ECE 842 Report
Invariants of Isometric Transformations

Wei-Yang Lin

December 16, 2004

Abstract

The aim of this report is to illustrate the issues in finding a transformation such that geodesic distance between each pair of points on the surface is maintained. There are at least three solutions for this problem. I will explain the simplest one since the others are based on the same idea. Then, the implementation results will show how it performs on synthesized data.

1 Introduction

In the past several decades, people are interested in recognizing objects in images. The basic idea is to to treat imaging process as a projection from 3-dimensional space to image plane. Then, use invariants of projection to recognize objects. The most famous example is probably *affine invariants*. People use affine transformation as the approximation of projection. Hence invariants of affine transformation can be used in recognizing 3D objects in image. In this type of methods, objects of interest are assumed to be rigid. In other words, objects won't change their shape all the time.

In many situations, objects are not only under projection but also subject to deformation. For example, face expression will change from time to time. Hence, invariants of non-rigid transformation have great importance in these situations.

In this report, I will focus on isometric transformation which is subset of the whole non-rigid transformation family. The reminder of this report is organized as follows. Section 2 presents a simple method for finding isometric invariants. Then, results of software implementation are shown in section 3. Section 4 conclude this report with some comments.

2 Invariants of Isometrical Transformation

Isometric transformation ia a length preserving transformation. In other words, there is no shearing and stretching on the surface of object during deformation. In some situations, deformation of objects can be approximated by isometric transformation. For example, varying of face expression is approximately isometric.

Zigelman et al. [6] use a surface flattening procedure to find an approximated isometric invariant. Let's consider a surface $S \in \mathbb{R}^3$ and points $\{p_1, p_2, ..., p_n\}$ on this surface. We define a n by 3 vector $\mathbf{X}$

$$\mathbf{X} = [p_1 p_2 \dots p_n]^T \tag{1}$$

and n by n matrix $\mathbf{G}$ with entries

$$g_{ij} = \text{ squared geodesic distance from } p_i \text{ to } p_j$$

We want to find points $\{q_1, q_2, ..., q_n\}$ such that the Euclidean distance from q_i to q_j equal to the geodesic distance from p_i to p_j. Let

$$\mathbf{Y} = [q_1 q_2 \dots q_n]^T$$

Let matrix $\mathbf{E}$ denote the squared Euclidean distances between each pair of points in $\mathbf{Y}$, that is

$$\mathbf{E}_{ij} = \text{ squared Euclidean distance from } q_i \text{ to } q_j$$

Then, $\mathbf{E}$ can be written as

$$\mathbf{E} = \mathbf{c}\mathbf{1}^T + \mathbf{1}\mathbf{c}^T - 2\mathbf{Y}\mathbf{Y}^T \tag{2}$$

where

$$\mathbf{c} = [c_1 c_2 \dots c_n]^T \text{ with } c_i = \|q_i\|^2$$

$$\mathbf{1} \text{ is } n \text{ by 1 vector with all entries equal to 1}$$

Consider the centering matrix

$$\mathbf{J} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T \tag{3}$$

It simply translate a set of points such that the origin to be the geometrical center of points. Now, watch the following equation carefully,

$$\begin{aligned} -\frac{1}{2}\mathbf{JEJ} &= -\frac{1}{2}\mathbf{J}(\mathbf{c}\mathbf{1}^T + \mathbf{1}\mathbf{c}^T - 2\mathbf{Y}\mathbf{Y}^T)\mathbf{J} \\ &= \mathbf{J}\mathbf{Y}\mathbf{Y}^T\mathbf{J} \\ &= (\mathbf{JY})(\mathbf{JY})^T \end{aligned}$$

because $\mathbf{J1} = \mathbf{1}^T\mathbf{J} = \mathbf{0}$. We can build isometric transformation from this equation.

Let's compute $-\frac{1}{2}\mathbf{JMJ}$ and its eigen decomposition

$$-\frac{1}{2}\mathbf{JMJ} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \tag{4}$$

then let

$$\mathbf{Y} = [q_1 q_2 \dots q_n]^T = \mathbf{Q}\mathbf{\Lambda}^{\frac{1}{2}}$$

In summary, we find points $\{q_1, q_2, ..., q_n\}$ with the Euclidean distance from q_i to q_j equal to the geodesic distance from p_i to p_j. Or, we can say the set of points $\{q_1, q_2, ..., q_n\}$, lying on a hypersurface in $\mathbb{R}^n$, is invariant under isometric transformation.

We can consider a special case where you can visualize the situation. If the rank of $\mathbf{Y}$ is 2, the points $\{q_1, q_2, ..., q_n\}$ will be on the surface in Euclidean plane. Under this situation, we said the original surface S is developable. You can think original surface S as a piece of paper. No matter how you twist this paper, as long as it's still isometrical transformation, you can always flatten it. The flattened surface is a isometric invariant.

Now, the only problem is that most surfaces are not developable. For example, face surface is obviously not developable. In this case, one trivial solution is to approximate the flattened surface by using two largest positive eigenvalues and their corresponding eigenvectors. Let $\hat{\mathbf{\Lambda}}$ denote the 2×2 diagonal matrix of first two largest eigenvalues and $\hat{\mathbf{Q}}$ be the $n \times 2$ matrix of their corresponding eigenvectors. Then, approximated isometric invariant $\hat{\mathbf{Y}}$ can be calculated by

$$\hat{\mathbf{Y}} = \hat{\mathbf{Q}}\hat{\mathbf{\Lambda}}^{\frac{1}{2}}$$

3 Software Implementation

In this section, I use $Matlab^{®}$ to implement the algorithm in previous section. First, I generate a parameterized surface and compute geodesic distance between pair of points on the surface. Computing geodesic distance on a surface is a classic problem in differential geometry. One can find fast and accurate numerical algorithm in [5]. Since computing geodesic distance is not main topic in this report, I will skip this part even though it is a critical step in real applications. I simply generate part of a sphere. The geodesic distance on a sphere is very easy to compute. This surface is shown in Figure 1.

The surface flattening procedure is very easy to program in $Matlab^{®}$. Given the squared geodesic distances matrix $\mathbf{M}$, the steps described in previous section is given by

$$\begin{aligned}
\mathbf{J} &= \text{eye}(n) - \text{ones}(n,1)\text{ones}(n,1)'/n; \\
\mathbf{B} &= -0.5 * \mathbf{JMJ}; \\
\mathbf{Q}, \mathbf{L} &= \text{eigs}(\mathbf{B}, 2, \text{'LM'}); \\
\text{newx} &= \text{sqrt}(\mathbf{L}(1,1)) * \mathbf{Q}(:,1); \\
\text{newy} &= \text{sqrt}(\mathbf{L}(2,2)) * \mathbf{Q}(:,2);
\end{aligned}$$

The flattened surface is shown in figure 2. The result is visually reasonable. The points near north pole are closer and the points near equator are far away.

4 Discussion and Conclusion

In [3], the authors also mention two other ways of finding isometric invariants. One is called *Least Squares MDS*. It is based on solving the minimization of cost function defined by

$$C(\mathbf{Y}) = \|\mathbf{M} - \mathbf{E}\|$$

where matrix norm is Frobenius norm. The details can be found in [1]. The other one is a computationally efficient algorithm proposed by Faloutsos and Lin [4].

In Israel, Bronstein brothers and Kimmel [2] build an $3D$ face recognition system based on isometric invariants discussed in this paper. Their experiment results demonstrate that isometric invariant is robust to facial expression.

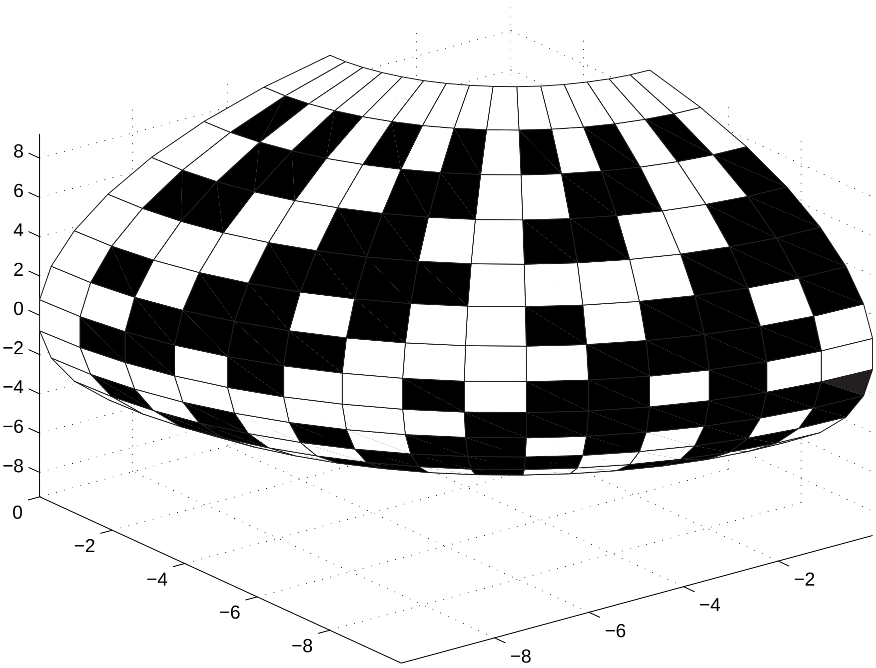

Figure 1: Part of a sphere : $\theta \in [\frac{\pi}{8}, \frac{7\pi}{8}]$ and $\phi \in [0, \frac{4\pi}{8}]$,

From my point of view, isometric invariants is really an interesting research area. It's relatively new compared with projective invariant. The related literatures I found are from the same research group. The most important of all, it shows promising potential in the problem of recognizing non-rigid objects. I strongly believe there will be more and more people working in this area.

References

[1] I. Borg and P. Groenen. *Modern Multidimensional Scaling-Theory and Applications.* Springer, 1997.

[2] A. M. Bronstein, M. M. Bronstein, A. Spira, and R. Kimmel. Face recognition from facial surface metric. In Anonymous, editor, *Computer Vision - ECCV 2004. 8th European Conference on Computer Vision.*, volume 2, pages 225–37. Springer-Verlag, / 2004.

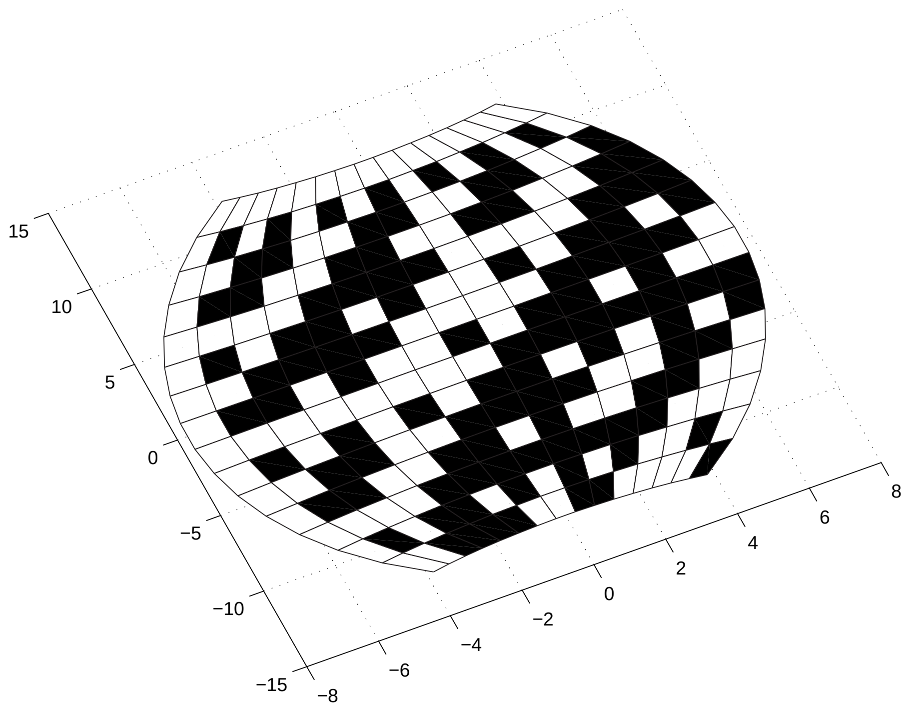

Figure 2: The surface after applying flattening procedure.

[3] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, OCT 2003.

[4] Christos Faloutsos and King-Ip Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, page 163, 1995.

[5] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences of the United States of America*, 95(15):8431–8435, JUL 21 1998.

[6] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):198–207, 04/ 2002.