1

Algorithms for texture mapping onto planar and curved surfaces

Chegu Vinod and Vipin Chaudhary

TR-93-11-11

[1]This work was supported by a research grant from G M R

Algorithms for texture mapping onto planar and curved surfaces

Chegu Vinod and Vipin Chaudhary

Abstract

This article gives a brief review of the fundamentals of the texture mapping, and presents a brief survey of the algorithms used to map a texture onto an arbitrary surface without distortion. Algorithm for *texture mapping* a scanned image onto curved surfaces using the piecewise flattening of the parametric surfaces is discussed. The approach, as cited from a recent research article, is based on concepts from differential geometry, where surfaces represented by isoparametric lines are projected locally onto a texture plane in a constructive manner, maintaining the geodesic curvature and arc length and then mapped on to the texture plane. The texture is then mapped on the projected surface. The algorithm has been implemented, with suitable ramifications using the visualization tools at General Motors Research Laboratories.

Keywords : Image synthesis, reaction diffusion, geodesic curvature, arc length, illumination mapping.

I. Introduction

Texture mapping is a shading technique for realistic image synthesis in which a texture is mapped on to a surface in the three dimensional scene. It adds realism to the scene being rendered by incorporating more information, at the cost of a modest increase in the rendering time. It is one of the most popular techniques in computer graphics and image processing and several algorithms have been proposed for it.

Texture mapping means, mapping of a function onto a surface in 3D. The domain of the function can be one, two or three dimensional, and it can be represented by either an array or as a mathematical function. The source image (texture space [u,v]) is mapped onto a surface (object space [x,y,z]), which is then mapped onto the destination image (screen space [x,y]), by the viewing projection.

*This work was supported by a research grant from G M R

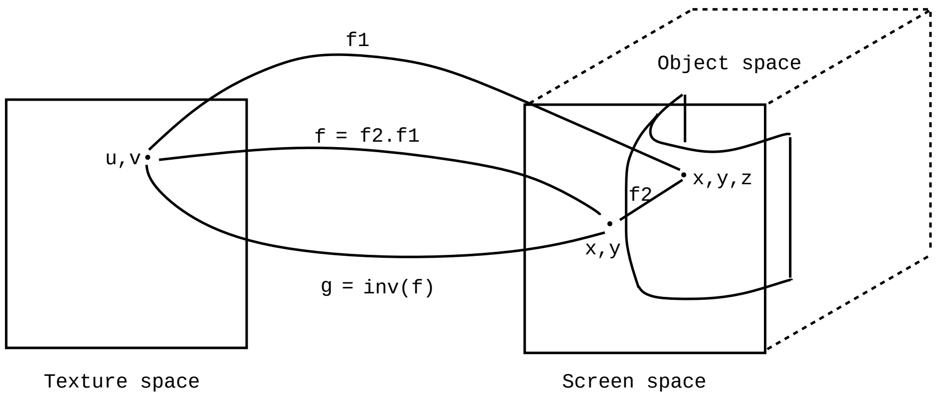

Figure 1: *Mapping functions*

The mapping techniques have been applied to modify various parameters. Surface color, specular reflection, normal vector perturbation, specularity, transparency, diffuse reflection, shadows are a few to mention.

Catmull [2] first introduced texture mapping and developed a subdivsion algorithm. Blinn and Newell [3] proposed an algorithm which calculates the inverse image of each pixel on the output screen and assigns the average intensity of the inverse image to the pixel. Catmull and Smith proposed a 2 pass algorithm which works in scanline order and can be implemented by video rate hardware if a proper inverse mapping function for the 2nd pass is available. A new technique based on the concepts of differential geometry was introduced [1]. In this approach isoparametric curves of a uniformly sampled surface in mapped onto a plane in a constructive fashion with the preservation of geodesic curvature preservation at each point.

This report is organized as follows, Section II gives a review of geometric mapping techniques for warping the texture onto the surface followed by a brief note on filtering necessary for preventing the artifacts due to aliasing in Section III. Section IV starts with a mention of the various algorithms used for mapping the texture onto arbitrary surfaces and explains the algorithm used for implementation. A brief note on the importance of illumination mapping follows in Section V.

II. Geometric mapping techniques

The mapping from a texture to a screen space is split into two phases [5], first is the surface parameterization that maps texture space to object space, followed by the standard modelling and viewing transformations that map object space to screen space, typically with a perspective projection. These two mappings are composed to find the overall 2D texture space to 2D screen space mapping, and the intermediate 3D space is often forgotten.

There are three general approaches to drawing a texture mapped surface, a scan in screen space, a scan in texture space, and two pass methods the three algorithms are given below

1. *Screen scanning :* Screen order scanning also known as the inverse mapping is one

of the most popular method. For each image in the screen space, the pre-image of the pixel in the texture space is found and its area is filtered. This method must be prefered when the mapping is invertible and the texture can be stored in the random access memory.

2. *Texture scanning :* Texture order is a simpler alternative as there is no need to find the inverse mapping function. However uniform sampling in the texture space does not guarantee uniform sampling of the screen space except for affine mappings. Adaptive sampling of the texture is used when the mappings are nonaffine. Texture order is an alternative when the texture to screen mapping is difficult to compute or when the texture has to be read sequentially from a tape and will not fit in the memory.

3. *Two pass method:* Two pass methods decomposes a 2D mapping into two 1D mappings with the first pass applied to to the row of an image and the second pass applied to the columns. These methods work well for affine or perspective mapping where the warps for each pass are linear or rational linear functions. Because the mapping and filter are 1D they are amenable to stream processing techniques such as pipelining. Two pass methods are preferable when the source image cannot be accessed.

A. *Parameterization for differrent types of surfaces*

Mapping a 2D texture onto a surface in 3D requires a parametrization of the surface. This comes naturally for surfaces that are defined parametrically. Non parametric surfaces which are usually defined implicitly can be parametrized by using the surface coordinates as in standard texture mapping; by the direction of the normal vector or a light ray as in illumination mapping; or by spacial coordinates for objects that should look as if they have been carved out of solid material.

Taking the simplest case for texture mapping : planar polygons , let us look at the parameterization, and later at the composite mapping scheme used. A triangle can be easily parametrized by specifying the texture space coordinates (u, v) at each of its three vertices. This defines an affine mapping between the texture space and 3D object space; each of x, y and z has the form $Au + Bv + C$. For polygons with more than three sides, non linear functions are needed on general, and one must decide whether the flexibility is worth the expense. The alternative is to assume linear parameterizations and subdivide into triangles where necessary. One non linear parametrization that is sometimes used is the bilinear patch:

$$[x_o \;\; y_o \;\; z_o] = [uv \;\; u \;\; v \;\; 1] \begin{bmatrix} A & E & I \\ B & F & J \\ C & G & K \\ D & H & L \end{bmatrix}$$

which maps rectangles to planar or nonplanar quadrilaterals. This parameterization has the strange property of preserving lines and equal spacing along vertical and horizontal lines but not along the diagonals. The use of this parameterization for plane quadrilaterals is not recommended, however, since inverting it requires the solution of quadratic equations. A better parameterization for planar quadrilaterals is the prespective mapping :

$$[x_o w_o \;\; y_o w_o \;\; z_o w_o \;\; w_o] = [u \;\; v \;\; 1] \begin{bmatrix} A & D & G & J \\ B & E & H & K \\ C & F & I & L \end{bmatrix}$$

where w_o is the homogeneous coordinate that is divided through to calculate the true object-space coordinates. Object coordinates x, y, z are thus of the form

$$\frac{Au + Bv + C}{Ju + Kv + L}$$

The perspective mapping preserves lines at all orientations but sacrifices equal spacing. Affine mappings are the subset of perspective matrices for which $J = K = 0$. The persepective mapping can be used whether or not the projection is perspective or not.

B. *Perspective projection of polygons*

A naive method for texture mapping in perspective is to linearly interpolate the texture coordinates u and v along the sides of the polygon and across each scan line, much as Gourdad or Phong shading is done. Linear interpolation will not give the proper effect of nonlinear foreshortening it is not a rationally invariant, and the error is evident in animation. The correct solution, however is to replace linear interpolation with the true formula, which requires a division at each pixel. In fact, Gourdad and Phong shading in perspective, which are usually implemented with linear interpolation, share the same problem, but the errors are so slight that they are rarely noticed.

Perspective mapping of a planar texture can be expressed using homogenous matrix notaion

$$[xw \;\; yw \;\; w] = [u \;\; v \;\; 1] \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix}$$

This mapping is analogous to the more familiar 3D perspective transformation using 4×4 homogeneous matrices. The inverse of this mapping (calculated using the adjoint matrix) is of the same form :

$$[uq \;\; vq \;\; q] = [x \;\; y \;\; 1] \begin{bmatrix} a & d & g \\ b & e & f \\ c & f & i \end{bmatrix} =$$

$$[x \;\; y \;\; 1] \begin{bmatrix} EI - FH & FG - DI & DH - EG \\ CH - BI & AI - CG & BG - AH \\ BF - CE & CD - AF & AE - BD \end{bmatrix}$$

The composition of two perspective mappings is also a perspective mapping. Consequently a plane using perspective parameterization that is viewed in perspective will have a compound mapping of the perspective form. For screen scanning we compute the u and v from x and y as follows:

$$u = \frac{ax + by + c}{gx + hy + i}, \qquad v = \frac{dx + ey + f}{gx + hy + i}$$

Perspective mapping simplifies to the affine form $G = H = g = h = 0$, which occurs when the surface is parallel to the projection plane.

III. Filtering

The filtering process involves resampling the screen grid after calculating the mapping functions. Aliasing results when a signal has unreproducable high frequencies. In texture mapping it becomes evident on high contrast or high frequency textures. The common approaches suggested for getting over the aliasing problem are point sampling at high resolution and low pass filtering before point sampling. Adaptive sampling is used to overcome the artifacts due to uniform sampling at high frequency regions of the texture. Supersampling is done is done in the high frequency region. The other suggested approach is based upon low pass filtering before sampling. When a signal is warped and resampled the following steps must be performed,

1. Reconstruct continuous signal from input samples by convolution.
2. Warp the absicca of the signal.
3. Low pass filter the signal using convolution.
4. Resample the signal at the output sample points.

For affine image warps, the filter is space invariant; the filter shape remains constant as it moves across the image. Non linear mappings require space-variant filters, whose shape varies as they move across the image. Excellent reference for filtering mechanisms can be found in [5]. Filtering mechanisms have been implemented as a hardware feature in the recent SGI-3D VGX graphics workstations.

IV. Texture mapping on to curved surfaces

Texture mapping onto an arbitrary surface is not a trivial problem as compared with that of the planar surfaces as the texture undergoes streching at the curved surface, which makes it appear unrealistic. Several solutions have been suggested to overcome this problem. This section mentions the various approaches that were used for mapping a given texture onto curved surfaces and explains the algorithm being used for implementation. A common approach as suggested by Catmull [2] and further refined by Bloomenthal was to define a mapping from the rectangle to the natural coordinate system of the target object surface taking care that the edges of the textures match, when mapping onto an object defined by several patches. Another approach as suggested by Bier and Sloan [4] involved mapping the texture onto an intermediate surface in the euclidean R^3 space and then projecting onto the given surface of the target object. Other approaches suggested, include making use of the polygonal nature of the graphical model and try to unfold the texture on the plane and compute the texture placement based on the average of the unfolded positions of each vertex.

Each of the above mentioned approaches cause texture distortion due to the fact that there is no common mapping function from the texture space to the surface of the object. The above approaches also require sufficient amount of user intervention. Texture synthesis and their growth on the surface using the principles of Reaction diffusion [7][8] are often thought of as an elegant alternative to overcome the problems encountered in mapping scanned textures onto any arbitrary surface.

A. *Definitions*

The algorithm that is being used for the implementation is based on the concepts from differential geometry [11], which express the local behavour of the curve using a local frame, known as the *Frenet frame* $(\vec{X}, \vec{t}, \vec{m}, \vec{b})$, with reference to its co-ordinate system. Let $\vec{X}$ be a point on the isoparametric curve given by the arc length parameterization as :

$$X\vec{(s)} = \begin{bmatrix} x(s) \\ y(s) \\ z(s) \end{bmatrix}, \quad s \in [0, a] \subset R$$

where x, y, z are the cartesian coordinates of the curve point. $t\vec{(s)}$,$m\vec{(s)}$,$b\vec{(s)}$ are the *tangent vector*, *main normal vector* and the *binormal vector* at s and defined as follows :

$$\vec{t} = \frac{d\vec{X}}{ds} \qquad \vec{m} = \frac{\frac{d^2\vec{X}}{ds^2}}{\| \frac{d^2\vec{X}}{ds^2} \|} \qquad \vec{b} = \vec{t} \times \vec{m}$$

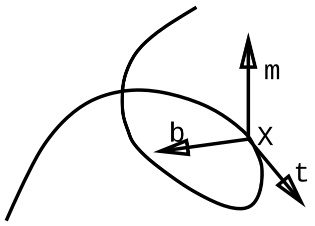

Figure 2: *Frenet frame*

The plane containing the curve at s is called as the osculating plane $O(s)$ generated by $t\vec{(s)}$ and $m\vec{(s)}$. The *curvature* and *torsion* at s are defined as the angular velocities of the tangent plane and the osculating plane respectively.

The surface S being considered is defined by an arc length parameteric function from R^2 to R^3 as :

$$X(\vec{u}, v) = \begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix}, \quad (u,v) \in [a,b] \times [a,b] \subset R^2$$

where x, y, z are the cartesian coordinates of a surface point and are second order differentiable functions of the parameters u, v.

Let vector $\vec{n}$, be defined as the normal vector at the surface point $X(\vec{u}, v)$.

$$\vec{n} = \frac{\frac{\partial X(\vec{u},v)}{\partial u} \times \frac{\partial X(\vec{u},v)}{\partial v}}{\| \frac{\partial X(\vec{u},v)}{\partial u} \times \frac{\partial X(\vec{u},v)}{\partial v} \|}$$

where, $\frac{\partial X(\vec{u},v)}{\partial u} \times \frac{\partial X(\vec{u},v)}{\partial v} \neq 0$, i.e., the isoparametric curves of the surfaces being considered are nowhere tangent to each other. The tangent plane to the surface at a point $\vec{X}$ is given by $T_p = (\vec{X} - \vec{Y}) \cdot \vec{n}$, where $\vec{Y}$, is any point in the plane T_p.

The *Geodesic curvature* of the curve C_s belonging to the surface S at a point $\vec{X}$, is defined as the norm of the tangential acceleration at $\vec{X}$ according to the arc length parameter :

$$| k_g | = \| t'_g \|$$

where k_g corresponds to the curvature of C_s seen from a view point attached to the surface S. It is different from the main curvature k. C_s is said to be *geodesic* at a point $\vec{X}$ iff. k_g is nil at $\vec{X}$ and C_s is called geodesic if it is geodesic at every point. One of the necessary and sufficient condition for C_s to be geodesic at a point $\vec{X}$ is that the main normal $\vec{m}$ of the curve C_s at $\vec{X}$ is parallel to the normal $\vec{n}$ to the surface S at X. At any point $\vec{X}$ the local projection of the curve C_s on the tangent plane along the normal vector to the surface S provides a straight line if C_s is geodesic, non-zero curvature at $\vec{X}$ on the planar curve otherwise.

The Geodesic curvature k_g of a curve C_s belonging to a surface S at a point X is equal to the curvature at X of the planar curve C_{T_p} obtained by locally projecting C_s onto the tangent plane (T_p) along the normal vector to surface S at point X. It can be intutively thought of as the angular velocity of tangents to the resulting projected curve C_{T_p} according to the arc length s.

B. *Algorithm[1]*

The surface to be textured is sampled into an uniform grid of 3D points along the isoparametrics distance between sucessive samples can be approximated by eucledian distance. The sampling should be refine inorder to ensure that the distance between the sample points along the surface curves are approximated by euclidean distance. An initial curve C is selected (choice of the curve dependent upon the location where the texture is expected to be

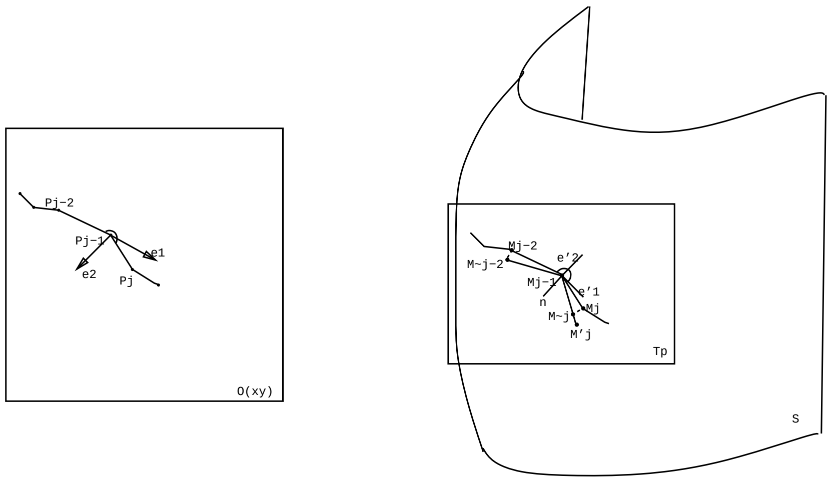

Figure 3: *Piecewise flattening*

relatively more distorted), and let n+1 be the number of sample points M_i, $i = 0, 1 ... n$. Let n_i and T_{p_i} , be the normal and the tangent plane to the surface at the point M_i.

The flattening algorithm used as given below.

1. The first curve segment $M_0 M_1$ onto a segment $P_0 P_1$ in the plane O_{xy} such that $d(M_0 M_1) = d(P_0 P_1)$. It is sufficient to fix an initial point P_0 and a direction in the plane.

2. The following process is iterated to compute the values of P_j , $2 \leq j \leq n$,

   (a) Project M_j and M_{j-2} onto the tangent plane to the surface at M_{j-1}. This provides two points in $T_{p_{j-1}}$ called $\tilde{M_j}$ and $\tilde{M_{j-2}}$ and given by the formulas,

   $$\tilde{M_j} = M_j + ((M_{j-1} - M_j) \cdot n_{j-1}) \cdot n_{j-1}.$$

   $$\tilde{M_{j-2}} = M_{j-2} + ((M_{j-1} - M_{j-2}) \cdot n_{j-1}) \cdot n_{j-1}.$$

   (b) The point $\tilde{M_j}$ on the plane $T_{p_{j-1}}$ is transformed to a point M'_j such that the $d(M_j, M_{j-1}) = d(M'_j, M_{j-1})$.

   $$M'_j = M_{j-1} + \frac{\| M_j - M_{j-1} \|}{\| \tilde{M_j} - M_{j-1} \|} \cdot (\tilde{M_j} - M_{j-1})$$

   (c) Using the precomputed values of the points P_{j-2}, P_{j-1} the desired point P_j on the plane O_{xy} that simultaneously preserves the angle between $\tilde{M_{j-2}} M_{j-1}$ and $M_{j-1} M'_j$, and the distance $d(M'_j, M_{j-1})$.

   To obtain P_j we compute the coordinates (x'_1, x'_2) of M'_j according to the local orthogonal frame (M_{j-1}, e'_1, e'_2) in $T_{p_{j-i}}$ where e'_1 and e'_2 and the coordinates are given by the formulas :

$$e'_1 = \frac{M_{j-1} - \tilde{M_{j-2}}}{\| M_{j-1} - \tilde{M_{j-2}} \|} \quad e'_2 = n_{j-1} \times e'_1$$

$$x'_1 = (M'_j - M_{j-1}) \cdot e'_1 \quad x'_2 = (M'_j - M_{j-1}) \cdot e'_2$$

The point P_j on O_{xy} haveing the same coordinates according to the orthogonal and positive frame are given by :

$$e_1 = \frac{P_{j-1} - P_{j-2}}{\| P_{j-1} - P_{j-2} \|} = a\vec{i} + b\vec{j} \quad e_2 = -b\vec{i} + a\vec{j}$$

$$P_j = P_{j-1} + (x'_1 e_1 + x'_2 e_2)$$

where, $(O, \vec{i}, \vec{j})$ is the canonical system $O(x, y)$.

The above steps lead to curve flattening with preservation of geodesic curvature and arc lengths within the chord length approximation, as $d(P_{j-1}, P_j) = d(M_{j-1}, M_j)$ for $0 \leq j \leq n$. and since we have considered the sampling to be highly refined, tangents to the curve can be approximated with chord segments and hence the preservation of the angle θ_{j-1} results in preserving the angular velocity of the tangents to the curve into $T_{p_{j-1}}$.

The steps are repeated on the left side of the initial curve untill the left side curve's distortion metric exceeds the threshold set by the distortion metric of the initial curve. The distortion metric for each curve is defined as the mean of the errors induced on its chord segments :

$$\frac{1}{N-1} \sum_{j=0}^{N-1} \frac{| d(M_{ij}, M_{ij+1}) - d(P_{ij}, P_{ij+1}) |}{d(M_{ij}, M_{ij+1})}$$

where N is the number of sample points on the curve. The same procedure is repeated for the right hand side of the initial curve.

The processed region of the grid is then T-meshed, and locally affine interpolation , affine in each triangle is used to texture them.

V. Illumination mapping

Another type of mapping that demands special attention, from the research point of view is illumination mapping. Also known as the reflection or environment mapping it involves mapping of specular or diffuse reflection, which is quite different from the traditional mapping in two main ways. It is not associated with any paricular object in the scene but with an imaginary sphere, box or cube surrounding the scene.Unlike the usual texture maps, specular reflection maps are indexed by the direction of the reflected rayand the diffuse reflection maps are indexed by the surface normal direction. Efficient filtering techniques are used to ensure proper antialiasing in the case of illumination maps.

Illumination mappings facilitate the simulation of complex 3D lighting environments, since the time required to shade a point is independent of the number of light sources used. They can be thought of as poor man's approximation to ray tracing for specular reflection mapping and radiosity for diffuse reflection mapping of objects in the environment.

VI. Conclusion & Acknowledgements

We have presented a brief survey of the various texture mapping algortihms, and suggested the appropriate approach for mapping a given scanned texture onto a CAD modelled surface. The algorithm was cited from a recent research publication [1], where it was suggested for shoe modelling. The algorithm can be further refined and modified to make it directly compatible with the design evaluation tools in the automobile industry.

Dr.Paul J Besl and Dr.Randal C Smith contributed useful ideas and provided timely help. Thanks to Dr.Robert Tilove, to my advisor Dr.Vipin Chaudhary and to all the helpful researchers at the Analytic Process department, General Motors Research Laboratories, for their encouragement.

References

[1] Chakib Bennis, Jean-Mark Ve'zien, Ge'rard Igle'sias, "Piecewise Surface Flattening for Non-Distrorted Texture Mapping", Computer graphics (Proc. SIGGRAPH 91), Vol.25. No.4 , July 1991. pp. 237-246.

[2] Ed. Catmull and Alvy Ray Smith, "3D Transformation of images in scan line order", Computer Graphics (Proc. SIGGRAPH 80), Vol.14, No.3, July 1980, pp. 279-285.

[3] James F Blinn and Martin E. Newell, "Textures and reflection in Computer generated images", Communications of the ACM, Vol. 19, No. 19, Oct. 1976, pp. 542-547.

[4] E. Bier and K. Sloan. Two part texture mapping. IEEE Computer Graphics and Applications, September 1986, pp. 40-53.

[5] Paul S Heckbert, "Fundamentals of Texture mapping and Image Warping", Master's thesis June 1989, Computer Science division, University of California, Berkeley, CA.

[6] Paul S Heckbert, "Adaptive radiosity textures for bidirectional ray tracing". Computer graphics (Proc. SIGGRAPH 90), Vol.24. No.4 , August 1990.

[7] Greg Turk, "Generating textures on arbitrary surfaces using Reaction-Diffusion". Computer graphics (Proc. SIGGRAPH 91), Vol.25. No.4 , July 1991. pp. 289 - 298.

[8] Andrew Witkin , Michael Kass, "Reaction Diffusion Textures", Computer graphics (Proc. SIGGRAPH 91), Vol.25. No.4 , July 1991. pp. 290-308.

[9] David F Rogers," Procedural elements for Computer graphics"., Mc Graw Hill New York, 1985.

[10] Foley, J.D.,A. van Dam, S.Feiner, Hughes,J.D.,"Computer Graphics : Principles and Practice",second edition., Addison Wesley Publishing Co. 1990.

[11] G. Farin, Curves and Surfaces for geometric design. Academic Press, San Diego, Inc., 1992

[12] "Graphics Library Programming Guide" Silicon Graphics Inc.,CA. 1992.