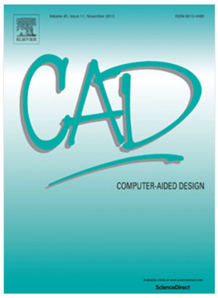
Accepted Manuscript



An efficient algorithm for minimum zone flatness based on the computation of the largest inscribed ball in a symmetric polyhedron

Yu Zheng

PII: S0010-4485(17)30033-7
DOI: http://dx.doi.org/10.1016/j.cad.2017.03.004
Reference: JCAD 2496

To appear in: *Computer-Aided Design*

Received date : 15 June 2016
Accepted date : 10 March 2017

Please cite this article as: Zheng Y. An efficient algorithm for minimum zone flatness based on the computation of the largest inscribed ball in a symmetric polyhedron. *Computer-Aided Design* (2017), http://dx.doi.org/10.1016/j.cad.2017.03.004

An Efficient Algorithm for Minimum Zone Flatness based on the Computation of the Largest Inscribed Ball in a Symmetric Polyhedron

Yu Zheng[1]

Department of Electrical and Computer Engineering, University of Michigan-Dearborn

Abstract

This paper presents an algorithm for the minimum zone flatness tolerance of a finite point set, which is defined to be the minimum Euclidean distance between two parallel planes that sandwich the point set. The algorithm is based on the observation that the flatness tolerance is equal to the radius of the largest inscribed ball in the convex hull of the Minkowski difference of the point set and itself, which is a symmetric polyhedron with respect to the origin. Then, an iterative procedure is developed to adaptively grow another symmetric polyhedron inside the convex hull of the Minkowski difference such that the radius of its inscribed ball monotonically increases and converges to the flatness tolerance. The algorithm is guaranteed to compute the globally minimum solution within finite iterations. Moreover, there is no need to compute the Minkowski difference or the convex hull of the point set, so the proposed algorithm is very fast and takes only several milliseconds for hundreds of thousands of points on a normal computer, such as a desktop computer with an Intel Xeon 3.70GHz CPU and 16GB RAM used in this work.

Keywords: geometric form, flatness, largest inscribed ball, minimum zone, width, computational geometry

1. Introduction

Flatness is one of the most fundamental geometric forms that are used in manufacturing to measure the quality of a machined surface. According to ASME Y14.5, Dimensioning and Tolerancing [1], "flatness is the condition of a surface or derived median plane having all elements in one plane and a flatness tolerance specifies a tolerance zone defined by two parallel planes within which the surface or derived median plane must lie." The actual value of flatness for a surface is the smallest flatness tolerance to which the surface will conform; that is, it is the minimum distance between two parallel planes that sandwich the surface. To practically evaluate the flatness of a surface, a set of points are often sampled from the surface and the value of flatness is calculated based on the sampled data points. While the sampling strategy and the number of points to be sampled can affect the computed value of flatness [2, 3, 4, 5], this paper focuses on how to accurately and efficiently compute the flatness tolerance for large data point sets.

A number of algorithms have been proposed to compute the value of flatness for data point sets. They can be classified into several categories:

1. Some algorithms calculate a flatness tolerance based on the maximum and minimum deviations from a plane best fitting the data points. The fitting plane can be determined by the traditional least-square approximation [6, 7, 8, 9] or an iterative reweighted least squares algorithm [10]. However, flatness tolerances obtained in these ways are often bigger than the minimum value.
2. The computing of the minimum flatness tolerance can be written as a nonlinear optimization problem that searches for the pair of parallel planes with the smallest space to bound the data points between them. General-purpose optimization algorithms were used or modified to solve this problem [11, 12, 13, 14, 15]. Because of the nonlinear and nonconvex nature of the problem, however, it is difficult to guarantee a solution to be globally optimal. These algorithms usually need be run for multiple times with different initial conditions to obtain a so-

Email address: yuzheng001@gmail.com (Yu Zheng)