Modulating View-dependent Textures

Martin Jägersand, Dana Cobzaş, Keith Yerex

Department of Computing Science, University of Alberta, Canada
{jag,dana,keith}@cs.ualberta.ca

Abstract

We present a texturing approach for image-based modeling and rendering, where instead of using one (or a blend of a few) sample images, new view dependent textures are synthesized by modulating a differential texture basis. The texture basis models the first order intensity variation due to image projection errors and parallax for a non-linear projective camera. Experimentally we compare rendered views to ground truth real images and quantify how the texture basis can generate a more accurate rendering compared to conventional view dependent textures.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Image Based Rendering, Texture

1. Introduction

Recently subspace methods have been popularized for view-dependent texture models (VDTM). Most work assumes an accurate object geometry and estimates or parameterizes the bi-directional texture function, e.g. [4, 12, 10, 11]. However, a spatially fixed basis can be used to modulate complex geometric motion. This has been shown in image plane synthesis of articulated and non-rigid deformations[9] and more lately, temporally parameterized image sequences[6]. It has also been applied to texture rendering using simple (linear) camera models[3]. In practice modulating a subspace basis to generate view-dependent textures also works quite well for general non-linear perspective cameras, and the objective of this paper is to expose both mathematically and experimentally how and why it works.

Important applications are mainly in conjunction with image-based modeling and rendering (IBMR) from uncalibrated video (e.g. from hand-held camcorders) using so called structure-from-motion (SFM) methods from Computer Vision[8]. These allow simple and inexpensive capture of scene geometry, but the resulting model is only moderately accurate, which causes problems with texture coordinate alignment and parallax. Traditional VTM addresses this by blending real images acquired from nearby viewpoints[5]. However, this results in small jumps or fades between views, and somewhat incorrect rendering of intermediate views. Here we show how a spatial basis can be derived and used to modulate texture views for continuously varying viewpoints without these jumps. The capability to texture an inexact geometry would also be applicable to replace the image blending in recent lightfield/lumigraph methods where the proxy geometry closely resembles the scene, e.g.[2].

In the following sections we first develop the ground work for our method by generalizing optic flow from image-plane x,y-translation to the multi-dimensional variability on the texture plane. In Section 3 we show the existence of a linear basis and its analytic form, which allow the synthesis of correct view-dependent textures around a reference viewpoint. Section 4 describes how a texture subspace-basis can be estimated from images without an exact knowledge of the geometry, and then shows how this basis is equivalent to the analytic basis, hence allows the continuous jump-free modulation. Finally we illustrate this texturing applied to rendering real scenes captured from uncalibrated video.

2. Background

A structure-from-motion (SFM) method starts with a set of m images $I_1 \dots I_m$ from different views of a scene. Through visual tracking the image projection $x_1 \dots x_n$ of n physical scene points are identified in every image. From this, the SFM algorithm[8] computes a structure, represented by a set of n scene points $X_1 \dots X_n$, and m view projections $P_1 \dots P_m$ that satisfy the reprojection property:

$$x_{j,i} = P_j X_i \quad i \in 1 \dots n, j \in 1 \dots m \tag{1}$$

Central to image-based modeling and rendering is that this structure can be reprojected into a new virtual camera and thus novel views can be rendered. Practically, the structure is divided into Q planar facets (we use triangles or quadrilaterals) with $x_{j,i}$ as node points. For texture mapping, each one of the model facets are related by a planar projective homography to a texture image. See Fig. 1.

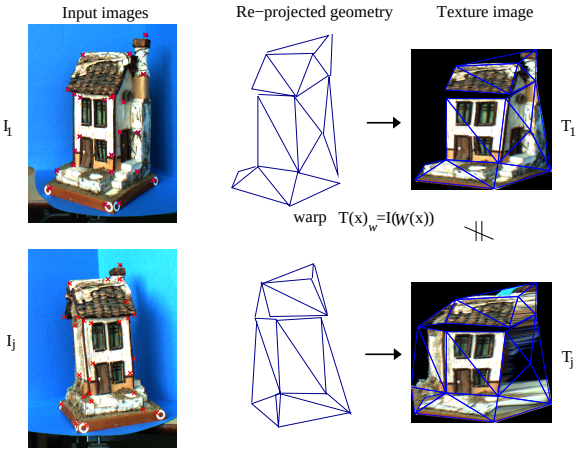

Figure 1: *Textures generated from two images using a coarse geometry.*

In conventional texture mapping, texture patches are extracted from one or more real images and warped onto the re-projected structure in the new view. Instead of using one image as a source texture, here we study how to relate and unify all the input sample images into a texture basis. Let $x_{T,i}$ be a set of texture coordinates in one-to-one correspondence to model points X_i and thus also with the image points $x_{j,i}$ for each view j. A texture warp function $\mathcal{W}$ uses the model vertex to texture correspondences to rearrange interior pixels from texture space T to the image space I.

$$T(\mathbf{x}) = I(\mathcal{W}(\mathbf{x};\mu)) \tag{2}$$

where μ are the warp parameters and $\mathbf{x}$ the texture pixel coordinates.

Now if for each sample view j, we warp the real image I_j from image to texture coordinates into a texture image T_j, we would find that in general $T_j \neq T_k,\ j \neq k$ as illustrated in Fig. 1. Typically, the closer view j is to k, the smaller is the difference between T_j and T_k. This is the rationale for view-dependent texturing, where a new view is textured from one to three (by blending) close sample images[5].

In this paper we will develop a more principled approach, where we seek a texture basis B such that for each sample view:

$$\mathbf{T}_j = B\mathbf{y}_j,\ j \in 1\ldots m. \tag{3}$$

Here, and in the following, $\mathbf{T}$ is a $q \times q$ texture image flattened into a $q^2 \times 1$ column vector, B is a $q^2 \times r$ matrix ($r \ll m$), and $\mathbf{y}$ is a modulation vector. The texture basis B needs to capture the geometric variability over the sample sequence, and correctly interpolate new in-between views.

3. Geometric texture variation

A simple example of geometric variation caused by small image-plane translations is the well known optic flow constraint that relates texture intensity change $\Delta\mathbf{T} = T_j - T_k$ to spatial derivatives $\frac{\partial}{\partial u}T, \frac{\partial}{\partial v}T$ with respect to texture coordinates $\mathbf{x} = [u,v]^T$ under an image constancy assumption[7].

$$\Delta T = \frac{\partial T}{\partial u}\Delta u + \frac{\partial T}{\partial v}\Delta v \tag{4}$$

Given one reference texture T_0 we can build a basis $B = [T_0, \frac{\partial T}{\partial u}, \frac{\partial T}{\partial v}]$ and from this generate any slightly translated texture $T(\Delta u, \Delta v) = B[1, \Delta u, \Delta v]^T = B\mathbf{y}$

In a real situation, the patch is deforming in a more complex way than pure translation. This deformation is captured by the warp parameters μ. Given a warp function $\mathbf{x}' = \mathcal{W}(\mathbf{x},\mu)$ we study the residual image variability introduced by the imperfect stabilization achieved by a perturbed warp $\mathcal{W}(\mathbf{x};\hat{\mu})$, $\Delta T = T(\mathcal{W}(\mathbf{x};\hat{\mu}), j) - T(\mathcal{W}(\mathbf{x};\mu))$. Similar image variability has been used for visual tracking. For uniformity we use a notation consistent with the literature, see e.g.[7]. Denoting $\hat{\mu} = \mu + \Delta\mu$ we rewrite ΔT as a first order approximation (dropping j):

$$\begin{aligned}
\Delta T &= T(\mathcal{W}(\mathbf{x};\mu+\Delta\mu)) - T_W \\
&= T(\mathcal{W}(\mathbf{x};\mu)) + \nabla T \frac{\partial \mathcal{W}}{\partial \mu}\Delta\mu - T_W \\
&\approx \nabla T \frac{\partial \mathcal{W}}{\partial \mu}\Delta\mu \\
&= \left[\frac{\partial T}{\partial u}, \frac{\partial T}{\partial v}\right] \begin{bmatrix} \frac{\partial u}{\partial \mu_1} \cdots \frac{\partial u}{\partial \mu_k} \\ \frac{\partial v}{\partial \mu_1} \cdots \frac{\partial v}{\partial \mu_k} \end{bmatrix} \Delta[\mu_1 \ldots \mu_k]^T
\end{aligned} \tag{5}$$

Next we give examples of how to concretely express image variability for a mesh element. In image-based modeling and rendering real source images are warped into new views given an estimated scene structure. Errors between the estimated and true scene geometry generate imperfect renderings. We divide these errors into *image plane* and *out of plane* errors. The out of plane errors arise when piecewise planar facets in the model are not true planes in the scene. The planar errors cause the texture to be sourced with an incorrect warp. In IBMR planar reprojection errors stem from errors in tracking point correspondences when computing the SFM, as well as projection errors due to e.g. lens distortions.

Planar texture variability In most rendering systems textures facets $\mathbf{T}$ are warped onto the rendered views using a a projective homography.

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \mathcal{W}_h(\mathbf{x}_h, \mathbf{h}) = \frac{1}{1 + h_7 u + h_8 v} \begin{bmatrix} h_1 u & h_3 v & h_5 \\ h_2 u & h_4 v & h_6 \end{bmatrix} \tag{6}$$

Specializing Eq. 5 with the derivatives of $\mathcal{W}_h$ we get:

$$\begin{aligned}
&\Delta\mathbf{T}_h(u,v) \\
&= \tfrac{1}{c_1}\left[\tfrac{\partial \mathbf{T}}{\partial u}, \tfrac{\partial \mathbf{T}}{\partial v}\right] \begin{bmatrix} u & 0 & v & 0 & 1 & 0 & -\frac{uc_2}{c_1} & -\frac{vc_2}{c_1} \\ 0 & u & 0 & v & 0 & 1 & -\frac{uc_3}{c_1} & -\frac{vc_3}{c_1} \end{bmatrix} \begin{bmatrix} \Delta h_1 \\ \vdots \\ \Delta h_8 \end{bmatrix} \\
&= [\mathbf{b}_1 \ldots \mathbf{b}_8][y_1, \ldots, y_8]^T = B_h \mathbf{y}_h
\end{aligned} \tag{7}$$

where $c_1 = 1 + h_7u + h_8v$, $c_2 = h_1u + h_3v + h_5$, and $c_3 = h_2u + h_4v + h_6$.

Non-planar parallax variation The real world scene is seldom perfectly piecewise planar. In rendering this gives rise to parallax errors. Fig. 2 illustrates how the texture plane image T changes for different scene camera centers C. Let $[\alpha, \beta]$ be the angle between the ray from the camera center C_j to each scene point. The pre-warp rearrangement needed on the texture plane to correctly render this scene using a standard homography warp is then:

$$\begin{bmatrix} \delta u \\ \delta v \end{bmatrix} = \mathcal{W}_p(\mathbf{x}, \mathbf{d}) = d(u,v) \begin{bmatrix} \tan\alpha \\ \tan\beta \end{bmatrix} \tag{8}$$

As before, taking the derivatives of the warp function with respect to a camera angle change and inserting into Eq.5 we get:

$$\Delta\mathbf{T}_p(u,v) = d(u,v) \left[\frac{\partial \mathbf{T}}{\partial u}, \frac{\partial \mathbf{T}}{\partial v} \right] \begin{bmatrix} \frac{1}{\cos^2\alpha} & 0 \\ 0 & \frac{1}{\cos^2\beta} \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\beta \end{bmatrix} = B_p \mathbf{y}_p \tag{9}$$

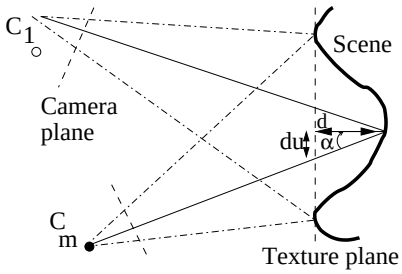

Figure 2: *Texture parallax between two views (planar representation).*

4. Estimating texture variability from video

Just as derivatives can be either analytical, or estimated by discrete differences, here we show how to estimate the texture basis. From an input video sequence we obtain a pose-labeled texture set $\tilde{T} = [T(\mathbf{x}_1) \ldots T(\mathbf{x}_m)]$ using Eq. 2. In principle this video texture set could be used for standard VDTM, but in practice the set would be too large, and one would have to select a small subset of views.

From the derivation in previous section we know to expect a texture variability of the form in Eq. 7 and 9. Hence the texture for a new view k can be written written $\mathbf{T}_k = [\mathbf{T}_0, B_h, B_p][1, y_2, \ldots, y_{11}]^T = B\mathbf{y}_k$ with respect to a reference texture T_0 (chosen e.g. from one view central in the sample set, or the mean of several views). Hence, note that B is contained as a subspace in $\tilde{T}$, i.e. $\text{span}(B) \subset \text{span}(\tilde{T})$. To be able to modulate textures from new viewpoints, we wish to extract a compact approximation of B. If there was no other variability in the video sequence, $\tilde{T}$ would span exactly B. In practice $\tilde{T}$ is full rank and contains variability also due to e.g. light, BRDF, noise etc. The two former may be significant depending on the scene. Our strategy is to from

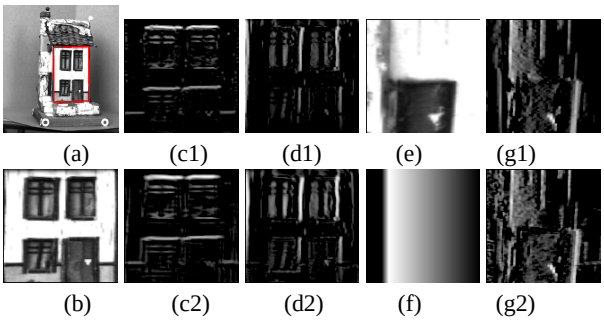

Figure 3: *Comparison of analytical and estimated basis for geometric variability. Plane variability: (a) original quadrilateral; (b) warped texture; (c1),(d1) analytical basis ($\mathbf{b}_1, \mathbf{b}_4$ from Eq. 7); (c2),(d2) corresponding recovered $\tilde{\mathbf{b}}$-basis. Parallax variability: (e) reference texture image; (f) depth map; (g1) analytical basis (Eq. 9); (g2) recovered basis by PCA*

$\tilde{T}$ extract a linear subspace $\tilde{B} = [\tilde{\mathbf{b}}_1 \ldots \tilde{\mathbf{b}}_r]$ somewhat larger than B using PCA. Typically we pick $r = 20$-dimensional subspace from the hundreds or more video images in $\tilde{T}$. (A more precise argument is that for a (near) Lambertian object a 9-dimensional basis of spherical harmonics span 98% of the light variability[1], hence a 20 dimensional $\tilde{B}$-space will capture both geometric and light variability, and contain B as a subspace.)

To validate that this 20-dimensional subspace actually contain B_h and B_p we computed both the analytical and PCA based variability for some texture elements. We found that $\tilde{B}$ contained 99.5% of the variability in the analytical basis B. Additionally, through a basis transform, the columns of $\tilde{B}$ can be aligned with a known B, and as illustrated in Fig. 3, the analytical and estimated basis vectors look virtually identical. The important conclusion to draw here is that when an appropriate size texture subspace-basis is estimated from a dense video sequence it will span the analytical basis. Unlike VDTM, where regular images are blended, this basis contains derivatives of images and Eq. 3 can thus be interpreted as a first order Taylor expansion, allowing continuous modulation of texture changes instead of fading between images. Obviously the validity of the first order model is limited to small variations, but works well for the small (a few pixels) texture mis-alignments encountered in IBMR.

5. Rendering objects and scenes

We tested the performance of the modulated texture for rendering objects and scenes. A sparse 3D geometry (see Fig. 1) is estimated from a set of training images $I_j, j = 1 \ldots m$ using structure-from-motion. The structure is then decomposed into planar facets that are projected into texture coordinates to generate a texture for each view T_j. From these texture images, we estimate a texture basis $\tilde{B}$ as described in Section 4. New views are rendered by modulating the texture basis and warping it to the projected geometry. The modulation coefficients $\mathbf{y}$ are calculated by interpolating the texture coefficients $\mathbf{y}_j$ from the training set for the new camera pose.

Figure 4: *Renderings of a captured dining room model. video 2*

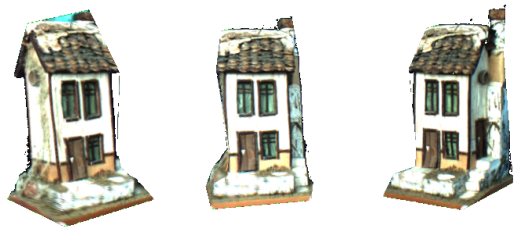

Figure 5: *Renderings of a toy house made of bark and wood video 1*

To illustrate the quality of the rendered images we present renderings of toy house (Fig. 5 and video 1) and an interior scene (Fig. 4 and video 2).

Quantitative comparison In order to quantitatively analyze how modulating a texture basis performs compared to standard view dependent texturing we captured a model $\tilde{B}$ of a wreath (almost planar with very fine depth detail). For standard VDTM we choose 30 images equally spaced over the viewpoint variation. To put the methods on an equal basis we choose the dimensionality $r = 30$ for $\tilde{B}$. Each model was rendered into 80 different poses and compared to ground truth (a real image). The pixel intensity error graph for 15 of these is shown in Fig. 6. As seen the modulated texture outperforms the VDTM for most views, except when the view is (near) identical to one of the sample images in the VDTM. video 3 shows in rendered views textured with the modulated basis texture (left) and one (not 30) sample image (right).

6. Discussion

We have presented a texturing method where for each new view a unique view-dependent texture is modulated from a texture basis. The basis is designed so that it encodes a texture intensity spatial derivatives with respect to warp and parallax parameters. (unlike conventional VDTM which blends images) In a rendered sequence the texture modulation plays a small movie on each model facet, which correctly represents the underlying true scene structure to a first order. This effectively compensates for small (up to a few pixels) geometric errors between the true scene structure and captured model. In particular we have derived the explicit analytic form for the texture variation under projective warps, and shown that these can be estimated to high (99.5 %) accuracy from uncalibrated video.

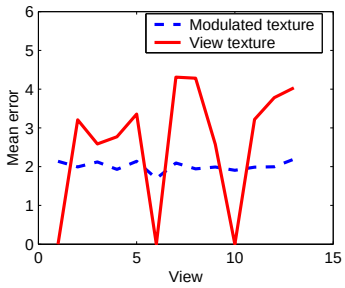

Figure 6: *Pixel intensity error when texturing from a close sample view (red) and by modulating the texture basis. For most views the texture basis gives a lower error. Only when the rendered view has the same pose as the one of the three source texture images (hence the IBR is a unity transform) is the standard view based texturing better*

The strength of our method lies in its ability to capture and render scenes with reasonable quality from uncalibrated video alone. Hence, neither a-priori models, expensive laser scanners or extensive human intervention is required. This can potentially enable applications such as virtualized and augmented reality in the consumer market.

References

[video 1-3] On-line mpeg movies of the experiments are available at http://www.cs.ualberta.ca/∼dana/EG04.

[1] R. Barsi and D. Jacobs. Lambertian refrectance and linear subspace. In *IEEE International Conference on Computer Vision*, pages 383–390, 2001. 3

[2] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *Computer Graphics (SIGGRAPH)*, pages 43–54, 2001. 1

[3] D. Cobzas, K. Yerex, and M. Jagersand. Dynamic textures for image-based rendering of fine-scale 3d structure and animation of non-rigid motion. In *Eurographics*, 2002. 1

[4] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1–34, 1999. 1

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from phtographs. In *Computer Graphics (SIGGRAPH'96)*, 1996. 1, 2

[6] G. Doretto and S. Soatto. Editable dynamic textures. In *ACM SIGGRAPH Sketches and Applications*, 2002. 1

[7] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998. 2

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1

[9] M. Jagersand. Image based view synthesis of articulated agents. In *Computer Vision and Pattern Recognition*, 1997. 1

[10] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *Computer Graphics (SIGGRAPH)*, pages 519–528. ACM Press, 2001. 1

[11] M. A. O. Vasilescu and D. Terzopoulos. Tensortextures. In *ketches and Applications SIGGRAPH*, 2003. 1

[12] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle. Surface light fields for 3d photography. In *Computer Graphics (SIGGRAPH'00)*, 2000. 1