IEEE VR 2003 tutorial 1

Recent Methods for Image-based Modeling and Rendering

Lecture 4: Texturing

Darius Burschka
Johns Hopkins University

Martin Jagersand
University of Alberta

Dana Cobzas
University of Alberta

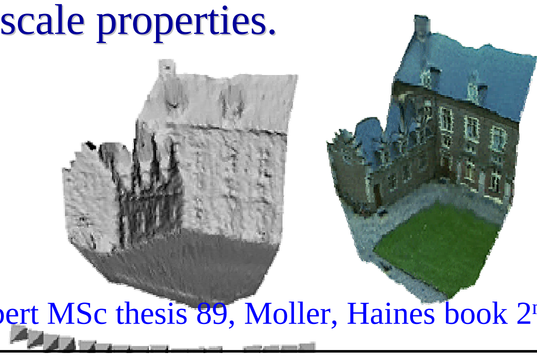
Keith Yerex
Virtual Universe Corporation

Zach Dodds
Harvey Mudd College

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

Traditional Texturing

- Texture = Fine scale surface appearance.
- In computer graphics texturing means endowing the surface of a geometric (polygonal) model with such fine scale properties.



Std ref: Heckbert MSc thesis 89, Moller, Haines book 2nd ed. 2002.

Traditional Texturing Methods and Uses

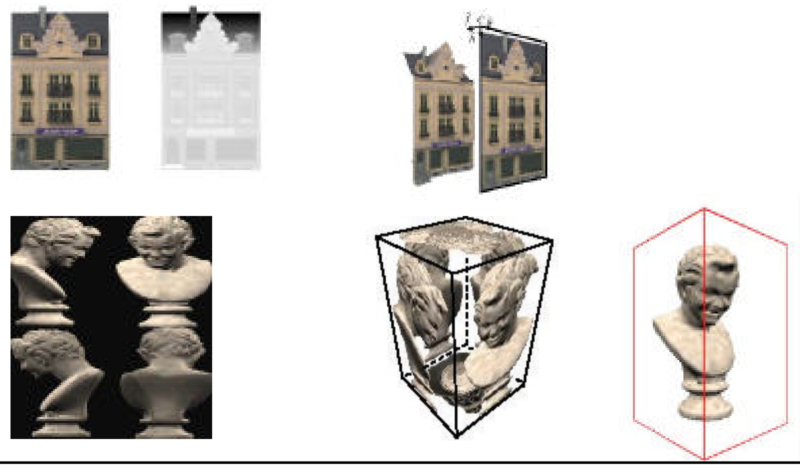
- **2D image texturing:** Wrap a 2-D image around the surface of a geometric object by stretching and replicating the image.
- **3D volumetric texturing:** "Carve" the object surface out of a 3D texture function, e.g. representing wood.
- **Light map:** An image that captures diffuse static lighting and combined with the R,G,B image texture generates a lit scene.
- **Bump map:** Generates an illusion of 3D "bumps" by perturbing surface normals for pixels in the light calculation.
- **Environment map:** Maps environment reflections onto a reflective object surface. Implemented as e.g. cube map.

Image-Based Texturing

- In Image-based Modeling and Rendering (IBMR) texture is sourced from real images of the scene or object.
- In rendering one or more of the source images are selected, combined, or interpolated/blended.
- In IBMR texture is sometimes used to capture both macroscopic and microscopic appearance, E.g.:
  1. Relief Texture
  2. View dependent texture (Facade, unstructured lumigraph)
  3. Dynamic Texture

Relief texture

- Oliviera 00, Policarpo 02, 1st price NVIDIA Cg



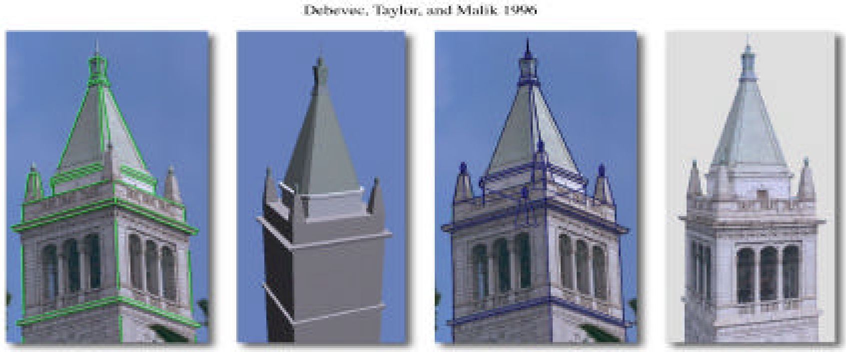
View-dependent texture selection

“Façade” (Debevec et. al.)

Texture covers large region with a flat image ignoring underlying fine scale structure:

Modeling and Rendering Architecture from Photographs

Debevec, Taylor, and Malik 1996



Original photograph with marked edges | Recovered model | Model edges projected onto photograph | Synthetic rendering

Texturing Warps

- From the geometry chapter we know that the correct plane-to-plane transform is

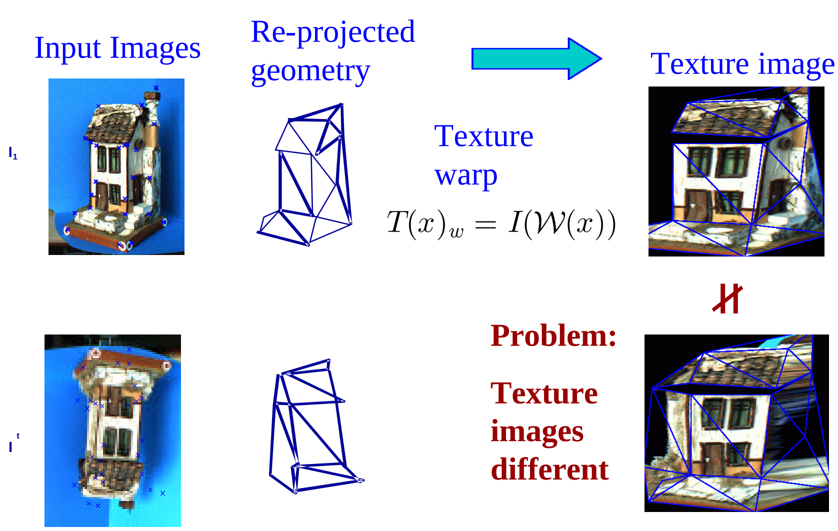
1. for a perspective camera the ***projective homography***

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \mathcal{W}_h(\mathbf{x}_h, \mathbf{h}) = \frac{1}{1+h_7 u + h_8 v} \begin{bmatrix} h_1 u & h_3 v & h_5 \\ h_2 u & h_4 v & h_6 \end{bmatrix}$$

2. for a linear camera (orthographic, weak-, para-perspective) the ***affine warp***

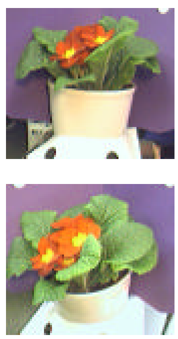
$$\begin{bmatrix} u_w \\ v_w \end{bmatrix} = \mathcal{W}_a(\mathbf{p}, \mathbf{a}) = \begin{bmatrix} \mathbf{a_3} & \mathbf{a_4} \\ \mathbf{a_5} & \mathbf{a_6} \end{bmatrix} \mathbf{p} + \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}$$

Images -> Texture

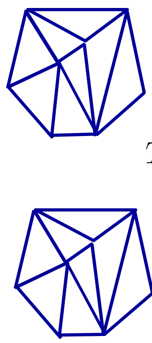


Images -> Texture

Images

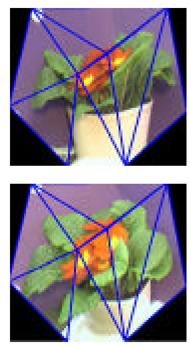


Re-projected geometry



$$T(x)_w = I(\mathcal{W}(x))$$

Textures



Representing Varying Textures

- Problem: For different views, texture $T_j \neq T_k, \quad j \neq k$
- Proposed solutions
  1. View dependent textures: (Debevec et. al. ’96 -)
     Note that for close views $T_j \approx T_k, \quad j \approx k$
     Texture from an input image from a close view
  2. Dynamic textures: (Jagersand 97, Cobzas et. al. 02)
     Modulate a texture basis

$$\mathbf{T}_j = B\mathbf{y}_j, \quad j \in 1 \ldots m$$

Dynamic Textures

Purpose

Model image intensity variations due to

1. Small geometric errors due to tracking (Image plane variations)
2. Non planarity of real surface (Out-of-plane variation)
3. Non-rigidity of real object
4. Pose varying lighting effects

Non-geometric, mixing of spatial basis $\mathbf{T}_{wt} = B\mathbf{y}_t + \mathbf{T}_0$

Spatial Basis Intro

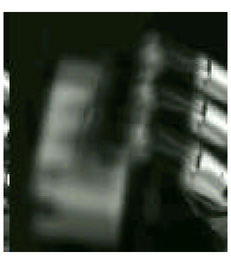
1. Moving sine wave can be modeled:

$$I = \sin(u + at) = \sin(u)\cos(at) + \cos(u)\sin(at) = \sin(u)y_1 + cos(u)y_2$$

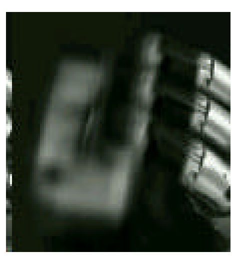
2. Small image motion

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

Spatially fixed basis



2 basis vectors



6 basis vectors

Image Variability

- Formally consider residual variation in an image stabilization problem

$$\Delta T = T_q(\hat{w}) - T(t)$$

- Optic flow type constraint

$$\Delta T = T(w+\Delta w) - T_w = T(w) + \frac{\partial}{\partial w}T(w)\Delta w - T_w = \frac{\partial}{\partial w}T(w)\Delta w$$

- Many ways to parameterize: e.g. in world coordinates using perspective eq:

$$\Delta T = \frac{1}{Z}\left(\frac{\partial T}{\partial u}\Delta X + \frac{\partial T}{\partial v}\Delta Y - \left(\frac{\partial T}{\partial u}u + \frac{\partial T}{\partial v}v\right)\Delta Z\right)$$

Planar Texture Variability 1
Affine Variability

- Affine warp function

$$\begin{bmatrix} u_w \\ v_w \end{bmatrix} = \mathcal{W}_a(\mathbf{p}, \mathbf{a}) = \begin{bmatrix} \mathbf{a_3} & \mathbf{a_4} \\ \mathbf{a_5} & \mathbf{a_6} \end{bmatrix} \mathbf{p} + \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}$$

- Corresponding image variability

$$\Delta \mathbf{T}_a = \sum_{i=1}^{6} \frac{\partial}{\partial a_i}\mathbf{T}_w \Delta a_i = \left[\frac{\partial I}{\partial u}, \frac{\partial I}{\partial v}\right] \begin{bmatrix} \frac{\partial u}{\partial a_1} & \cdots & \frac{\partial u}{\partial a_6} \\ \frac{\partial v}{\partial a_1} & \cdots & \frac{\partial v}{\partial a_6} \end{bmatrix} \begin{bmatrix} \Delta a_1 \\ \vdots \\ \Delta a_6 \end{bmatrix}$$

- Discretized for images

$$\Delta \mathbf{T_a} = \left[\frac{\partial \mathbf{T}}{\partial \mathbf{u}}, \frac{\partial \mathbf{T}}{\partial \mathbf{v}}\right] \begin{bmatrix} \mathbf{1} & \mathbf{0} & *\mathbf{u} & \mathbf{0} & *\mathbf{v} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & *\mathbf{u} & \mathbf{0} & *\mathbf{v} \end{bmatrix} \begin{bmatrix} \mathbf{y_1} \\ \vdots \\ \mathbf{y_6} \end{bmatrix}$$

$$= [\mathbf{B}_1 \ldots \mathbf{B}_6][y_1, \ldots, y_6]^T = B_a \mathbf{y_a}$$

Planar Texture Variability 2
Projective Variability

- 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:

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

$$= [\mathbf{B}_1 \ldots \mathbf{B}_8][y_1, \ldots, y_8]^T = B_h \mathbf{y}_h$$

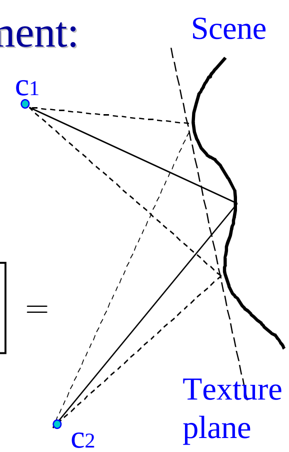
- Where $c_1 = 1 + h_7u + h_8v$, $c_2 = h_1u + h_3v + h_5$

  and $c_3 = h_2u + h_4v + h_6$

Out-of-plane variability

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

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



- Texture basis

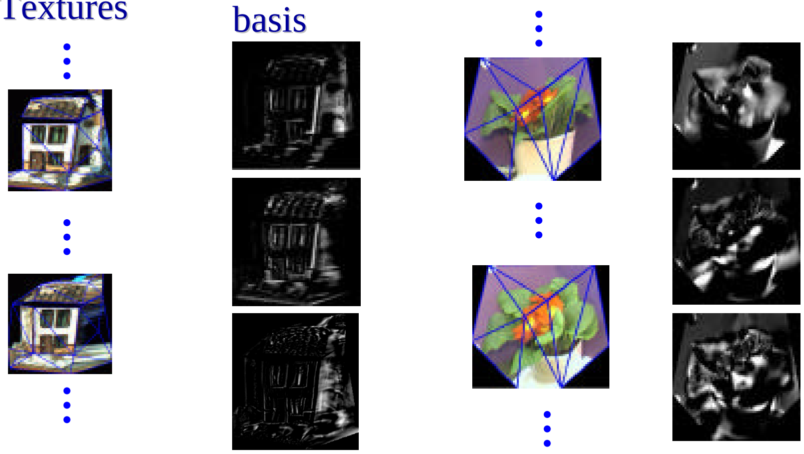
$$\Delta\mathbf{T}_\mathbf{p} = \mathbf{d}(\mathbf{u}, \mathbf{v}) \left[\frac{\partial\mathbf{T}}{\partial\mathbf{u}}, \frac{\partial\mathbf{T}}{\partial\mathbf{v}}\right] \begin{bmatrix} \frac{1}{\cos^2\alpha} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\cos^2\beta} \end{bmatrix} \begin{bmatrix} \mathbf{\Delta}\alpha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

$$= \mathbf{B}_\mathbf{p}\mathbf{y}_\mathbf{p}$$

Geometric Texture Variability Examples

Textures

Variability basis



Light basis

$$\Delta \mathbf{T}_l = \mathbf{B}_l \mathbf{y}_l$$

Demo E:\demos\light\light.exe

Composite Image variability

- Similarly can show that composite image variability

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

Struct Depth Non-plan Light Res Err

- Can be modeled as sum of basis

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

Statistical Image Variability

- In practice image variability hard to compute from one image
- Instead we use PCA to estimate image variability from a large sequence of images
- From previous analysis we expect

$$\text{rank}[\mathbf{T}_1, \ldots, \mathbf{T}_m] \approx 20$$

- Hence keep 20 or more eigen-vectors

Statistical Image Variability

- PCA yields basis that spans the texture varibility, but up to a linear transform

$$\Delta \mathbf{I} = \hat{B}\hat{\mathbf{y}}$$

- Can estimate local linear model J between

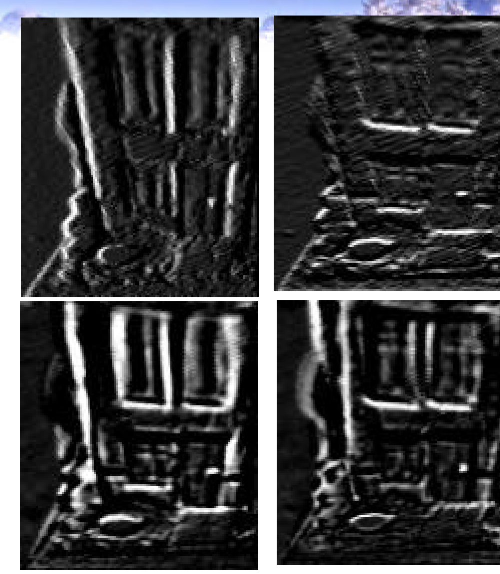
$$\Delta \hat{\mathbf{y}} = J \Delta \mathbf{y}$$

- In practice Delaunay triangulation & interpolation (bi-linear) on triangular element.

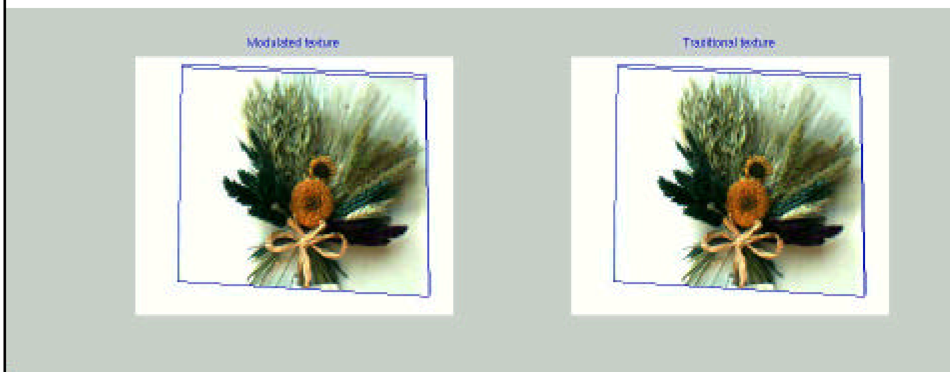
Image variability comparison

Derivatives from one picture

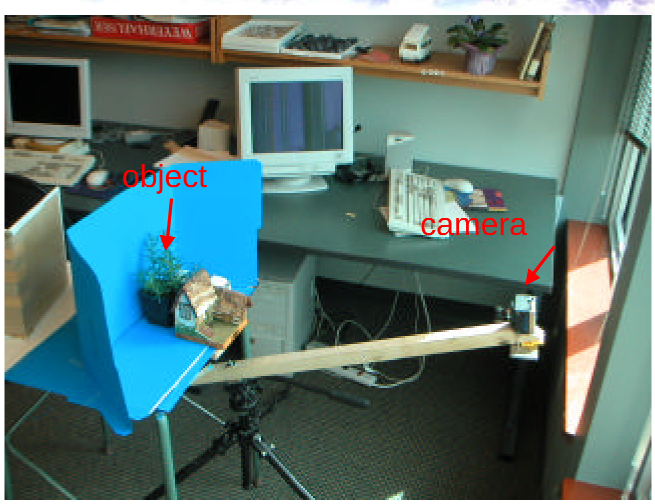
Statistically estimated variability



Dynamic texture example

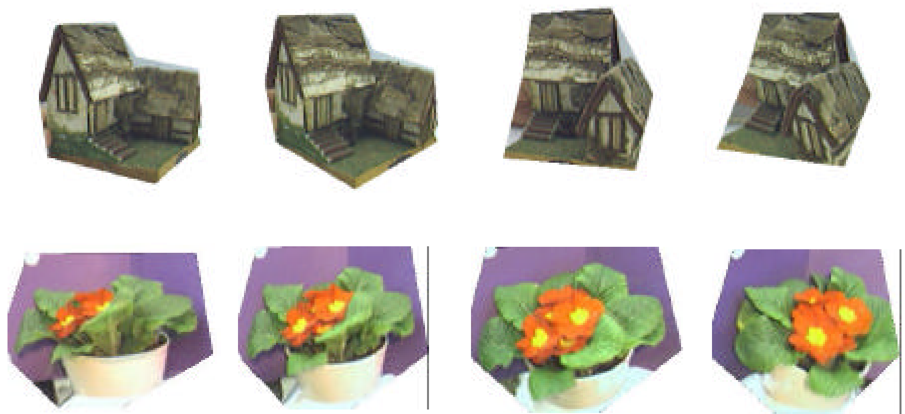


Texture Capture Setup

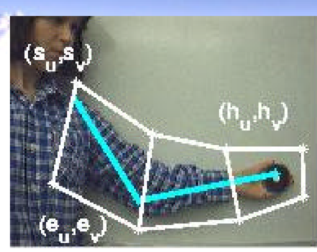


256-512 training images; 20-50 basis textures

Example Renderings



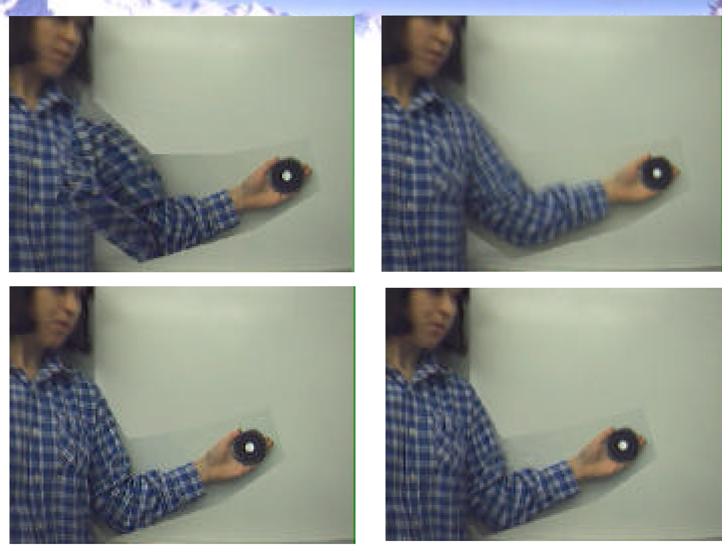
Kinematic arm



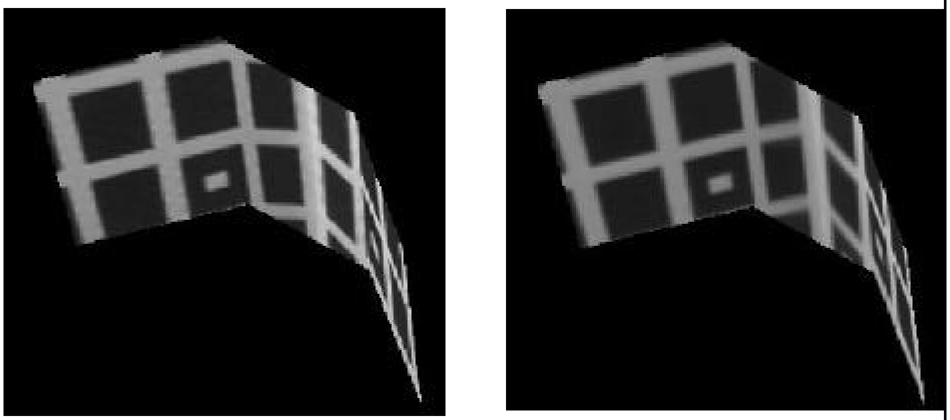
Geometric errors

static dynamic

Geometric errors

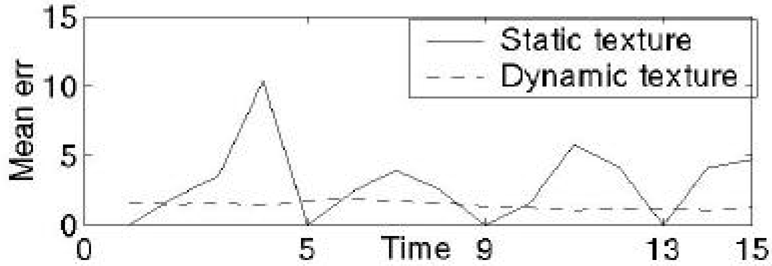


Geometric errors



Static texturing — Dynamic

Pixel error



	Vertical jitter	Horizontal jitter
Static texture	1.15	0.98
Dynamic texture	0.52	0.71

Dynamic Texturing Conclusions

- Tractable to estimate from images
- Compensates for errors from small geometric misalignments.
- Captures existing scene lighting
- Runs on consumer PC with “gaming” graphics card
- Applications
  - Capture of real objects, scenes
  - Insert characters into games
  - Video phone