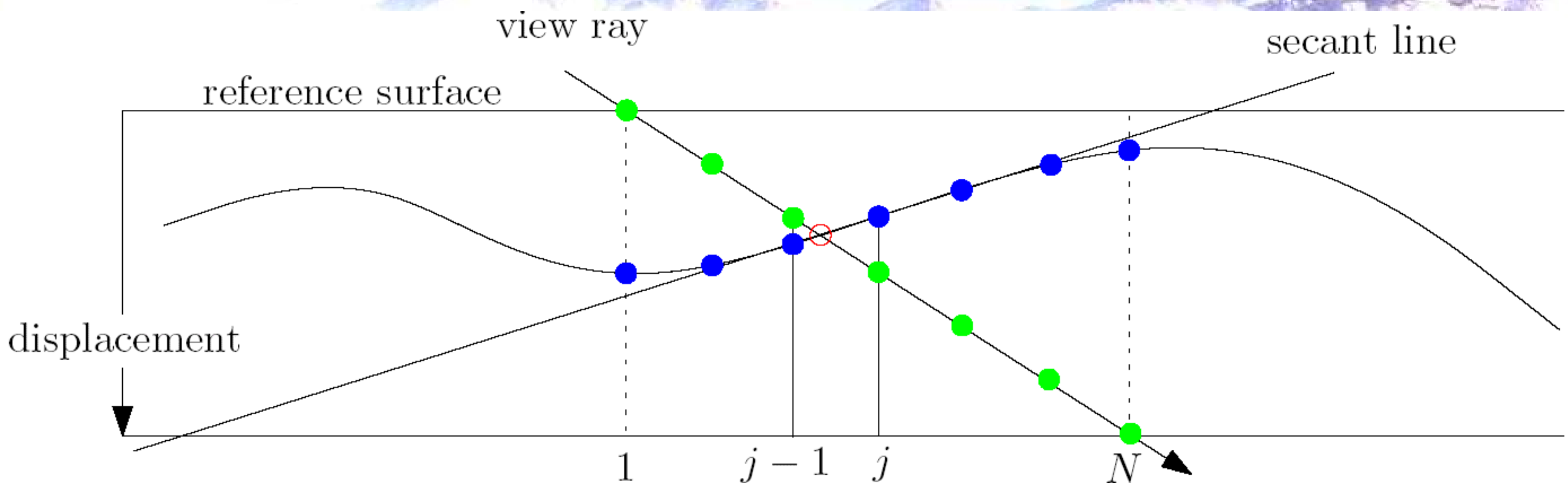
Depth from Base





Martin Jagersand
U of Alberta

Rendering **Meso** Structure:
GPU: 83 pixel shader instructions



1. Sample d and ray at N (say15) points.
2. Find point location j of intersection
3. Approximate d with line, calculate intersection
4. Potentially iterate if needed for accuracy

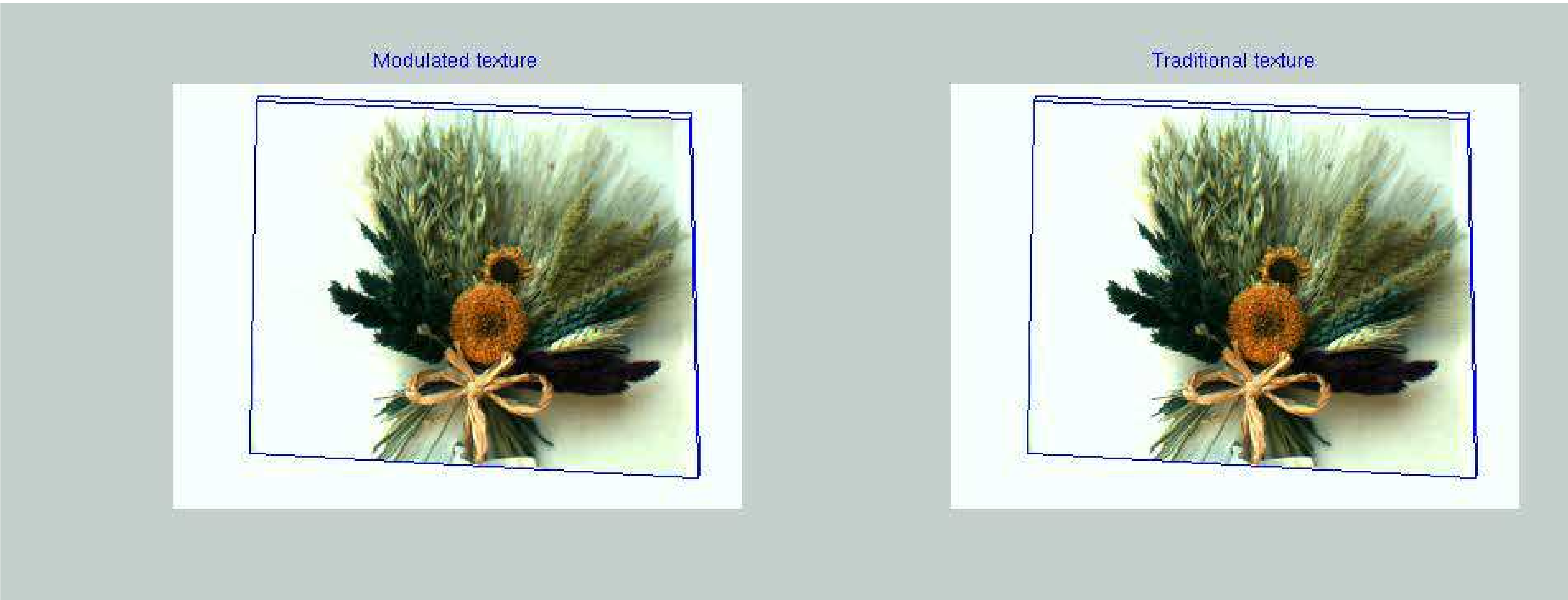
Martin Jagersand
U of Alberta

Results:

Over 100 fps on consumer graphics cards

Martin Jagersand
U of Alberta

3. **Micro** structure: Spatial texture basis



=> fixed execution and data access pattern

=> very fast implementation in graphics hardware

Martin Jagersand
U of Alberta

How/why do dynamic textures work?

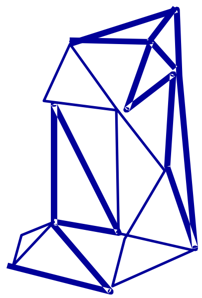
3D geometry and texture warp map between views and texture images

View

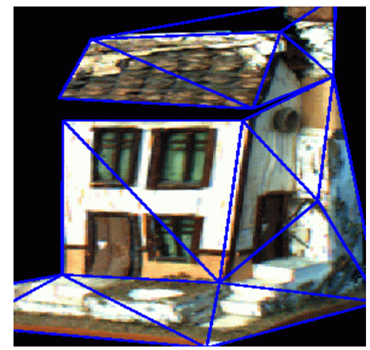
Re-projected geometry

Texture

I_1

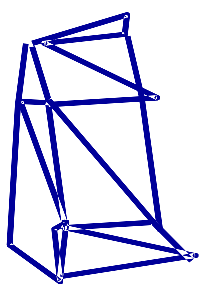


Texture warp



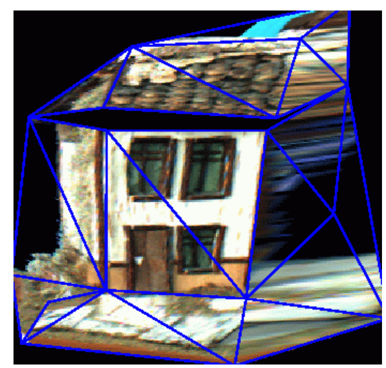
≠

I^t



Problem:

Texture images different



Martin Jagersand
U of Alberta

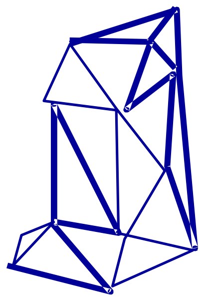
Sources of errors:

3D geometry and texture warp map between views and texture images

View

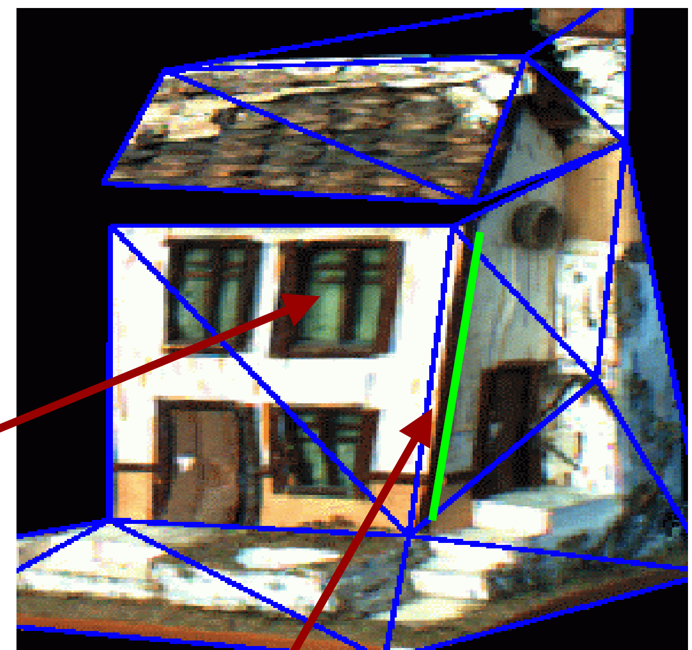
I_1

Re-projected geometry



Texture warp

Texture



2: Out of plane error:

Object surface /= texture plane

t

1: Planar error: Incorrect texture coordinates

Martin Jagersand
U of Alberta

Spatial basis intro

1. Moving sine wave can be modeled:

$$\begin{aligned} I(t) &= \sin(u + at) \\ &= \sin(u)\cos(at) + \cos(u)\sin(at) \\ &= \sin(u)y_1(t) + cos(u)y_2(t) \end{aligned}$$

Spatially fixed basis

2. Small image motion

$$I = I_0 + \frac{\partial I}{\partial u}\Delta u + \frac{\partial I}{\partial v}\Delta v$$

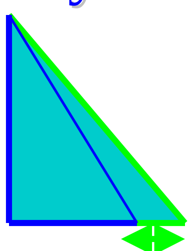
Spatially fixed basis

Martin Jagersand
U of Alberta

Linear basis for spatio-temporal variation

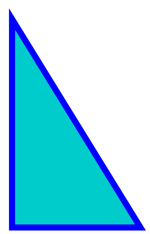
On the object/texture plane:

- Variation resulting from small warp perturbations
- Taylor expansion:

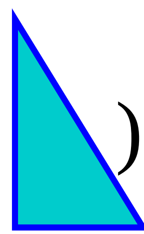


$\Delta\mu$

$=$



$+\frac{\partial}{\partial\mu}($



$)\Delta\mu + h.o.t.$

Small if $\Delta\mu$ small and T_0 smooth

$$T(view) = T_0 + \underbrace{\frac{\partial}{\partial\mu} T_0}\Delta\mu + h.o.t.$$

Similarly: Can derive linear basis for out of plane and light variation!

Geometric spatio-temporal variability

Martin Jagersand
U of Alberta

Image "warp"

$$T(\mathbf{x}) = I(W(\mathbf{x}, \mu))$$

Image variability caused by an imperfect warp

$$\Delta T = I(W(\mathbf{x}, \mu + \Delta\mu)) - T_w$$

First order approximation

$$\Delta T = I(W(\mathbf{x}, \mu)) + \nabla T \frac{\partial W}{\partial \mu} - T_w = \nabla T \frac{\partial W}{\partial \mu}$$

Concrete examples

- Image plane
- Out of plane

Martin Jagersand
U of Alberta

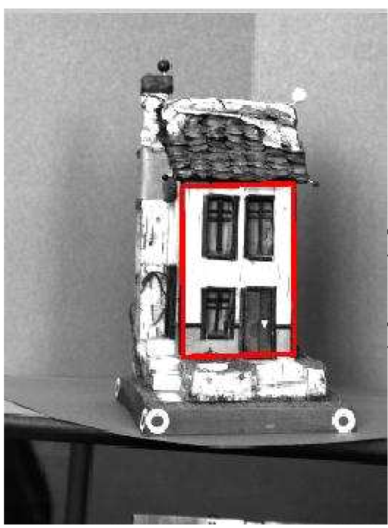
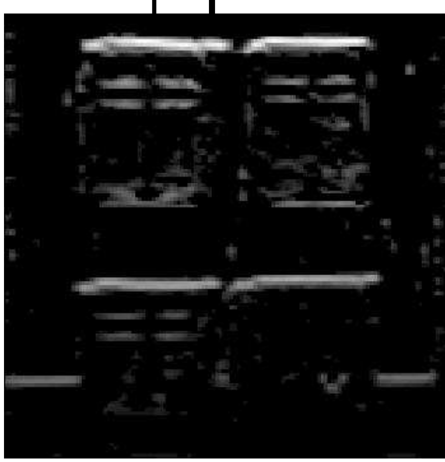
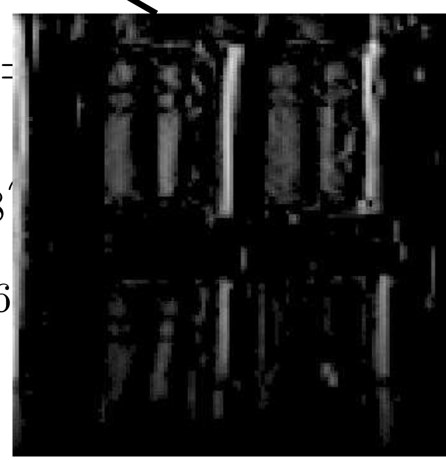
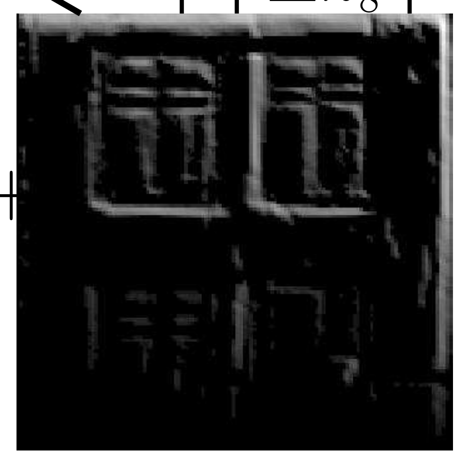
Variability due to a planar projective warp (homography)

- Homography warp

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \mathcal{W}_h(\mathbf{x}_h, \mathbf{h}) = \frac{1}{1+h_7u+h_8v}\begin{bmatrix} h_1u & h_3v & h_5 \\ h_2u & h_4v & h_6 \end{bmatrix}$$

- Projective variability:

$$\Delta\mathbf{T}_h = \frac{1}{c_1}\left[\frac{\partial\mathbf{T}}{\partial u}, \frac{\partial\mathbf{T}}{\partial v}\right]\begin{bmatrix} u & 0 & v & 0 & 1 & 0 & -\frac{uc_2}{c_1} & -\frac{vc_2}{c_1} \\ 0 & u & 0 & v & 0 & 1 & -\frac{uc_3}{c_1} & -\frac{vc_3}{c_1} \end{bmatrix}\begin{bmatrix} \Delta h_1 \\ \vdots \\ \Delta h_8 \end{bmatrix}$$

Martin Jagersand
U of Alberta

Out-of-plane variability

- Let $r = [\alpha, \beta]$ angle for ray to scene point
- Pre-warp texture plane rearrangement:

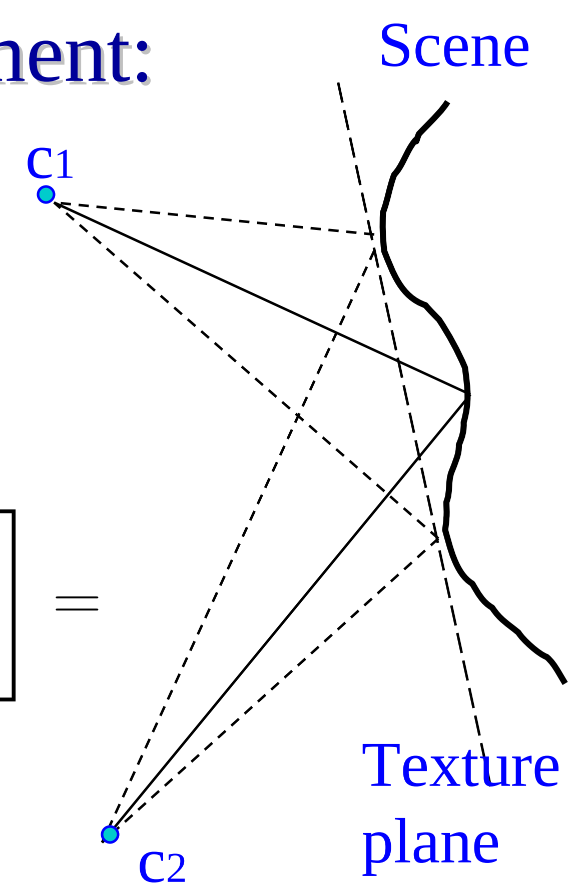
$$\begin{bmatrix} \delta u \\ \delta v \end{bmatrix} = \mathcal{W}_p(\mathbf{x}, \mathbf{d}) = \mathbf{d}(\mathbf{u}, \mathbf{v}) \begin{bmatrix} \tan\alpha \\ \tan\beta \end{bmatrix}$$

Depth w.r.t. model facet

- Texture basis

$$\Delta \mathbf{T_p} = \mathbf{d}(\mathbf{u}, \mathbf{v}) \left[\frac{\partial \mathbf{T}}{\partial \mathbf{u}}, \frac{\partial \mathbf{T}}{\partial \mathbf{v}}\right] \begin{bmatrix} \frac{1}{\cos^2 \alpha} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\cos^2 \beta} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\alpha \\ \boldsymbol{\Delta}\beta \end{bmatrix} =$$

$$= \mathbf{B_p} \mathbf{y_p}$$



Martin Jagersand
U of Alberta

Photometric variation

Analytic formula for irradiance for a convex Lambertian object under distant illumination (with attached shadows) - spherical harmonics

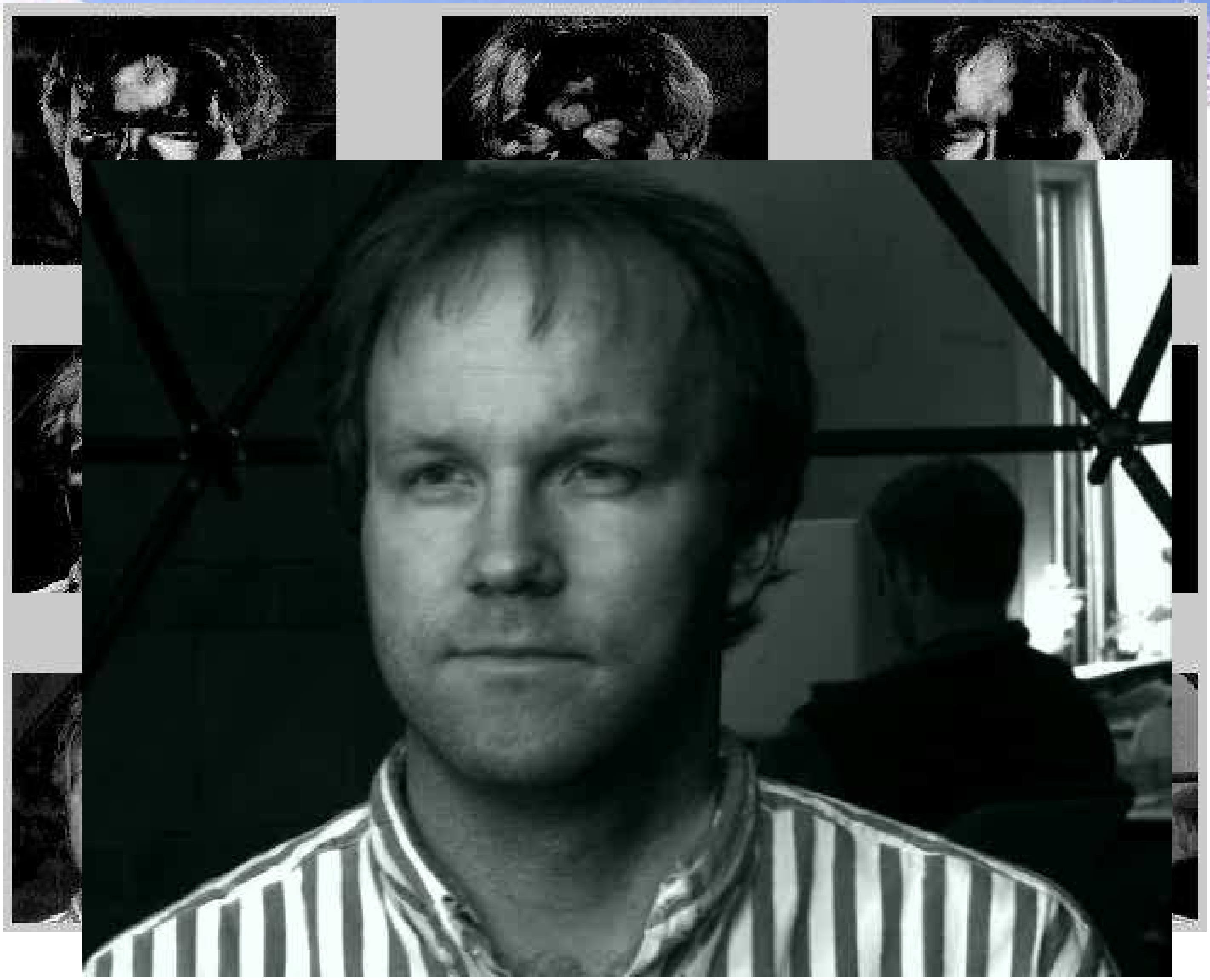
[Barsi and Jacobs, Ramamoorthi and Hanrahan 2001]

$$T(\alpha,\beta,\theta,\phi) \approx \sum_{l=0}^{2} \sum_{k=-l}^{l} L_{lk}(\alpha,\beta) A_l Y_{lk}(\theta,\phi)$$

$$T = [B_1 \cdots B_9][L_1 \cdots L_9]^T$$

Martin Jagersand
U of Alberta

Example of photometric variation



Composite variability

Similarly, composite texture intensity variability

$$\Delta \mathbf{T} = \Delta \mathbf{T}_s + \Delta \mathbf{T}_d + \Delta \mathbf{T}_l + \Delta \mathbf{T}_e$$

Planar Depth Light Res Err

Can be modeled as sum of basis

$$\Delta \mathbf{T} = \mathbf{B_s y_s} + \mathbf{B_d y_d} + \mathbf{B_l y_l} + \mathbf{\Delta T_e}$$

$$= \mathbf{By} + \mathbf{\Delta T_e}$$

How to compute?

From a 3D graphics model:

1. Texture intensity derivatives
2. Jacobian of warp or displacement function

- Results in about 20 components:
  - T_0
  - 8 for planar,
  - 2 out-of plane (parallax),
  - 3-9 light

From video:

- We can expect an approximately 20dim variation in the space of all input texture images.

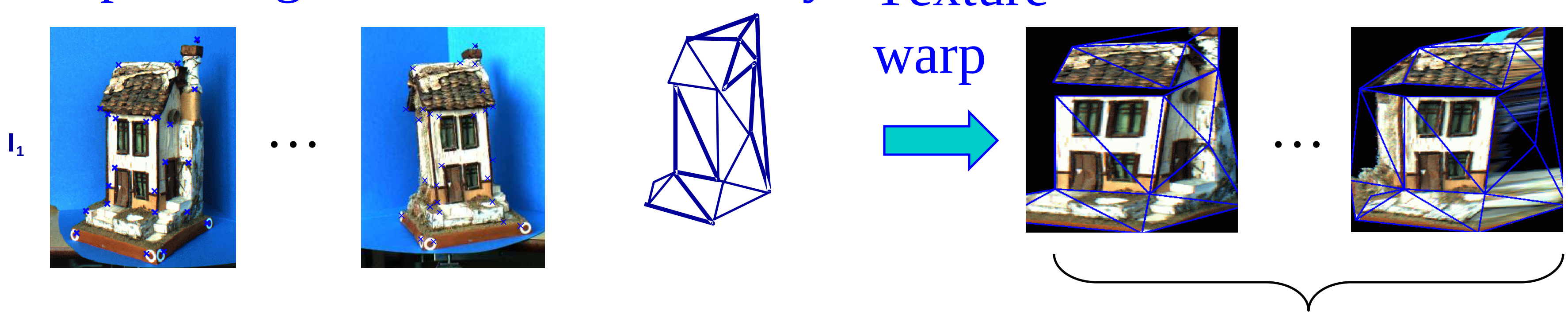
=> Extract this subspace

How to compute from images (cont)…

Martin Jagersand
U of Alberta

1. Take input video sequence, use SFS/SFM geometry to warp into texture space

Input Images   Geometry   Texture warp



PCA

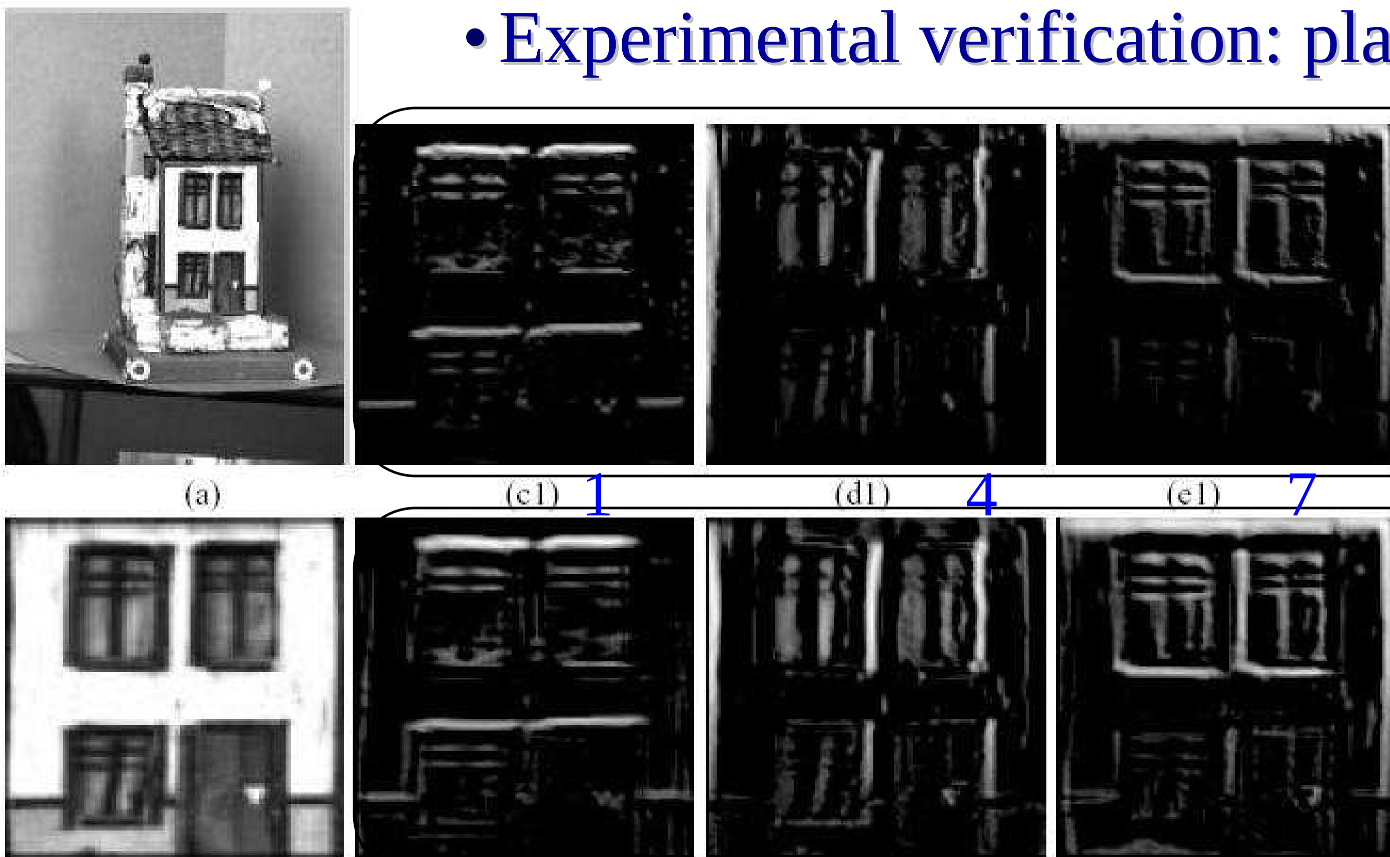
2. Extract a 20-dim subspace through PCA

TexDemo

Martin Jagersand
U of Alberta

Are analytic image derivatives and PCA basis the same?

- Same up to a linear transform!



- Experimental verification: planar homog

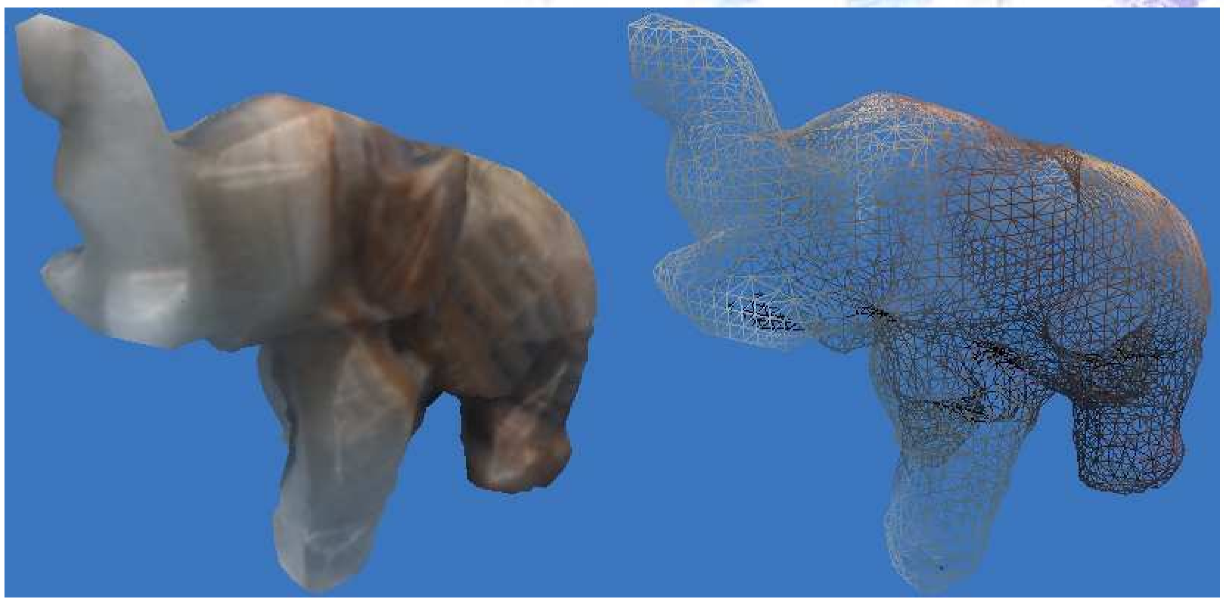
Derivatives

99% agreement

PCA

Martin Jagersand
U of Alberta

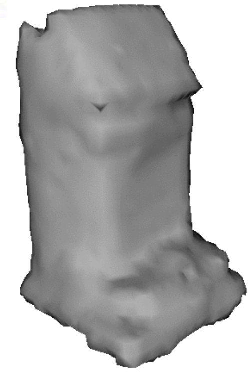
Example renderings from 3D models



Recap: hierarchical model scale levels

Martin Jagersand
U of Alberta

1. Macro:



- SFM, SFS can generate coarse geometry but not detailed enough for realistic rendering
- Integrate tracking and structure computation

Scale: dozen pixels and up

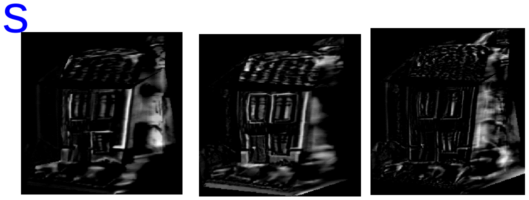
2. Meso :

- Refine coarse geometry and acquire reflectance– variational surface evolution

Scale: 1-dozen pixels

3. Micro spatial basis :

- Represents appearance and corrects for small geometric texture errors limited by linearity of image Scale: 0-5 pixels



Martin Jagersand
U of Alberta

Comparison

1. Static texturing: (Many, e.g. Baumgartner et al. 3DSOM)
   – Average color projected to point.
   – Better: Pick color minimizing reprojection error over all input images

   Works when model geometry is close to ground truth and light simple
2. Viewdependent texture (Debevec et al)
   – Pick color from closest input photograph (or interpolate from nearest 3)

   Works when possible to store large numbers of images
3. Lumigraph / Surface light field (Buehler et al / Wood et al)
   – Store all ray colors (plenoptic function) intersecting a proxy surface

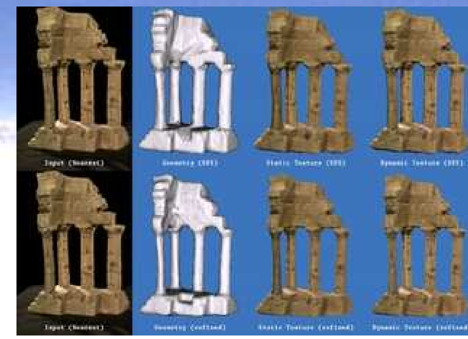
   Works if proxy surface close to true geometry
4. Dynamic texture (Ours: Jagersand '97/ Matusik / Ikeuchi99 /Vasilescu04...
   – Derive a Taylor expansion and represent derivatives of view dependency

   Works for light and small (1-5 pixel) geometric displacements.

videos

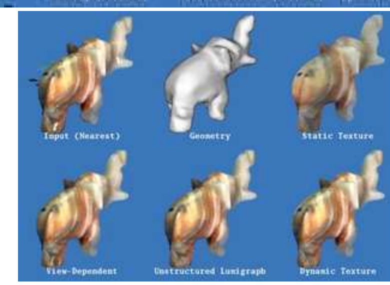
Martin Jagersand
U of Alberta

From Simple to Complex Scenes
4 test cases



1. Simple Geom: SFS alone ok

2. General Geom: SFS + Variational Shape and Reflectance fitting (+View dep texture)



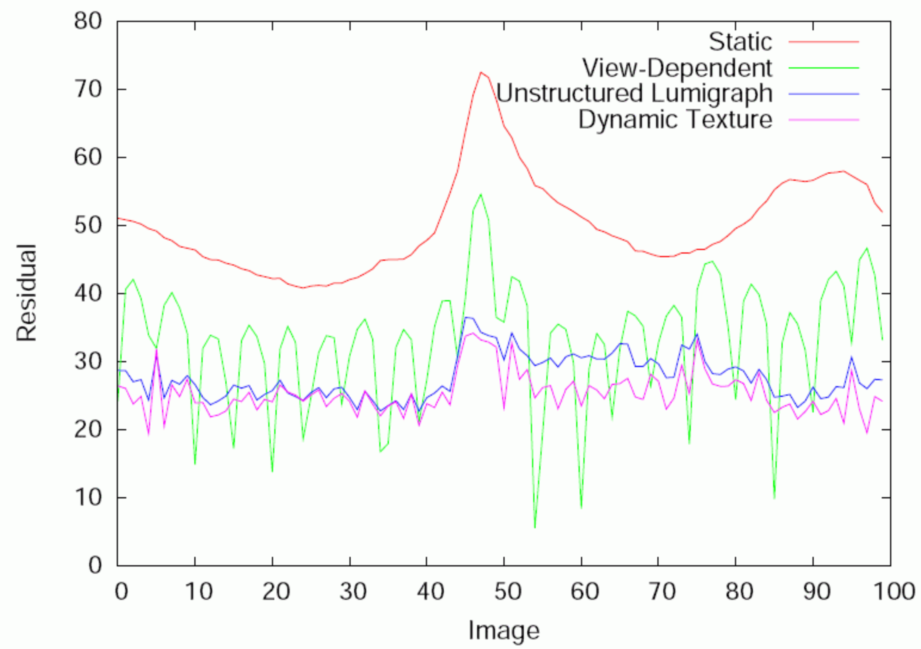
3. Complex Light: Dynamic Texture / Lumigraph

4. Challenge for Computer Vision

From Simple to Complex Scenes
4 test cases

Martin Jagersand
U of Alberta

1. Simple Geom: SFS alone ok
2. General Geom: SFS + Variational Shape and Reflectance fitting (+View dep texture)
3. Complex Light: Dynamic Texture / Lumigraph
4. Challenge for Computer Vision



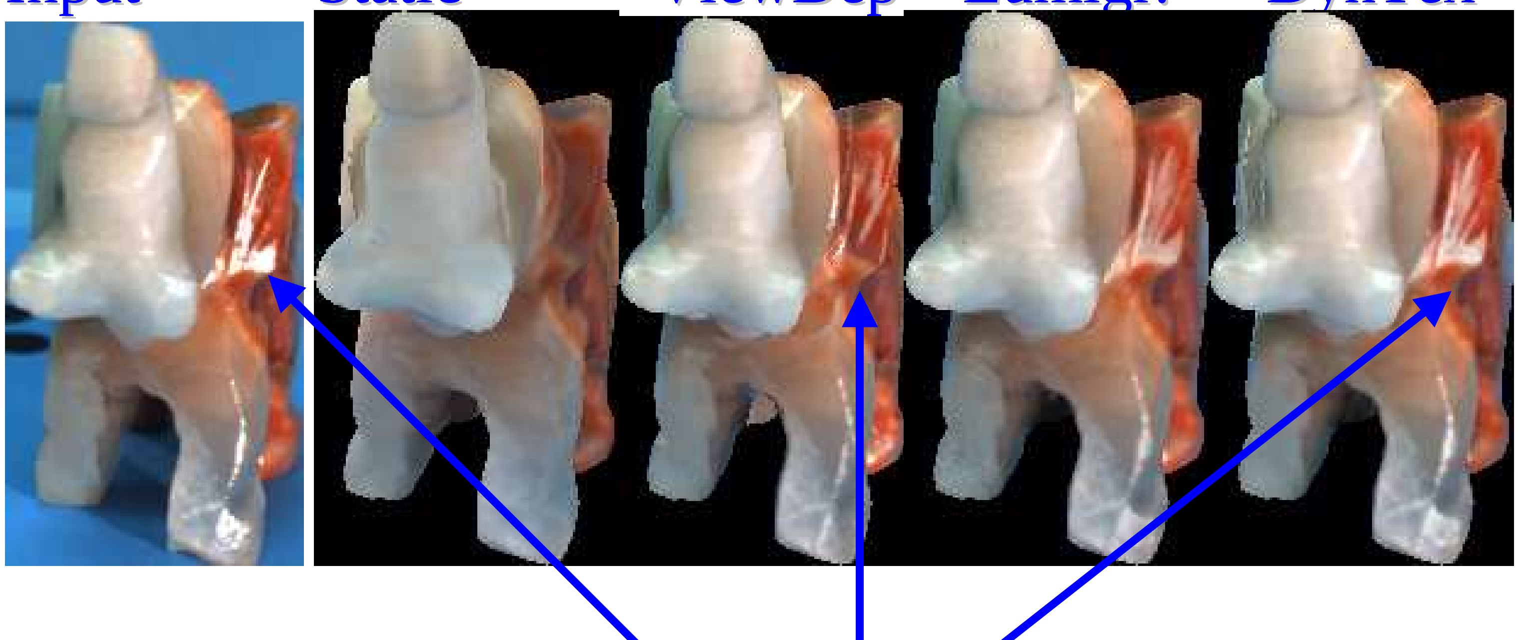
err (var)	temple	house	eleph.	wreath
Static	10.8(1.5)	11.8(1.2)	19.0(1.4)	28.4(2.8)
VDTM	8.3(1.9)	9.8(1.3)	10.1(1.9)	21.4(3.5)
Lumigr	10.8(2.5)	9.8(1.2)	5.9(0.7)	14.3(1.3)
DynTex	7.3(1.0)	9.4(1.0)	6.6(0.7)	13.4(1.2)

Table 1. Numerical texture errors and variance. %-scale.

Example of render differences

- Jade Elephant
  - Complex Reflectance (specularities and scattering)

Input Static ViewDep Lumigr. DynTex



Specular highlight

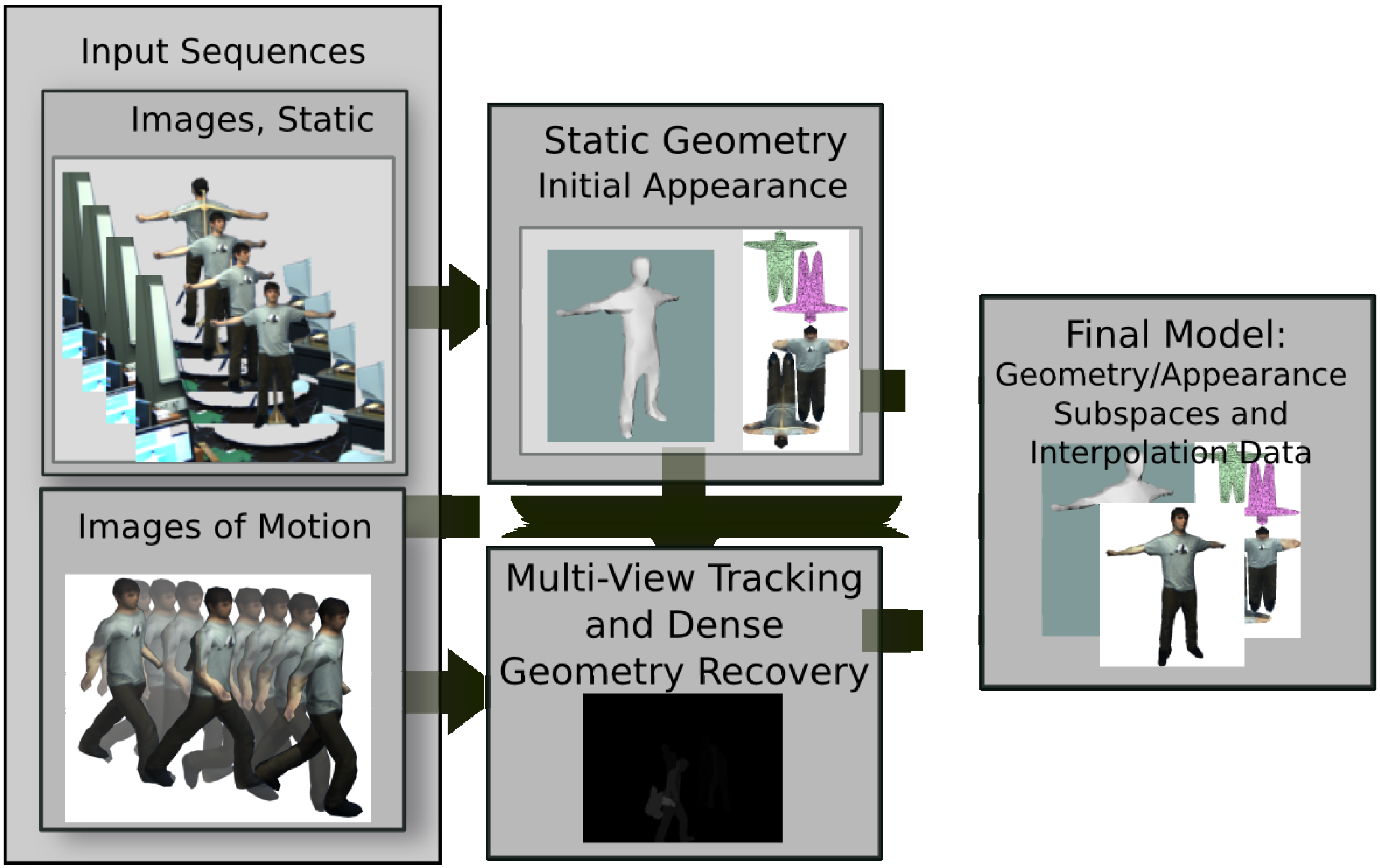
Martin Jagersand
U of Alberta

Tracking with a dyntex model + AR

Martin Jagersand
U of Alberta

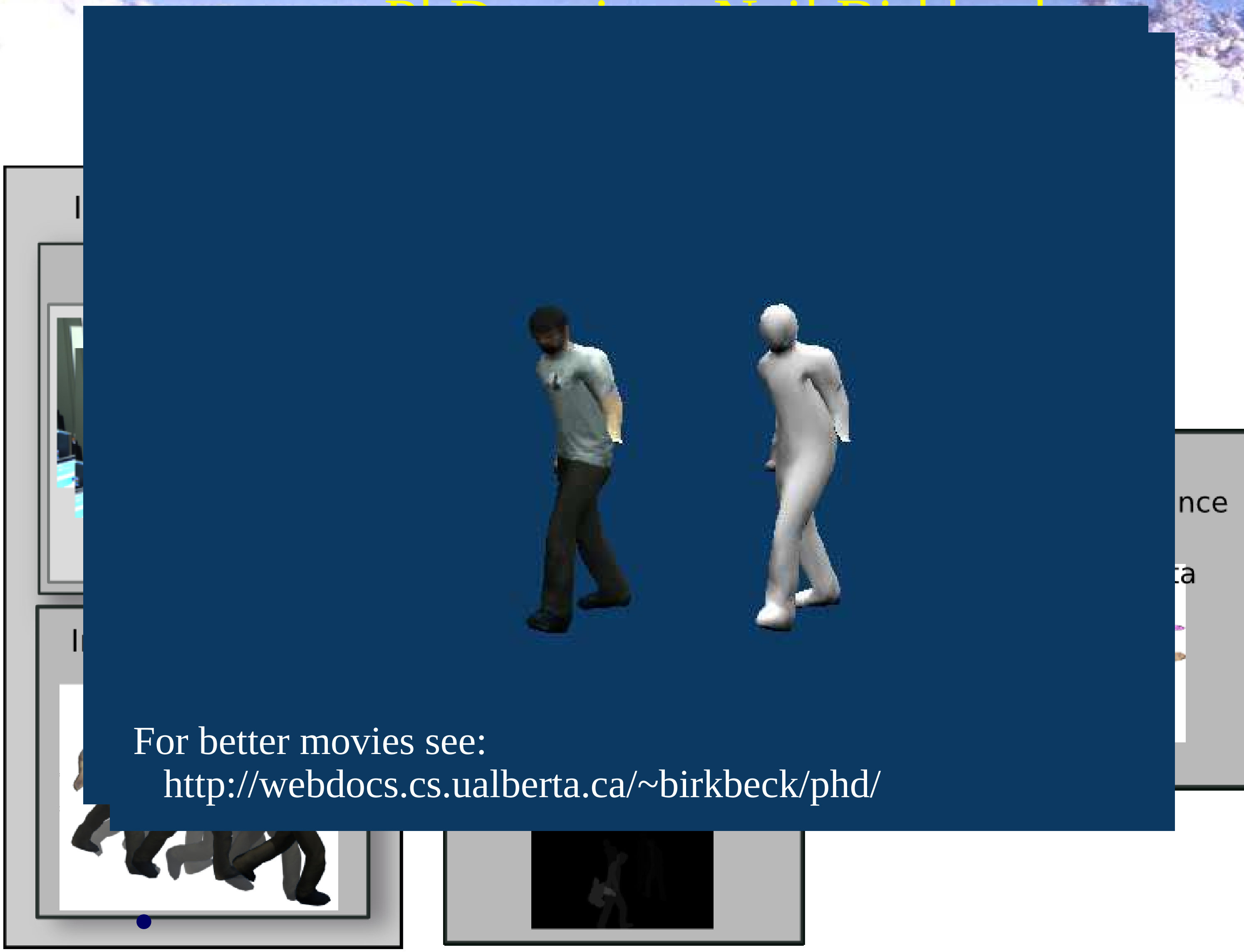
Capturing non-rigid animatable models

current PhD project, Neil Birkbeck



Martin Jagersand
U of Alberta

Capturing non-rigid animatable models



For better movies see:
http://webdocs.cs.ualberta.ca/~birkbeck/phd/

Questions?

More information:

•Downloadable renderer+models

www.cs.ualberta.ca/~vis/ibmr

•Capturing software + IEEE VR tutorial text

www.cs.ualberta.ca/~vis/VR2003tut

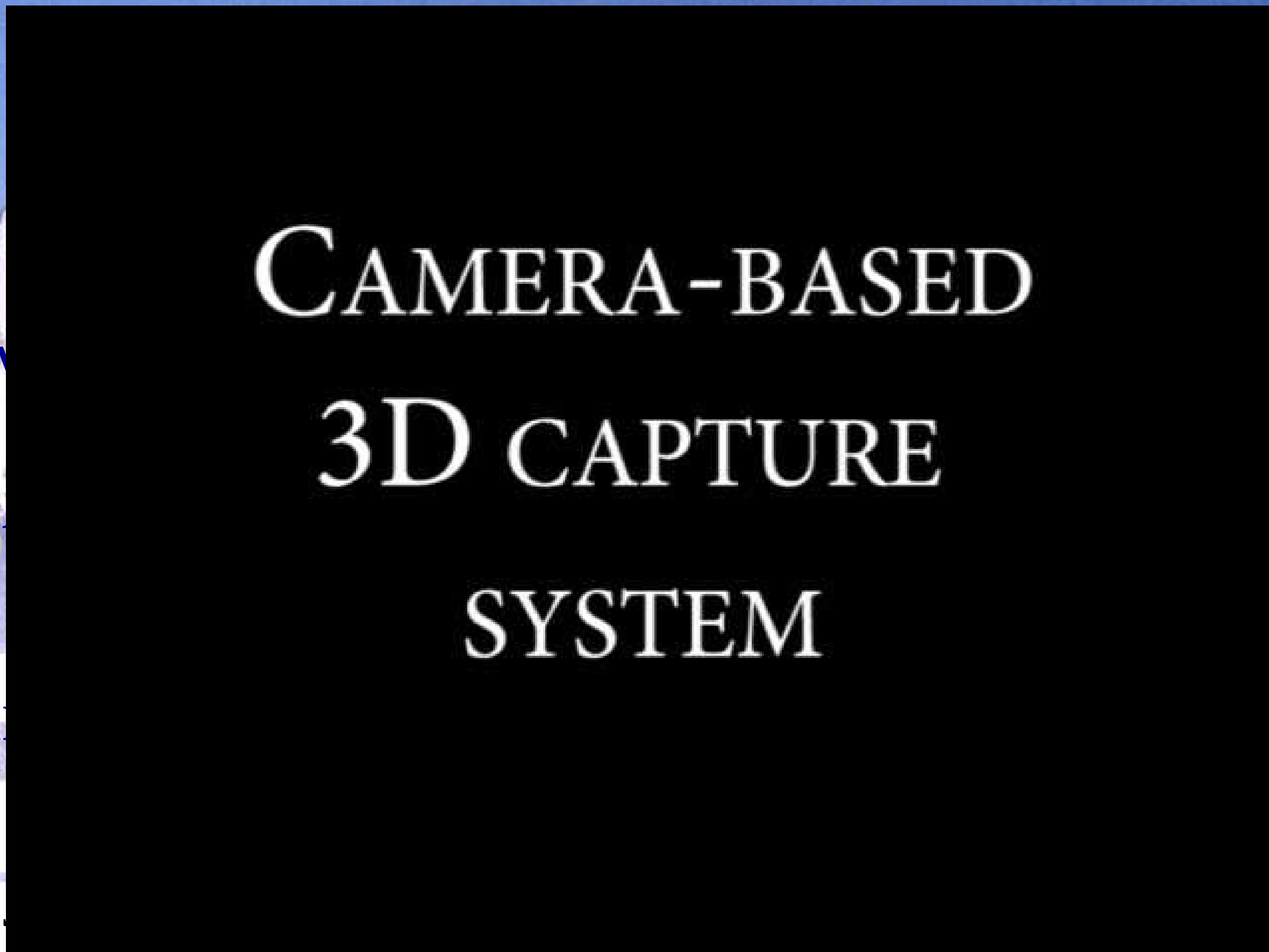
•Main references for this talk:

Jagersand et al “Three Tier Model” 3DPVT 2008 ….

Jagersand “Image-based Animation…” CVPR 1997

•More papers: www.cs.ualberta.ca/~jag

Questions?



•Dow

•Capt

•Mai

•More papers: www.cs.ualberta.ca/~jag