Deriving surface node displacements from a modified discrete Laplacian

Brook Gotschall

Department of Mathematics and Statistics

California State University, Long Beach

INTRODUCTION

Optical distortion is of significant concern in the aircraft windshield production industry. Figure 1 illustrates optical distortion with a surface generated using CAD software.

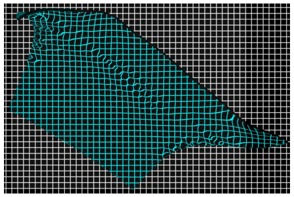

Figure 1: Virtual grid-board image of a distorted surface. A virtual grid-board is a computer generated image to simulate the inspection grid used to evaluate distortion.

Eliminating optical distortion requires a good defining surface that is continuous, tangentially smooth, and has smooth curvature. Continuity and smooth tangents are easily achieved. Smooth curvature, however, is often a challenge when preserving the original shape of the surface is necessary.

METHODS

The m (rows) x n (columns) nodal points from a surface, as plotted in Figure 2, appear to be smooth, however, the surface plot of the discrete Laplacian in Figure 3 shows a significant amount of distortion.

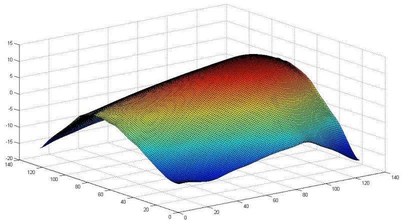

Figure 2: MATLAB surface plot of 123X123 nodal points taken from the illustration surface and reoriented to be approximately parallel to the x-y plane.

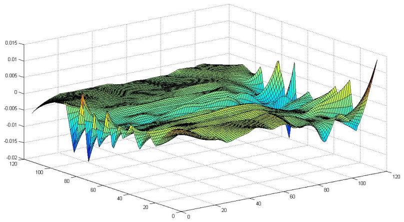

Figure 3: MATLAB surface plot of the discrete Laplacian. Pronounced distortions can be noted especially in the central area of the surface.

Consider the smoothed discrete Laplacian in Figure 4.

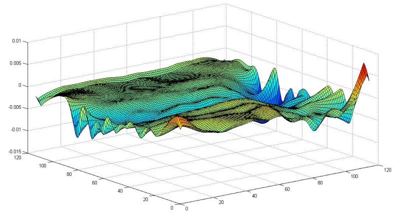

Figure 4: MATLAB surface plot of smoothed discrete Laplacian. The data was smoothed using a simple 3x7 averaging filter:

$$\frac{1}{15}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Variations are significantly reduced.

Now to determine what displacement values, when added to the original data, will produce this smoothed discrete Laplacian:

Since the discrete Laplacian, with the i^{th}, j^{th} surface node z-coordinate given by u_{ij}, is defined by:

$$l_{ij} = \tfrac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) - u_{ij}$$

consider this modification, where a_{ij} are the desired displacement values:

$$\begin{aligned} sl_{ij} &= \tfrac{1}{4}((u_{i+1,j} + a_{i+1,j}) + (u_{i-1,j} + a_{i-1,j}) + (u_{i,j+1} + a_{i,j+1}) + (u_{i,j-1} + a_{i,j-1})) - (u_{ij} + a_{ij}) \\ &= \tfrac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) - u_{ij} + \tfrac{1}{4}(a_{i+1,j} + a_{i-1,j} + a_{i,j+1} + a_{i,j-1}) - a_{ij} \\ &= l_{ij} + \tfrac{1}{4}(a_{i+1,j} + a_{i-1,j} + a_{i,j+1} + a_{i,j-1}) - a_{ij} \end{aligned}$$

subtract l_{ij} from each side and multiply by 4:

$$4(sl_{ij} - l_{ij}) = a_{i+1,j} + a_{i-1,j} + a_{i,j+1} + a_{i,j-1} - 4a_{ij}$$

rewrite in matrix notation:

$$4(sL - L) = MaA$$

where sL, L, and A are columnated vectors and Ma is a square matrix.

Next is to solve for A in:

$$MaA = 4(sL - L) = B$$

Since Ma is sparse, it can be factored using the MATLAB command:

$$[C,R] = qr(Ma,B,0)$$

where $C=Q'*B$ is produced without computing Q. R is a Cholesky factorization, and is produced as sparse.[1] The solution to A is then given by:

$$A = R \backslash C$$

A is reshaped to (m-2)x(n-2), zeros are padded on all sides, then it is added to the original surface node z-coordinates. The improved surface is thus achieved.

RESULTS

Some improvement was achieved in the resulting surface is shown in Figure 5. The largest displacement, in absolute value, was .048".

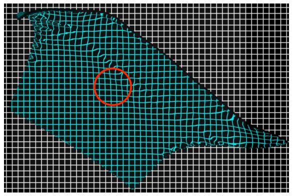

Figure 5: Virtual grid-board image of an improved surface. Considerable distortion is still present. However, some improvement is detectable, especially in the area indicated.

SUMMARY

Some improvement to distortion was realized. However, a significant amount of displacement from the original surface was required.

CONCLUSIONS

Challenges:

- Size of matrix Ma:
  - Over 200x10^6 elements → Exceeds 800 mb of memory, at single precision.
  - Very sparse → Less than 1 mb of memory, at double precision.

$$Ma = \begin{bmatrix} Ma_{11} & I & [0] & \cdots & [0] \\ I & \ddots & \ddots & \ddots & \vdots \\ [0] & \ddots & Ma_{ii} & \ddots & [0] \\ \vdots & \ddots & \ddots & \ddots & I \\ [0] & \cdots & [0] & I & Ma_{n-2,n-2} \end{bmatrix} \quad \text{where,} \quad Ma_{ii} = \begin{bmatrix} -4 & 1 & 0 & \cdots & 0 \\ 1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & -4 \end{bmatrix}$$

- Solution to A:
  - Gaussian Elimination → Memory overflow errors.
  - SVD → MATLAB does not allow sparse matrices.
  - QR on sparse matrices → Less than 1 minute for result.

Moving forward:

- Additional improvement to distortion while reducing displacement.
- Further investigation into the use of filters, peak clipping, or a combination of the two.

ACKNOWLEDGEMENTS

[1] MATLAB documentation: "*[C,R]=qr(A,B,0)*" in the function reference for "qr".

The guidance and teachings of Dr. Chang, Dr. Chaderjian, Dr. Lee, Dr. Xu, and the Department of Mathematics at CSULB were invaluable throughout this project.