Convex Hull Approach for Minimum Zone Evaluation of Straightness and Flatness

Gyula Hermann

Budapest Tech
Bécsi út 96/B, H-1034 Budapest, Hungary
hermann.gyula@nik.bmf.hu

Abstract: Algorithms to determine the minimum zone straightness and flatness have been successfully established by a number of researchers. This paper after presenting algorithms based on techniques borrowed from computational geometry focuses on the robustness and simplicity of the mathematical techniques. As the most resource consuming part of these algorithms is the determination of the convex hull, both in two and three dimensions, emphasize is given to them. Subsequently the complexity and implementation issues are discussed. The paper outlines an application using the described algorithms.

Keywords: Minimum zone methode, straightness, flatness, coordinate measuring, evaluation software

1 Introduction

The geometric features of manufactured components are generally simple shapes like straight lines, planes, circles, spheres, cylinders and cones. These features can be checked by using specific gauges, but currently these tasks are more and more performed by coordinate measuring machines, because they are powerfull and flexible devices. Obtaining a set of datapoint representing accuratly the workpiece to be checked proved to be difficult. The measurement accuracy depends on the sampling schema, systematic errors, measurement uncertainty and the robustness and efficiency of the verification algoritms.

Form tolerances attached to single features control how close to ideal form the feature must be. Form tolerances most commonly attached to planar surfaces are straightness and flatness. Accurate algorithms determining the deviation from the ideal feature will reduce the possibility to reject good workpieces.

2 Literature Overview

Direct search methods for determining the minimum zone were first mentioned by Murthy and Abdin, who describe a simplex search method starting at the least-square solution and terminating after a number of iteration. Anthony et al. discuss the theoretical basis behind the exchange algorithms, which are essentially geometrically based non-linear programming algorithms, and develop a feasible-descent data exchange algorithm. They also describe a subset algorithm that starts with the solution to a subset of datapoints and iteratively adds points until all points are inside the solution boundary. Carr and Ferreira proposed a linear program model using the small displacement screw matrix to linearize the non-linear constrains, and they used this technique to solve the minimum zone problem. Huang, Fan and Wu proposed the smallest parallelepiped enclosure method for spatial straightness error evaluation from the composition of two orthogonal planer straightness errors (vertical and horizontal). This solution cannot guarantee that the evaluated error has the same value in every direction. Kaiser and Krishnan developed a 'brute force' approach, which finds the minimum zone using a simple iterative algorithm by rotating the initial guess of the envelope lines. The process stops as the condition mentioned in the definition of straightness is reached. This algorithm was extended to determine flatness of a plane face. Samuel and Shunmugam used the computational geometric technique to solve the minimum zone and function-oriented solutions for the straightness and flatness problem. However, the flatness obtained from the 2-2 case was not taken into account in their algorithm. The paper by Lee deals extensively with the 2-2 model and gives a solution based on the convex hull. Suen and Chang propose the application of neural networks to determine the minimum zone straightness and flatness using interval regression analysis. Zhang et al. state in their paper that finding the spatial straightness error based on the minimum zone condition can not be found using the simplified linearized model. They propose evaluation using the original nonlinear model.

Algorithms to determine the minimum zone straightness and flatness have been successfully established by a number of researchers. Because the number of datapoints involved in the evaluation is not extremely large (usually less than 50) this paper focuses on the simplicity of mathematical techniques but at the same time keeps an eye on the efficiency as well.

The paper begins with preliminaries presenting the definition of straightness and flatness as it is given in the standard [ANSI/ASME Y14.5M] followed by the description of the proposed algorithms based on techniques borrowed from computational geometry. Subsequently the performance of these algorithms and implementation issues are discussed. Finally an application using the described algorithms is presented.

3 Preliminaries

In this paper the evaluation of straightness and flatness is considered in accordance with the ANSI/ASME Y14.5 and the ISO/R1101 standards and other related papers listed in the references.

Straightness is a condition where an element of a surface is a straight line. A straightness tolerance specifies that each line element must lie in a zone bounded by two parallel lines separated by the specified tolerance and that are in the cutting plane defining the line element (Fig. 1).

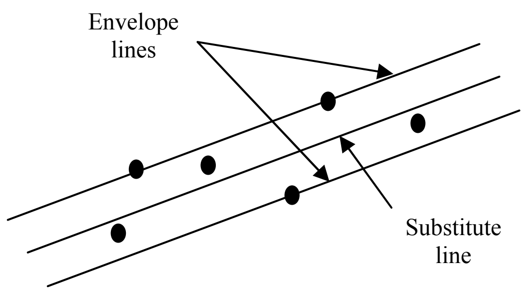

Figure 1
Definition of the straightness with given datapoints

Flatness is the condition of a surface having all elements in one plane. A flatness tolerance specifies a zone defined by two parallel planes separated by the specified tolerance within which the surface must lie (Fig. 2).

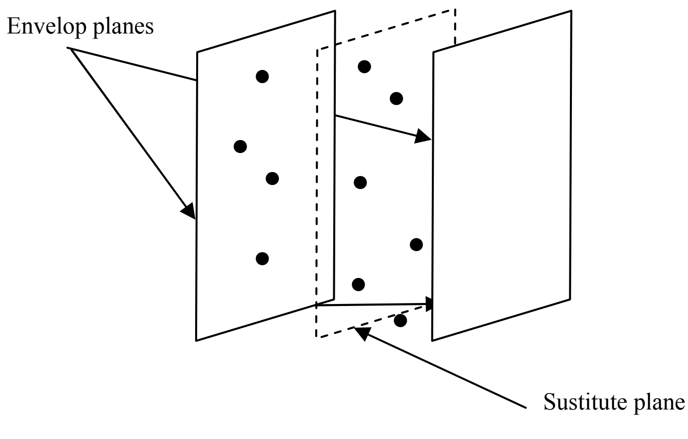

Figure 2
Definition of the flatness with given datapoints

The L_p-norm solution minimizes the following function:

$$L_p = \sum_{i=1}^{n} |d_i|^p$$

Where n is the number of data points, d_i is the distance between the i-th datapoint and the ideal feature and $1 \leq p \leq \infty$. The L_2 norm the so-called least square-fit, widely used in coordinate measuring machine software, minimizes the sum of the squares of the residual errors d_i ; $\mathrm{Min}\sum|d_i|^2$. The Chebyshev best-fit (L_∞-norm solution) minimizes the maximum deviation from the ideal feature and results in a $\mathrm{Min}\{\mathrm{Max}|d_i|\}$ objective function. The Chebyshev best-fit problem can be transformed into a non-linear constrained optimization problem. The solution of it requires iterative process whose convergence depends to a large extend on the initial guess.

In case of straightness and flatness result from computational geometry based on the convex hull can be used to obtain the exact solution in a number of definitive steps.

4 The Proposed Algorithm

Because the problem we are considering here are essentially discrete from its nature we are looking for a robust solution in the context of combinatorial geometry.

The following conditions must be met by the minimum zone straightness solution:

- The minimum zone envelop lines must contact at least three datapoints.
- These datapoints must be in a upper-line/lower-line/upper-line or a lower-line/upper-line/lower-line sequence.

The algorithm determining the substitute line and the straightness error consists of two main steps:

1 Computing the convex hull of the datapoints, resulting in a list of points in boundary traversal order.

2 Determination of the substitute line for the convex pointset and the width of the convex hull in one step.

For the determination of the convex hull of a planar pointset a number of algorithms have been invented in the recent past. Most widely known are: the gift wrapping, the quickhull invented by Preparata and Shamos, the Graham's scan, the divide and conquer and the incremental algorithms. While the Graham's scan provides the solution in optimal time, O(nlogn) it has no obvious extension to three dimension. Therefore other solution having similar execution property was taken into consideration. It is the modified incremental algorithm:

```
Let H2 ← convex_hull{p0,p1,p2}
        for k = 3 to n-1 do
        Hk ←convex_hull{Hk-1 U pk}
```

The first convex hull is the triangle p_0,p_1,p_2 . Let Q=H_{k-1} and p=p_k. The computation of H_k falls into two cases: if p_kЄQ (even on the boundary) it can be discarded. p_k is not in Q than the convex hull should be modified. Therefore we need only to find the two tangent lines from p to Q. p_i is a tangency point if for two subsequent edges the LeftOn test (on which side of the line the point is) gives different results. This is shown in the subsequent figure.

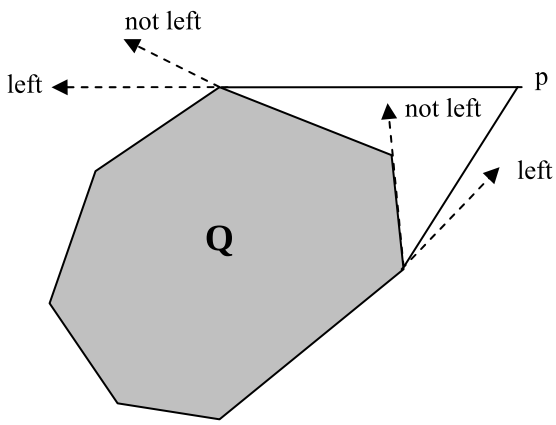

Figure 3
Detemination of the tangency lines from p to Q

The algorithm runs in O(n^2) time. However by sorting the points by their x coordinates this can be decreased to O(nlogn).

Given the points of the convex hull the next step is the computation of the substitute line and the width of the convex hull. The boundary lines of the minimum zone are determined by three points: two of them defining one envelope line is colinear with one of the edges of the convex hull, the third one is a point with the maximum perpendicular distance to this line. The algorithm next does the job in O(n^2) time, where n is the number of extreme points on the convex hull:

```
d=0
for i = 1 to n
        {c=0, j=i+2
        while j<2n dist(e_i,p_j)>d  {c= dist(e_i,p_j), j=j+1}
        if d<c {d=c, l=i, m=j-l}
        }
```

The substituting line is defined by the points $(p_l+p_m)/2$ and $(p_{l+1}+p_m)/2$. The envelop lines are given respectively by p_l, p_{l+1} and $p_l,p_m-p_l+p_{l+1}$.

Analogically the minimum zone flatness solution must satisfy the following conditions:

- The minimum zone lines must contact at least four datapoints.
- If the four contact points form a 2-2 model (two points contacting each plane), the line connecting the two upper plane points will intersect the line connecting the two lower plane points when the points are projected onto the same plane.
- If the four contact points form a 3-1 model (three points contacting one plane and one point contacting the other plane), the single point must be inside the triangle formed by the three points on the other plane when the points are projected onto the same plane.

From the above properties it be came obvious that the datapoints defining the minimum zones are vertex points on the convex hull of the original dataset. The straightness error is equal to the distance between the one data point and the line spanned by the two other points while the flatness error is given by the distance between the two lines (in case of the 2-2 model) or by the distance between the point and the plane determined by the three points (3-3 model).

The overall structure of the algorithm is the same as in two dimensions. In each step the computation of the new convex hull is again the main problem. If p is inside Q then it is discarded. If not the cone with apex p_i and tangent to Q is computed. It is clear that those faces are to be discarded which are visible from p_i.

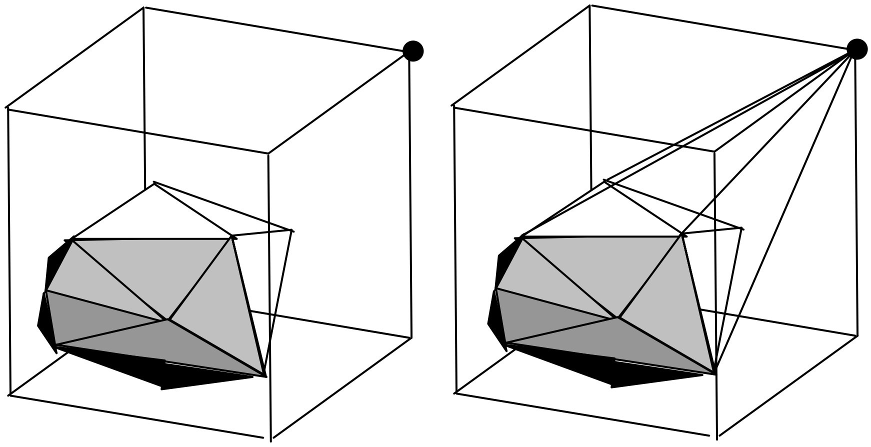

Figure 4
Convex hull before and after adding a point

The incremental convex hull algorithm is summerized as follows:

```
Initialize H3 to tetrahedron (p0,p1,p2,p3)
for i = 4 to n-1 do
for each face f of Hi-1 do
        Mark f if visible
if no faces are visible
        then dicard pi (it is inside Hi-1)
        else
             for each border edge e of Hi-1 do
                  Construct cone face determined by e and pi
             for each visible face f do
                  Delete f
             Update Hi
```

Since the loops marking the visible faces and constructing the cones are inbedded inside a loop that iterates n times the complexity of the algorithm is $O(n^2)$.

Given the points of the convex hull the next step is the computation of the substitute planes and the width of the convex hull. The boundary planes of the minimum zone are determined by the 3-1 model: the minimum distance between the face defined by three points and one point with the maximum perpendicular distance to this line. The algorithm next does the job in $O(m^2)$ time, where m is the number of edges on the convex hull:

```
d=0
for i = 1 to n
        Compute the equation of the face fi from it's cornerpoints
        for j = 1 to n
        Determine the distance of the point pj from the face
        if  dist(pj,fi) > d then d = dist(pj,fi)
for i =1 to m
    for j = 1 to m
         if dist (ei, ej) > d then d = dist (ei, ej) > d
```

5 Implementation Issues

The computation of straightness is straightforward, it does not require any particular datastructure. In the three dimensional case the topology both of the polytope and the convex hulls should be explicitly represented in order to speed up the search processes. For its effectiveness the socalled quad-edge data structure invented by Guibas and Stolfi for the representation of any subdivision 2-manifolds was selected.

In the data structure each edge record is part of four circular lists. Two list are the lists for the endpoints and two lists reperesent the adjacent faces. Members of the list are linked by pointers. By introducing convention for the loops representing the faces boundary the direction of the surface normal vector are defined implcitly. Using this information the hidden faces can be easily be determinated and eliminated.

6 The Host Application

As the problem was raised in connection with the calibration on straight edges and on a coordinate measuring machine it was an obvious choise to implement the above described algorithm within the calibration software. It is written in Visual Basic and receives data from an Excel table. The output is the final calibration certificate while the data values and the evaluation result are archived.

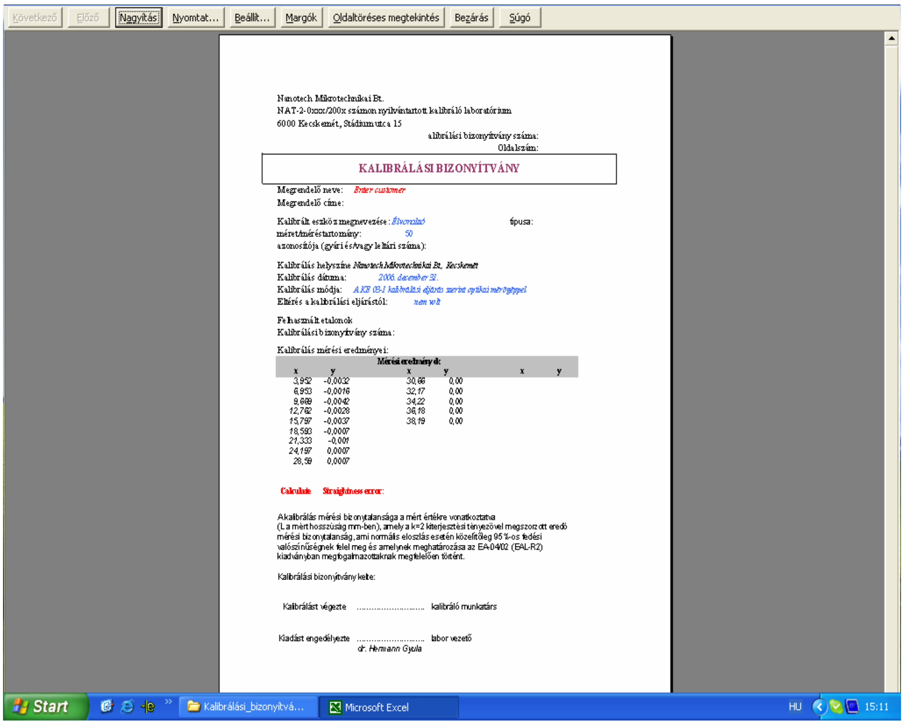

Figure 5
The user interface of the calibration software

Conclusions

The paper describes a simple robust algorithm for determining the straightness and flatness of surface features. The algorithm was implemented as part of a software package supporting the calibration of gauges and it is currently in usage. It's price/performance ratio is superior to other software currently available on the market.

References

[1] G. T. Anthony, H. M. Anthony, B. Bittner, B. P. Buttler, M. G. Cox, R. Driescher, R. Elligsen, A. B. Forbes, H. Gross, s.a. Hannaby, P. M. Harris, J. Kok. „Reference Software for Finding Chabysev Best-Fit Geometric Elements" Precision Engineering 19 (1996) 28-36

[2] ANSI/ASME Y14.5M Dimensioning and Tolerancing. New York, American Society of Mechanical Engineers, 1994

[3] K. Carr, P. Ferreira „Verification of Form Tolerances Part I: Basic Issues, Flatness, and Straightness" Precision Engineering 17 (1995) 144-156

[4] K. Carr, P. Ferreira „Verification of Form Tolerances Part II:Cylindricity and Srtraightness of a Median Line” Precision Engineering 17 (1995) 131-1

[5] L. J. Gubia, J. Stolfi „Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams”, ACM Transactions on Graphics 4, 74-123

[6] J. Huang „Evaluation of Angular Error between Two Lines” Precision Engineering 27 (2003) 304-310

[7] Huang, K. C. Fan, J. H. Wu „A New Minimum Zone Methode for Evaluating Straightness Errors” Precision Engineering 15 (1993) 158-165

[8] M. J. Kaiser, K. K. Krishnan „Geometry of the Minimum Zone Flatness Functional: Planar and Spatial Case” Precision Engineering 22 (1998) 174-183

[9] M. K. Lee „A New Convex-Hull-based Approach to Evaluating Flatness Tolerance” Computer Aided design 29 (1997) 861-868

[10] T. S. R. Murty, S. Z. Abdin „Minimum Zone Evaluation of Surfaces” Intenational Journal of Machine Tool Design and Research, 20 (1980) 123-136

[11] J. O'Rourke „Computational Geometry in C” Cambridge University Press 1998

[12] F. P. Preparata, M. Shamos „Computational Geometry” New York Springer Verlag, 1985

[13] G. L. Samuel, M. S. Shunmugam „Evaluation of Straightness and Flatness Error Using Computational Geometris Techniques” Computer Aided design 31 (199) 829-843

[14] D. S. Suen, C. N. Chang „Application of Neural Network Interval Regression Methode for Minimum Zone Straightness and Flatness” Precision Engineering 20 (1997) 196-207

[15] Qing Zhang, K. C. Fan, Zhu Li „Evaluation Methode for Spatial Straightness Errors Based on Minimum Zone Condition” Precision Engineering 23 (1999) 264-272