Isometric Texture Mapping for Free-Form Surfaces

Nur Arad Gershon Elber
HP Israel Science Center*

Abstract

Texture mapping is a frequently exploited technique in computer graphics aimed at the emulation high-resolution details in surfaces. In this paper we present a distance-ratios preservation method for bivariate raster texture mapping to free-form surfaces in an arbitrarily precise manner. The proposed method reduces the original general problem of computing the inverse of a three-dimensional parametric surface mapping into a problem of two-dimensional image warping. Several examples that demonstrate the proposed approach are also provided.

Keywords: Texture mapping, image warping, isometries, distance preserving, developable surfaces.

*Address: HP Israel Science Center, Technion City, Haifa 32000, Israel. Email: [nur,gershon]@hp.technion.ac.il

1 Introduction

Photo-realistic rendering has been a prime goal of computer graphics research for over two decades. Numerous methods have been developed to improve the accuracy of the constructed image by employing physically based optical models of simulated illumination.

Texture mapping is a frequently exploited technique in contemporary computer graphics. As opposed to physically based models, this method lacks any physical foundation. However its success in emulating high resolution details on surfaces is unquestionable.

Two major types of texture mappings are frequently employed. The first defines the texturing information in 3-space as a trivariate scalar or vector-valued texture function $T(x, y, z)$ [17, 18]. Given a 3-space location, T provides an intensity level, an RGB color, or even a normal perturbation factor, also known as bump mapping [6]. In practice, the trivariate texture is typically functional. A significant amount of ingenuity is required in order to achieve appealing texture styles such as marble or wood. A second approach exploits the mapping of the bivariate parametric surface, $S(u, v) = (x(u, v), y(u, v), z(u, v))$. A bivariate scalar or vector-valued texture function $T(u, v)$ provides for each 3-space location on S the texture information via the mapping function of S. This second method is more attractive because $T(u, v)$ can be easily approximated using real life texture imagery, such as photographs of wood, grass, or clouds.

Both the above methods have their advantages and drawbacks. The use of trivariate texture mapping imposes a greater computational burden, in general, and has little support in hardware based graphics. Bivariate (also refered to as raster) texture mapping is typically faster to evaluate, and is supported by many hardware based platforms such as SGI's GL and Open GL, or HP's Starbase. However, while the trivariate texture is computed in the Euclidean 3-space, the bivariate raster texture is transformed along with the surface mapping. The latter can cause severe distortions to the input texture image, affecting both distances and angles (see Figure 1). Clearly, in many instances, the original intent is to preserve angles and distances in the image of the texture.

A mapping $S = S(u, v)$ is called *isometric* if it preserves the inner product of two directional partials derivatives at any given point on surface S, Namely if $\bar{x}_i = (x_i^u, x_i^v),\ i = 1, 2$ are vectors in the parametric space of S then,

$$\left\langle \left(\frac{\partial S}{\partial x_1^u} + \frac{\partial S}{\partial x_1^v} \right), \left(\frac{\partial S}{\partial x_2^u} + \frac{\partial S}{\partial x_2^v} \right) \right\rangle = \langle \bar{x}_1, \bar{x}_2 \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product operator. Because inner products are preserved by isometries, the first fundamental form is also preserved. Since the Gaussian curvature, K, depends only on the first fundamental form and its derivatives, K is also invariant under isometries, a result known as *Theorema Egregium* of Gauss [9]. Using bivariate raster texture mapping, one always starts with a flat planar parametric domain for which $K \equiv 0$. Hence, a necessary (but not sufficient) condition for a parametric surface $S(u, v)$ to be an isometry is that it has zero Gaussian curvature throughout. In other words, S must be developable.

Clearly, most parametric surfaces are not developable. Moreover, the developability of a surface is intrinsic, while parameterization may be arbitrary. Consider a developable parametric surface $S_1(u, v),\ 0 \le u, v \le 1$, and its reparameterized alias $S_2(u, v) = S_1(2u, 3v),\ 0 \le u \le \frac{1}{2},\ 0 \le v \le \frac{1}{3}$. While S_1 and S_2 are both developable, they cannot simultaneously be isometries.

Techniques to alleviate all these difficulties and provide a close approximate solution that preserves angles and/or distances in the bivariate raster texture have been suggested in recent years. In [5], an arbitrary parametric surface is decomposed into strips, examining and preserving the geodesic curvature of the curves that subdivide the different domains of the surface when they are mapped onto the plane. The resulting decomposition is used to provide a skeleton for the texture mapping that is used, and to possibly preserve distances in the texture during the mapping process. This approach preserves the geodesic curvature as well as arc-length by tracing a piecewise linear approximation of the real curves, one point at a time. In [13], a different approach that provides a piecewise developable surface approximation with a global error bound to an arbitrary surface is described. This approximation can be employed in a similar manner to approximate a bivariate raster texture map that preserves distances.

In this paper, a different approach is taken. Since one is, in general, unable to accurately preserve the distances in the texture, we restrict ourselves in two ways. First, we consider only distance-ratios. That is, we allow uniform scaling while the surface mapping takes place. Hence after, we use the term *isometry* to denote a distance-ratios preserving transformation. Moreover, we examine the problem of mapping a texture onto a surface such that distance ratios are preserved when the surface is orthographically projected at a particular direction. Thus, when viewed from that direction the texture mapped surface appears as an isometric copy of the original.

By considering this problem we are able to reduce the general problem of an isometry approximation of the bivariate texture mapping to that of two-dimensional image warping. Section 2 presents the proposed algorithm while in Section 3 we demonstrate the proposed solution on several examples. Finally, we conclude in Section 4 by discussing some theoretical and implementation issues associated with the problem as well as the generalization of the problem to other types of isometries.

2 The Algorithm

Assume one desires to map a certain shape onto some location on a free-form surface. For clarity and without loss of generality, we assume one needs to place a square texture shape, $\mathcal{T}$, on a free-form surface so that from the $+X$ orthographic viewing direction, $\mathcal{V}$, the shape is *seen* as square.

2.1 Reduction to 2-D

Consider the plane $\mathcal{P} : z = z_0$ that contains a horizontal boundary of the projected texture map of $\mathcal{T}$. Parameterize $\mathcal{P}$ as $P(r, s) = (r, s, z_0)$. The intersection of $\mathcal{P}$ and surface $S(u, v) = (x(u, v), y(u, v), z(u, v))$ yields,

$$x(u, v) = r, \quad y(u, v) = s, \quad z(u, v) = z_0. \tag{1}$$

The third equation, $z(u, v) = z_0$, implicitly defines the curve of intersection in u and v. Parameterizing this implicit curve with the parameter t results in a representation of the

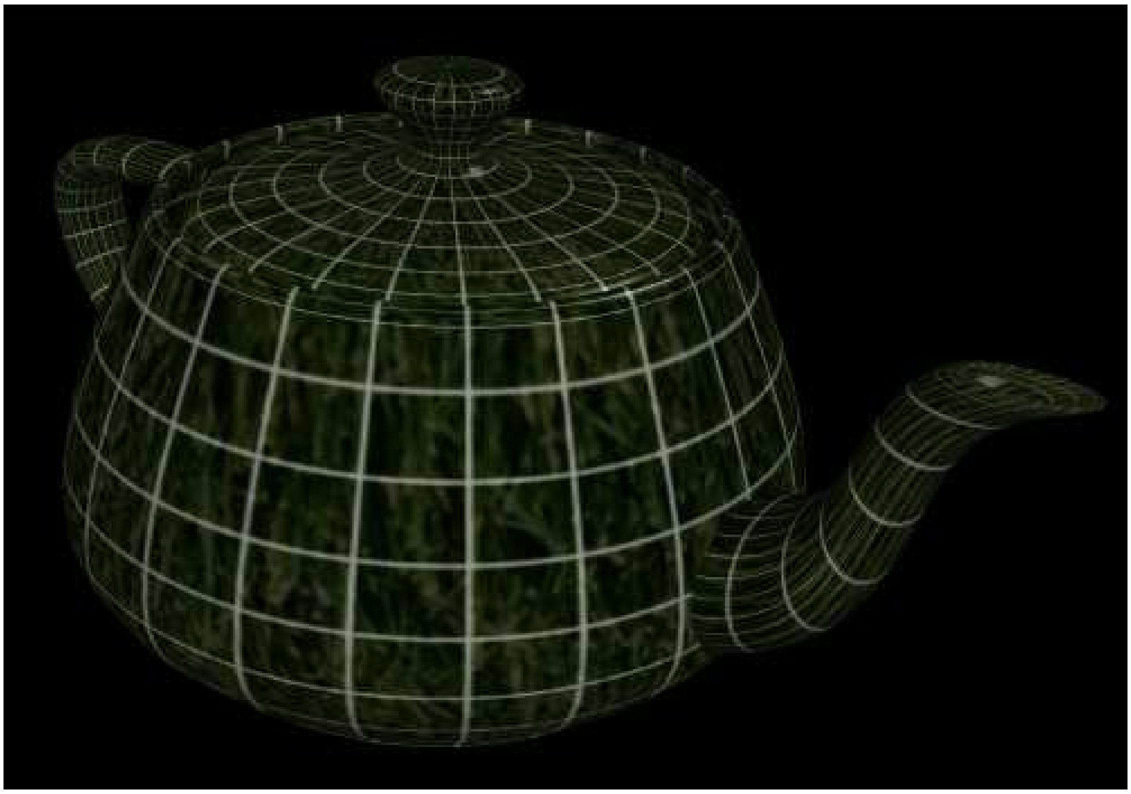

Figure 1: The grid/grass texture on this model is a result of employing the exact same bivariate raster texture for all four (body, handle, spout, and cap) surfaces of the Utah teapot. Notice the way the grid is distorted on the spout and the cap.

intersection curve in the parametric space of S as $C(t) = (u(t), v(t))$. The 3-space Euclidean representation of the curve of intersection can be computed as the composition $S(C(t)) = S(u, v) \circ C(t)$. $S(C(t))$ from direction $\mathcal{V}$, is seen as a straight line (the horizontal boundary of the mapped texture $\mathcal{T}$) along the Y axis.

In addition to seeing the boundary of $\mathcal{T}$ as a straight line, the speed along the line should be constant. The curve $C(t(s))$ provides a reparameterization for $C(t)$ that is of constant speed for the projection of $S(C(t))$ on the Y axis. The proper constant speed reparameterization for the projection of $S(C(t))$ on the Y axis can be computed by solving for the inverse of $y(C(t)) = y(u(t), v(t)) = s$.

In general, $C(t)$ can assume an arbitrary shape in the parametric space of S. Thus, after $C(t(s))$ has been derived we will need to warp one boundary of the texture map of $\mathcal{T}$ to fit

$C(t)$, in the parametric space of S.

Compute a similar surface-plane intersection for the other three boundary lines of the required mapped texture, $\mathcal{T}$, resulting in four intersection curves $C_i(t),\ 0 \leq i < 4$. Then, for the boundary of a square shaped texture, $\mathcal{T}$, to show up square from viewing direction $\mathcal{V}$, one needs to warp the square image of the texture to fit the four intersection curves, $C_i(t),\ 0 \leq i < 4$.

One can compute the surface-plane intersections by exploiting Boolean operation tools: Position a two-dimensional rectangular frame in an YZ-parallel plane sufficiently far away from the textured model, such that no intersection between the plane and the textured model occurs. Sweep the frame along the $-X$ direction. The front-most intersection of the rectangular tube with the original surface, S, provides the four boundary curves, $C_i(t),\ 0 \leq i < 4$, of the texture in the parametric space of S.

Constraining only the four boundaries provides no guarantee as to the behavior of the texture mapping in any interior location of the domain. However, assuming all mappings concerned are continuous and a bounded Lipschitz condition on S is provided, one can clearly bound the expected error by adaptively introducing curves into the interior of the domain to guide this inverse transformation. We refer to this network of curves as *guiding curves*. A sufficiently fine network of guiding curves will provide an arbitrarily precise isometric texture mapping throughout the approximated domain.

Clearly, one should avoid such costly computations when real-time rendering is required. Alternatively, if one could pre-compute a warped bivariate texture mapping that preserves the shape of the square as seen from direction $\mathcal{V}$, the real-time rendering will suffer no penalty, thus enabling the texture mapping of the pre-warped raster image in a conventional way and with no delays.

The rectangular texture should be warped to fit the guiding curves in order for it to be seen rectangular from the $+X$ direction. An image warping technique (see Section 2.2) that takes a rectangular mesh into the domain that is bounded by the guiding curves of the

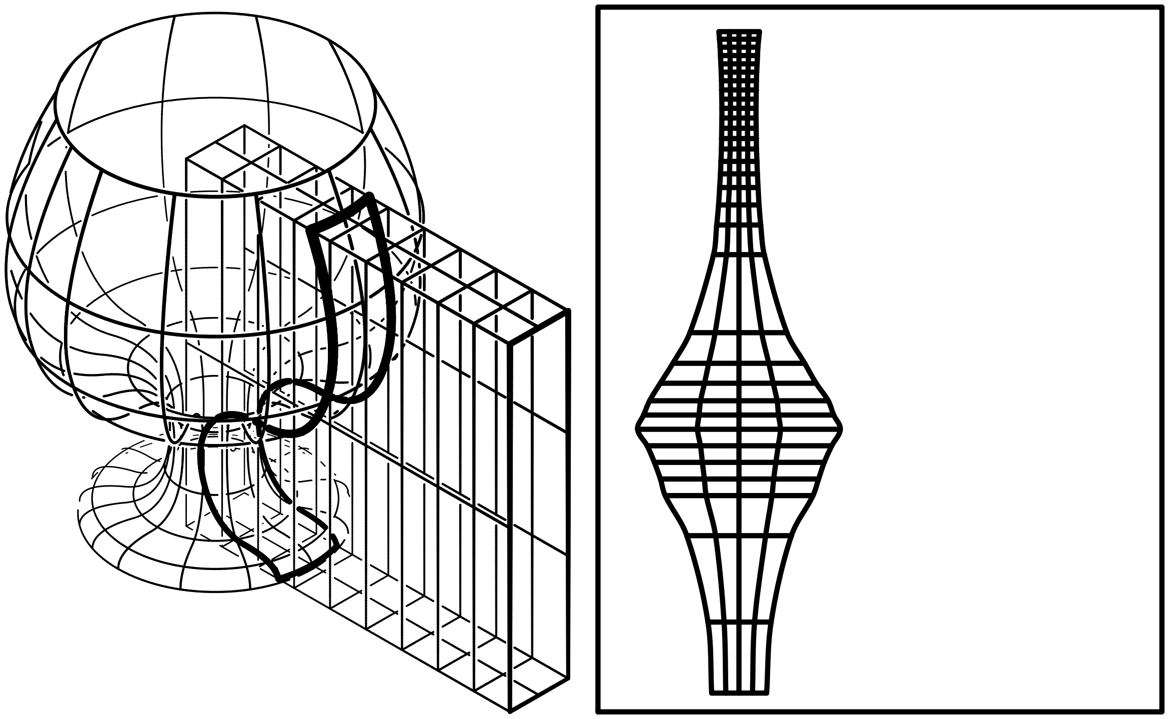

Figure 2: **Left:** A rectangular frame is swept in space in the $+X$ direction and the resulting tube is intersected with the glass surface. **Right:** The corresponding intersection curves in the parametric space of the glass are shown, including several interior guiding curves, forming a network of size 4×32.

rectangular tube and S, in the parametric space of S, will satisfy the isometry constraint on the boundary of $\mathcal{T}$.

Figure 2 illustrates the first part of the procedure. A rectangular frame is swept along the $-X$ direction and intersected with a free-form surface of a glass. Additional interior curves are also constructed by intersecting the surface of the glass with parallel planes, forming a network of guiding curves of size 4×32. The intersection curves are computed using regular Boolean operation tools, extracting the intersection curves in the parameter space of S, the glass.

2.2 Image Warping

Image warping techniques are computer graphics/image processing procedures that have gained increasing popularity in recent years. In general an image warp is a 1–1 continuous transformation of the plane (or a bounded usually simply connected subset of the plane) onto itself with grey-level or color associated with each point transformed accordingly. This technique is currently widely used as an image registration tool [11, 14] and for the creation of special effects in computer graphics [4, 3].

In the sequel, we will be (pre)-warping a parametric domain defined and bounded by a network of guiding curves onto a domain with equivalent topology. This leads us naturally to the notion of grid distortion [1] – a warping procedure first suggested by D. Thompson [21] for the investigation of related forms in comparative anatomy.

As opposed to the well-known technique of point-to-point warping, in which the mapping is governed by the movement of a finite, and usually small, set of points [20, 7, 8, 19], grid-distortions are controlled by the mapping of equivalent sets of curves arranged in a grid structure. In most practical cases, the grids are of a regular rectangular structure. However, regularity is not essential, and only minor geometric restrictions need to be adhered to [2]. The resulting warp is guaranteed to be parametrically smooth (C^1) throughout, and geometrically smooth (G^1) on the guiding curves themselves. It is designed to be as smooth as possible in the sense that the mapping function minimizes (in each coordinate) the well known thin-plate warping functional [7]

$$J(f) = \iint_{\Omega} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)\, dx\, dy, \tag{2}$$

where Ω is any one of the patches defined by the bounding curves. This implies that the warp will be, in a certain sense, the simplest possible. For example, it results in an affine mapping if the geometry of the guiding curves allows. Finally, the curves composing the grid are regarded as point-sets, i.e. they are not assumed to belong to any particular algebraic family, and thus only geometric properties of the curves are used.

Minimizing the functional (2) is equivalent (under suitable boundary conditions) to solving the biharmonic equation $\Delta^2 f = 0$ [10]. This is an elliptic fourth-order partial differential equation, and as such admits to extremely efficient numerical solutions [15].

Figure 4 demonstrates the role of the image warper, and continues the example of Figure 2.

Figure 3: A naive texture mapping of a rectangular shape resulting in a distorted shape once mapped to the Euclidean 3-space parametric surface of the glass.

A rectangular grid of the same order as the one computed from the model is placed on the texture, and the texture plane is warped so that the the two sets of guiding curves are mapped one to another. The resulting warped texture plane is then mapped by the mapping function S to the surface.

3 Examples

In this section, we present several examples that employ the proposed scheme. In figure 3, an example of a naive texture mapping attempt is shown. A raster texture map of a rectangular shape is mapped onto a wine glass with no pre-warping. The result is a large shrinkage of the texture image along the neck of the glass, totally distorting and loosing distance ratios within the texture map.

We now constrain the rectangular shape to be seen as a rectangle from the $+X$ direction.

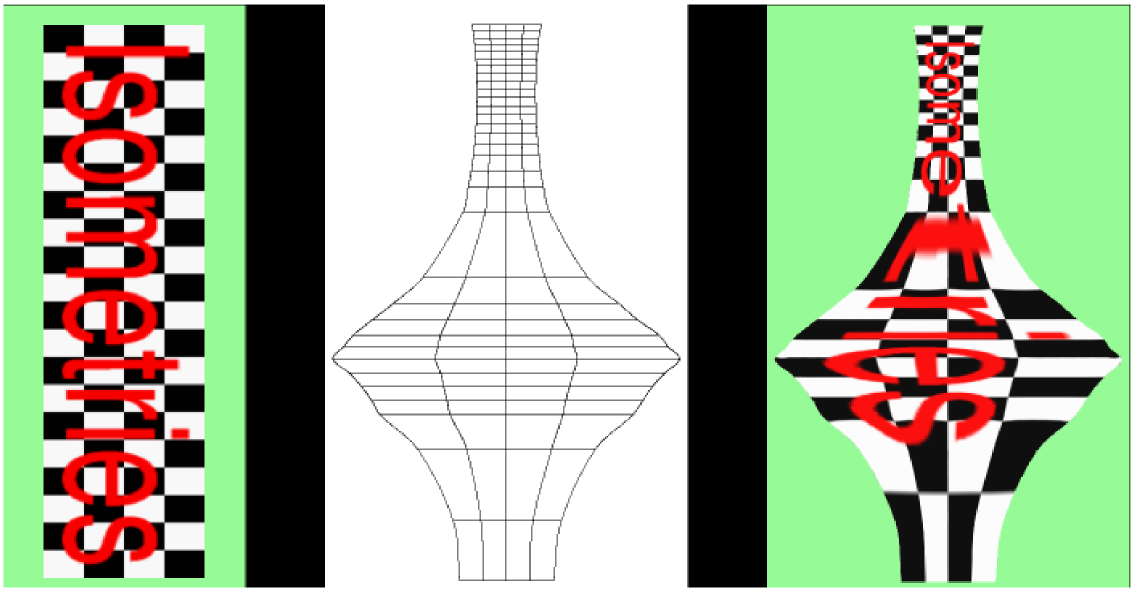

Figure 4: The pre-warped raster texture map computed on the right using an image warping tool with the aid of the network of guiding curves seen in the middle from the original texture map on the left. See also Figure 2.

In Figure 2(a), this rectangle is swept in space, and intersected with the surface of the wine glass of Figure 3. Also shown in Figure 2 (b) is a network of guiding curves in the parametric space of the surface of the glass. This network is used as input to the two dimensional image warper that pre-warps the image of the texture. Then, this pre-warped raster image can be used as input for the regular raster texture mapping.

Figures 5 and 6 shows two different approximations of the pre-warped image needed for an isometric raster texture mapping. Figure 5 exploits a network of guiding curves of size 8×8 while Figure 6 shows the result of a network of size 4×32. The second approximation is indeed coarser in one dimension while still able to deliver a much better result, hinting on the need for adaptive sampling of the network.

Finally, in Figures 7 and 8, a model of a B-58 bomber is decorated with several emblems and symbols, all preserving relative distances. These marks were placed by sweeping different rectangular tubes, with interior parallel planes, and intersecting them against the model. The resulting networks were then fed into the image warping tool to pre-warp the different symbols. The pre-warped textures of the different surfaces of the model are shown in Figure 9.

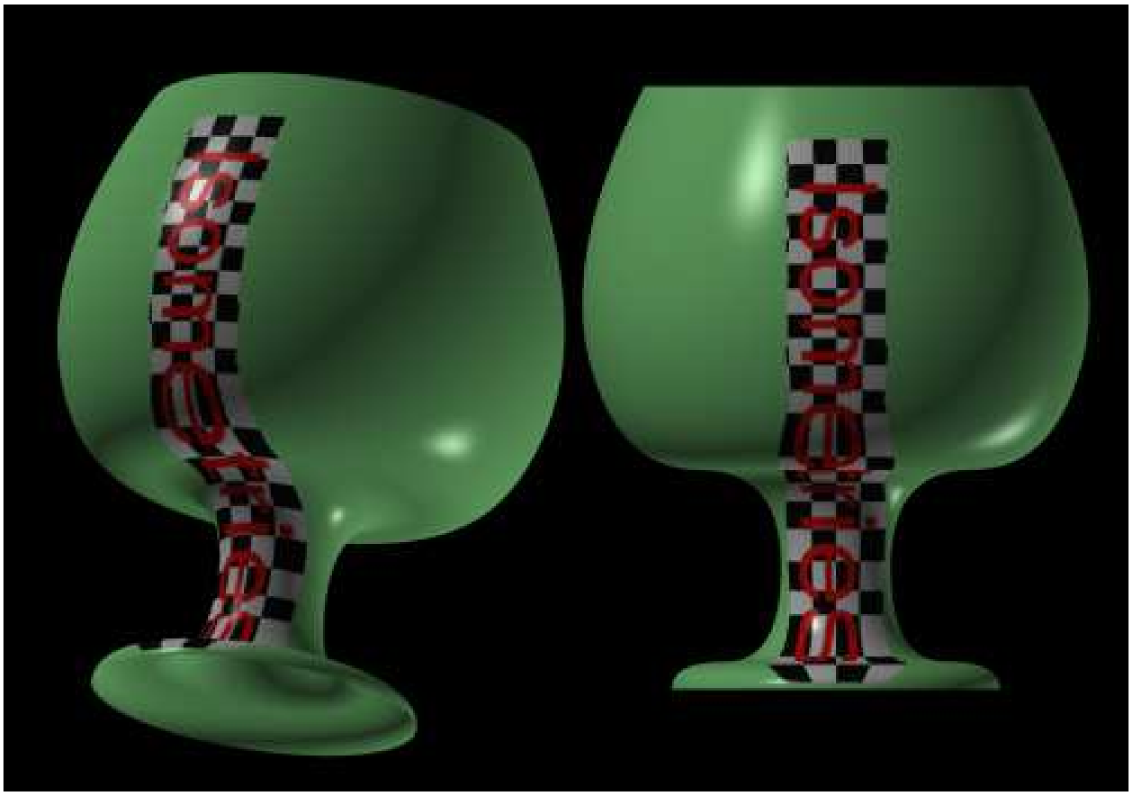

Figure 5: Approximating an isometry by using a network of guiding curves of size 8×8. Notice the distortion near the bottom and top regions of the neck of the glass. Compare with 6

All the images presented were rendered using a software based scan conversion renderer that supports bivariate raster texture mapping, while reading the pre-warped raster texture maps. The pre-warped texture maps are regular raster textures, and as such are completely transparent to the scan converter. Thus they may be as easily fed into any hardware-based scan conversion Z-buffer that supports bivariate raster texture mapping.

4 Conclusion

We have presented a method to approximate bivariate raster texture map that appears isometric from a particular viewing angle with the aid of a network of guiding curves. While in this work the guiding curves are automatically extracted using Boolean operations, any method that generate guiding curves may fit into the framework presented. Specifically, consider the application of parallel curves. Measure distances between two points on the surface

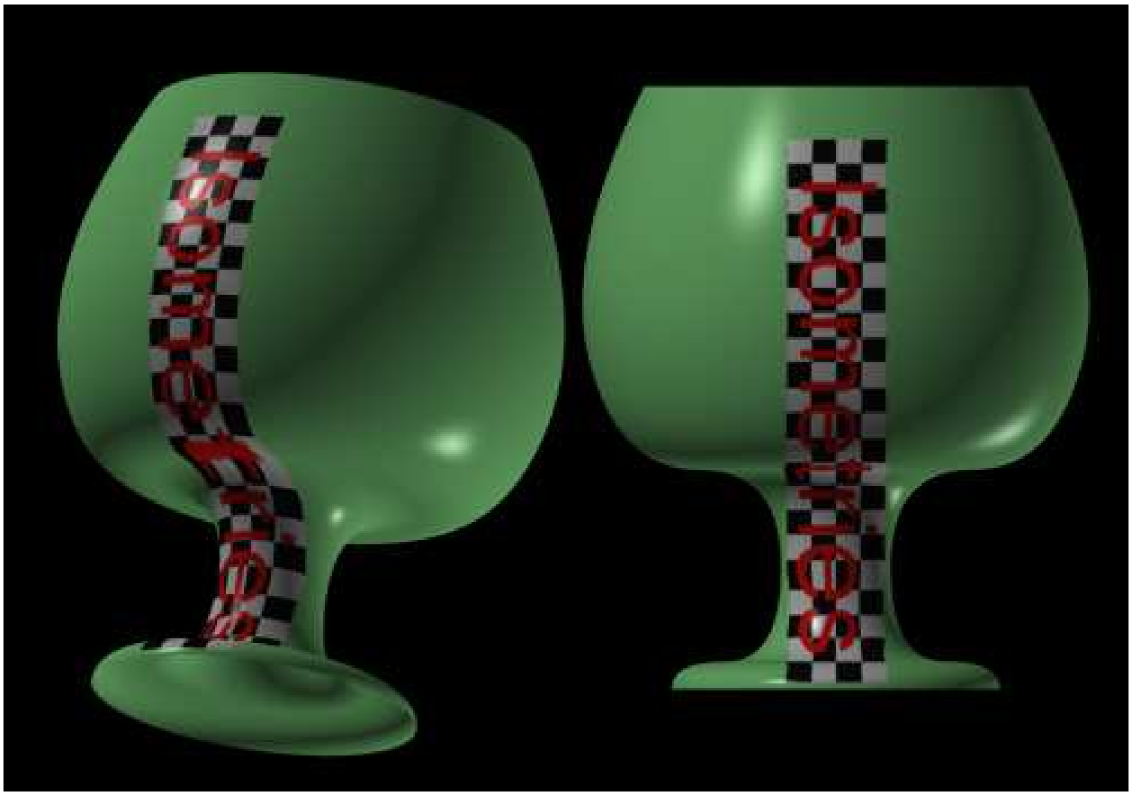

Figure 6: Approximating an isometry by using a network of guiding curves of size 4×32. Compare with 5

Figure 7: Side view of a B-58 bomber with various emblems and symbols, all preserved by locally approximating an isometric mapping.

Figure 8: General view of a B-58 bomber with various emblems and symbols, all preserved by locally approximating an isometric mapping.

S along geodesic curves and measure distances between two curves using the Hausdorff metric [16]. Then, a rectangular domain with two vertical and two horizontal boundaries can be possibly mapped into a strip on S with two pairs of parallel curves, that are not necessarily orthogonal. If these curves can be computed, then the approach proposed here can be exploited to pre-compute the necessary pre-warped raster texture for a parallel constraint preservation. This parallel constraint will converge to a real distance as well as angle isometry on developable surfaces. Furthermore, any set of guiding curves that preserves some prescribed property may be employed in a similar manner.

The adaptive computation of the network of guiding curves is a non-trivial task. The first fundamental form [9] of S plays a major role in how a surface S locally behaves as an isometry. The magnitude of the two first partials of S as well as the angle between the two partials can serve as indicators as to the quality of S in a local neighborhood and hence can hint as

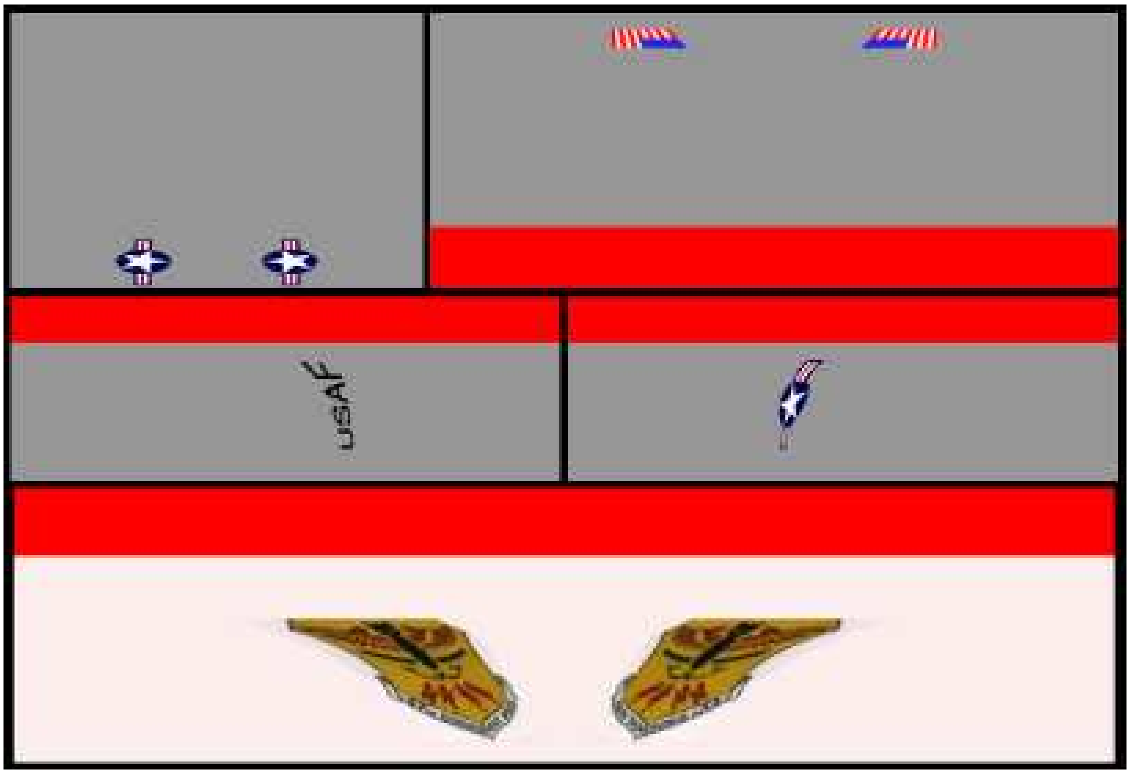

Figure 9: The prewarped texture maps of the B-58 bomber. The top row contains the two texture maps of the front (top right) and back (top left) sides of the fuselage. The middle row contains the two pre-warped textures of the wings while the bottom row contains the texture map of the rudder.

to the necessary refinement of the network that is required in this neighborhood. Symbolic computation of the magnitude square of the partials [12] as well as the angle between them can provide global bounds on the local behaviors of S as well as a Lipschitz bound.

5 References

[1] N. Arad. Designing and implementing a grid-distortion mapping based on variational principles. *Computer Graphics Forum*, 13(3):C–259–C–270, 1994. Eurographics '94 conference proceedings.

[2] N. Arad. Grid distortion on non-rectangular grids, 1995. Submitted for publication.

[3] N. Arad and D. Reisfeld. Image warping using few anchor points and radial functions. *Computer Graphics Forum*, 14(1):35–46, March 1995.

[4] T. Beier and S. Neely. Feature-based image metamorphosis. *Computer Graphics*, 26(2):35–42, 1992. Siggraph '92 conference proceedings.

[5] C. Bennis, J. M. Vezien, and G. Iglesias. Piecewise surface flattening for non-distorted texture mapping. *Computer Graphics*, 25(4), 1991. Siggraph '91 conference proceedings.

[6] J. F. Blinn. Simulation of wrinkled surfaces. *Computer Graphics*, 12(3):286–292, 1978. Siggraph '78 conference proceedings.

[7] F. L. Bookstein. Principal warps: Thin-Plate Splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:567–585, 1989.

[8] P. Borrel and A. Rappoport. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Computer Graphics*, 13(2):137–155, April 1994. Special issue on interactive sculpting.

[9] M. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[10] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 4 of *Studies in Mathematics and its Applications*. North-Holland, Amsterdam, 1978.

[11] S. Edelman, D. Reisfeld, and Y. Yeshurun. Learning to recognize faces from examples. In *Proceedings of the 2nd European Conference on Computer Vision*, pages 787–791, Santa Margherita Ligure, Italy, May 1992.

[12] G. Elber. Hybrid symbolic and numeric operators as tools for analysis of freeform surfaces. In B. Falcidieno and T. L. Kunii, editors, *Working Conference on Geometric Modeling in Computer Graphics (IFIP TC 5/WG 5.10)*, Genova, 1993.

[13] G. Elber. Model fabrication using surface layout projection. *Computer Aided Design*, 27(4):283–291, April 1995.

[14] J. Flusser. An adaptive method for image registration. *Pattern Recognition*, 25:45–54, 1992.

[15] G.Strang. *Introduction to applied mathematics*. Wellesley-Cambridge press, Wellesley, Massachusetts, 1986.

[16] J. R. Munkres. *Topology, a first course*. Prentice-Hall, Inc., 1975.

[17] D. R. Peachey. Solid texturing of complex surfaces. *Computer Graphics*, 19(3):279–286, July 1985. Siggraph '85 conference proceedings.

[18] K. Perlin. An image synthesizer. *Computer Graphics*, 19(3):287–296, July 1985. Siggraph '85 conference proceedings.

[19] D. Ruprecht and H. Müller. Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, 15(2):37–43, March 1995.

[20] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, 20(4):151–160, 1986. Siggraph '86 conference proceedings.

[21] D. W. Thompson. *On Growth and Form*, volume 2. Cambridge University Press, second edition, 1942.