Texture Mapping using Surface Flattening via Multi-Dimensional Scaling

Gil Zigelman Ron Kimmel
Department of Computer Science,
Technion, Haifa 32000, Israel
and
Nahum Kiryati
Department of Electrical Engineering–Systems,
Tel Aviv University, Tel Aviv 69978, Israel

February 21, 2000

Abstract

We present a novel technique for texture mapping on arbitrary surfaces with minimal distortions, by preserving the local and global structure of the texture. The recent introduction of the fast marching method on triangulated surfaces [9], made it possible to compute geodesic distances in $O(\tilde{n} \lg \tilde{n})$ where $\tilde{n}$ is the number of triangles that represent the surface. We use this method to design a surface flattening approach based on multi-dimensional scaling (MDS). MDS is a family of methods that map a set of points into a finite dimensional flat (Euclidean) domain, where the only given data is the corresponding distances between every pair of points. The MDS mapping yields minimal changes of the distances between the corresponding points. We then solve an 'inverse' problem and map a flat texture patch onto the curved surface while preserving the structure of the texture.

1 Introduction

The computer graphics community has made many attempts to solve the problem of mapping flat texture images onto curved surfaces. The main problems with most of the existing methods are that they

- Introduce large deformations and distortions to the original texture.

- Suffer from aliasing problems, usually solved by filtering that blurs and smoothes the texture.

Environment mapping [1, 2], is one technique that creates the effect of environment reflections on surfaces. It maps the original 2D texture to a sphere or a cube surrounding the surface. Then, the surface normal at each point is used to find the intersection of the reflected viewing vector with the surrounding simple object, and assigns the texture at that point to the corresponding surface point. These methods do not preserve the local area of the texture and introduce local deformations. Moreover, mapping the 2D texture onto a sphere causes distortions to begin with. In order to minimize these artifacts, one has to distort the original flat texture image before mapping onto a sphere.

Bier and Sloan [3], extended the environment mapping idea and proposed a two steps procedure. First the texture is mapped onto a simple object (preferably preserving the area) and then it is mapped from the simple object to the given surface, using, for example, the surface normal's intersection with the simple object. This method also introduces visible deformations, however, it can decrease the distortions which exist in the previous methods.

Kurzion, Möller and Yagel [4] try to preserve area. They use a cube as a simple surrounding object. For each point they find two curvature values in specially selected directions, and then change the density of the surrounding image respectively. This method is area preserving, however, it creates shear effects. It is also limited to smooth surfaces with C^2 continuity.

Arad and Elber [5], preserve the local texture area by finding, for a specific viewing direction, the four intersection curves (in the parametric space) between a swept rectangle in the viewing direction and the surface. Then they warp the square texture image to fit the four intersection curves. The texture image is warped before mapping. This method is useful in cases where one wants to map a texture on a small portion of a surface.

Bennis, Vézien and Iglésias [6], first piecewise flatten the surface and then map the texture onto each flattened part. The flattening of a region grows around an isoparametric curve selected by hand. They use a distortion metric as a control and stop the growth when the accumulated distortion exceeds a given threshold. They permit discontinuities on the mapped texture in order to minimize distortions.

This paper introduces a new mapping method that preserves both the local and the global structure of the texture, with minimal shearing effects. It enables realistic texture mapping on any given surface. Our method avoids the need for an intermediate surface and does not require any smoothness of the surface.

The proposed method is based on two numerical tools, the fast marching method [9] and multi-dimensional scaling [8]. In Section 2, we briefly review the

MDS method. Section 3 explains how the MDS can be used for texture mapping. Section 4 presents experimental results and Section 5 gives concluding remarks.

2 The Flattening Approach

MDS (Multi-Dimensional Scaling) is a set of mathematical techniques used to uncover the "geometric structure" of datasets. See e.g. [7]. For example, given a set of objects with proximity values amongst themselves, we can use MDS to create a 2D map of these objects, that is easier to comprehend or analyze. In this paper we use the MDS in a similar way. As proximity values we use the geodesic distances measured between every two points on the surface, and the resulting map represents the flattening of the curved surface. The flattening preserves the original geodesic distances quite well, as can be seen in Figure 1.

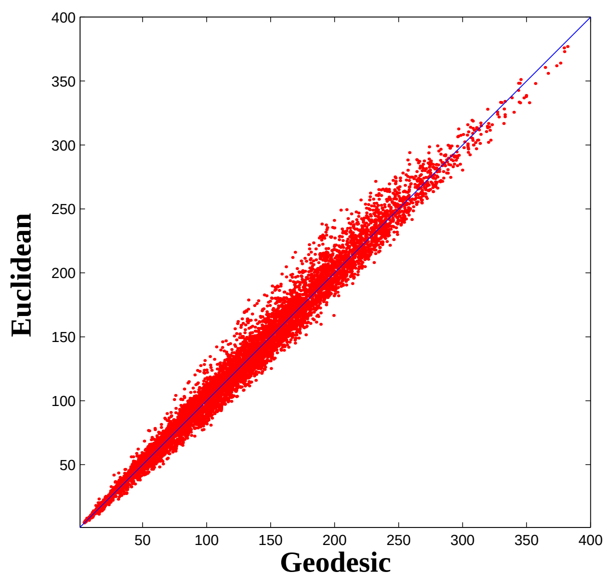


Figure 1: The error between the geodesic distance and the Euclidean distance on the flattened surface. The values were taken from the face surface shown in Figure 2. The result approximates the diagonal line, which would have been the (geometrically impossible) perfect warping result.

The input to the MDS is an $n \times n$ symmetric matrix $\mathbf{M}$. The M_{ij} element is the squared geodesic distance between point i and point j, where n is the number of points on the surface. We calculate the geodesic distances efficiently by using the fast marching method on triangulated domains introduced by Kimmel and Sethian in [9]. This method enables us to compute geodesic distances in $O(\tilde{n} \lg \tilde{n})$ where $\tilde{n}$ is the number of triangles that represent the curved surface.

Most MDS methods are based on finding the coordinates x^k, $k \in [1, ..., m]$ where m is the dimension we are interested in (in our case $m = 2$), from the given distances. One direct and simple approach presented below is known as 'Classical Scaling', see [8]. It is closely related to the singular value decomposition (SVD) method, and involves the following steps

- Compute the $n \times n$ matrix $\mathbf{M}$, as previously described.
- Apply double centering to this matrix: $\mathbf{B} = -\frac{1}{2}\mathbf{JMJ}$, where $\mathbf{J} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}'$, $\mathbf{J}$ is an $n \times n$ matrix, $\mathbf{1}$ is a $1 \times n$ column of ones, and $\mathbf{1}'$ is the transpose of $\mathbf{1}$.
- Compute the eigendecomposition of $\mathbf{B}$, $\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}'$. We are interested only in a subset containing the largest eigenvalues and their corresponding eigenvectors. For this, the 'power method' is an efficient numerical tool.
- We restrict the dimension of the solution to be two. Denote the 2×2 matrix of the first 2 largest positive eigenvalues as $\mathbf{\Lambda}_+$, and denote $\mathbf{Q}_+$ as the $n \times 2$ matrix of their corresponding eigenvectors.
- The $n \times 2$ coordinate matrix of classical scaling is given by $\mathbf{X} = \mathbf{Q}_+\mathbf{\Lambda}_+^{\frac{1}{2}}$. Row i contains the flattened coordinates of the original surface point i.

We thereby obtained an area preserving flattening of the surface. Computing the geodesic distances between every pair of vertices in a complex triangulated surface, and the eigendecomposition of the corresponding distance matrix is computationally expensive. In practice we select a sub-set of the vertices and apply the flattening procedure on this sub-set. The geodesic distance between each pair of points in this set is calculated using the complete surface model. Thus, after proper flattening of the sub-set of anchor vertices, we need to correctly interpolate the local coordinates in order to find the local map of the rest of the vertices. For example, in Figure 2.b we show the flattening of a 3D object shown in Figure 2.a.

3 Using MDS for Texture Mapping

Local area preserving mapping of a curved surface onto the texture image plane is useful for mapping with minimal deformations. After applying classical scaling to the original surface, we get a 2D flattened version of the surface. We now have the mapping of every point on the 3D surface to its corresponding 2D flattened

(a) (b)

Figure 2: a. A 3D reconstruction of a face. b. The flattened texture image of the face

point. Given a texture image we can easily map each point from the 2D flattened map to a point on the texture plane.

Next, we would like to map the texture back to the surface. The technique is straightforward. For each vertex P on the surface

- Find the corresponding 2D point in the flattened map.
- Translate the 2D coordinates to the texture image coordinates.
- Use this point's color as texture.

If the triangulation of the surface is not dense enough, we might encounter aliasing effects. These can be solved by selectively subdividing large triangles into small ones as was implemented in our experiments. Determining the texture in the newly created vertices is done by applying the same subdivision on the 2D corresponding triangle in the texture image plane, and taking the proper interpolated colors from the corresponding image points.

4 Experimental Results

We tested our technique on results obtained using a 3D laser scanner built by the first author at the Technion. The scanner creates a textured range image on a rectangular grid, where each grid point is a perspective projection of a corresponding surface point in 3D. A sub-set of the selected vertices for the MDS was chosen as a sub-grid of this range image.

In order to map the texture onto the surface with minimal distortions we take the following steps. First, we flatten the surface by classical scaling applied to the geodesic distances between a sub-set of all the surface vertices. The flattening procedure gives us a simple mapping between the plane and the surface. Since we consider only a sub-set of all the vertices we still need to locally interpolate the map for the rest of the vertices.

In our experiments we have chosen to associate each bilinear patch defined between neighboring selected vertices on the surface (P_a^S, P_b^S, P_c^S, and P_d^S in Figure 3) to a bilinear patch defined by the corresponding vertices of the flattened surface, P_a^{MDS}, P_b^{MDS}, P_c^{MDS}, and P_d^{MDS} in Figure 3. We next apply a scan conversion procedure to map the points within each planar patch to a bilinear patch in 3D. For example, the point P^{MDS} in Figure 3 is mapped to the point P^N. Finally, we project the points in the bilinear patch to the actual surface point P^S. This last step is a local procedure where in our experiments we used a simple perspective projection. A full intrinsic approach would be to use the chart defined by the local geodesic curves connecting the neighboring vertices on the surface. This chart could be easily computed via the fast marching on triangulated domains [9].

Figure 4 demonstrates how the chess-board texture fits the surface with minimal distortions. Figure 5 shows another example.

5 Conclusions

We presented a simple and general feature preserving texture mapping approach with minimal distortions. Using the fast marching method on triangulated domains we calculate geodesic distances between pairs of surface points. It enables us to achieve accurate measurements that characterize the geometry of the surface, with a reasonable computational complexity. Next, we used the simplest MDS method, known as ‘classical scaling’, to flatten the surface, and used the flattened surface to back project a flat texture image onto the curved surface. The method is computationally efficient, and does not require any assumptions, such as smoothness, of the given surface. It is also unnecessary to apply any pre-warping or deformations to the original texture image.

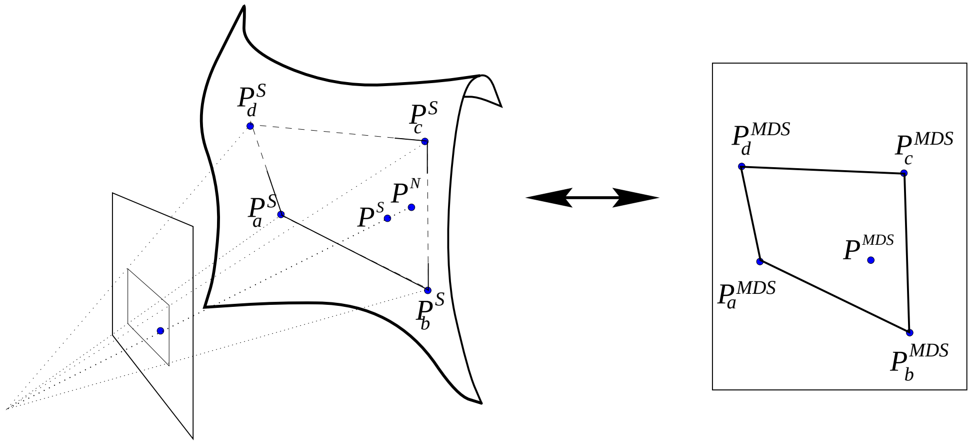


Figure 3: Left: The sub-grid rectangle‘s vertices selected for the MDS procedure. Each vertex corresponds to a surface point in 3D. Right: After flattening, we find four new corresponding points P^{MDS} which are overlaid on the texture image.

6 Acknowledgments

We thank Haim Schweitzer from the University of Texas at Dallas for raising our interest in MDS, and Gershon Elber from the Technion, Israel, for his valuable help in pointing us to related work. At TAU the research was supported in part by the Adams Super-Center for Brain Studies.

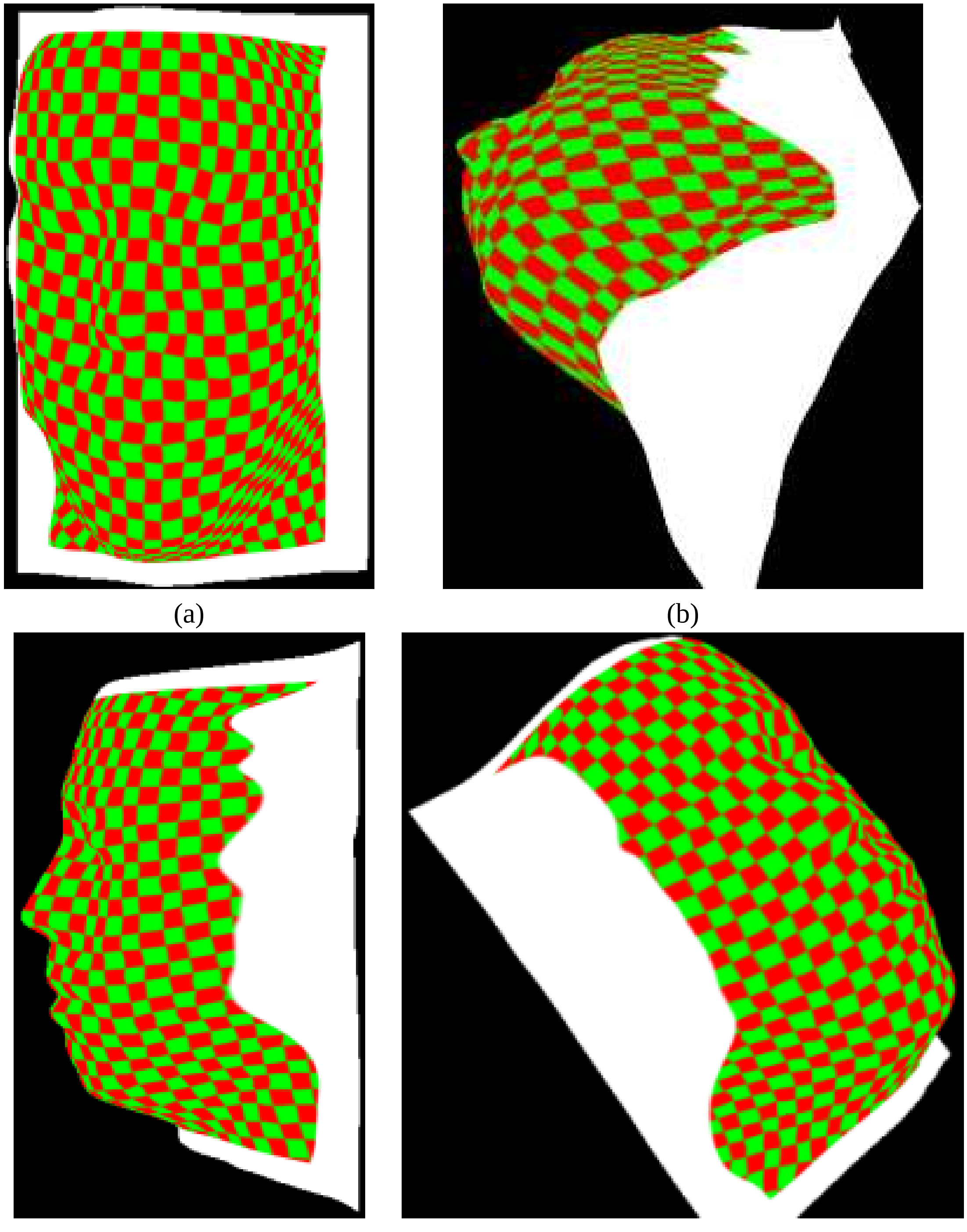
References

[1] J. F. Blinn and M. E. Newell, “Texture and Reflection in Computer Generated Images”, Comm. of the ACM, Vol 19, No. 10, Oct. 1976, pp. 542-547.

[2] N. Greene, “Environment Mapping and Other Applications of World Projections”, IEEE Computer Graphics and Applications, November 1986, pp. 21-29.

[3] E. A. Bier and K. S. Sloan, “Two-Part Texture Mapping”, IEEE Computer Graphics and Applications, September 1986, pp. 40-53.

[4] Y. Kurzion, T. Möller and R. Yagel, “Size Preserving Pattern Mapping”, Proc. of IEEE Visualization ’98, Research Triangle Park, NC, October 1998, pp. 367-373.



(a) (b) (c) (d)

Figure 4: Chess board texture mapped on the head object. Shading is turned off

(a) (b)

Figure 5: Another example of a texture mapped on the head object

[5] N. Arad and G. Elber, “Isometric Texture Mapping for Free-Form Surfaces”, Computer Graphics forum, 16(5), December 1997, pp. 247-256.

[6] C. Bennis, J. M. Vézien and G. Iglésias, “Piecewise Surface Flattening for Non-Distorted Texture Mapping”, ACM Computer Graphics, 25(4), July 1991, pp. 237-246.

[7] J. B. Kruskal and M. Wish, “Multidimensional Scaling”, Sage Publications, Beverly Hills, California, 1978.

[8] I. Borg and P. Groenen, “Modern Multidimensional Scaling, Theory and Applications”, Springer Series in Statistics, 1997.

[9] R. Kimmel and J.A. Sethian, “Computing geodesics on manifolds”, Proceedings of National Academy of Sciences, USA, 95(15), 1998, pp. 8431-8435.