Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems Raleigh, NC July 7-10, 1992

A CAD-Based System for Automated Inspection of Machined Parts *

Bharath R. Modayur †and Linda G. Shapiro ‡

†Department of Electrical Engineering, FT-10
‡Department of Computer Science & Engineering, FR-35
University of Washington
Seattle WA 98195
U.S.A.

Abstract

Although many special purpose inspection systems have been developed, general purpose systems utilizing CAD models of the parts are still in the research stage. While it is easy to define ad hoc algorithms for inspection, it is much more difficult to justify the algorithms with solid theory. In this paper we describe a CAD-model-based machine vision system for dimensional inspection of machined parts, with emphasis on the theory behind the system. The original contributions of our work are: 1) the use of precise definitions of geometric tolerances suitable for use in image processing, 2) the development of measurement algorithms corresponding directly to these definitions, 3) the derivation of the uncertainties in the measurement tasks, and 4) the use of this uncertainty information in the decision-making process. Our experimental results have verified the uncertainty derivations statistically, proved that the error probabilities obtained by propagating uncertainties are lower than those obtainable without uncertainty propagation, and demonstrated that the inspection system responds in a predictable manner when applied to deformed objects.

1 Introduction

CAD-based vision is the automatic production of vision procedures for a specific task, given CAD models of the objects involved in the task and knowledge of the environment in which the task is to be performed. This approach is much more cost-effective than the older approach of designing specialized techniques for each new part and has been advocated by a number of people (e.g. [10], [8], [2]) for use in machine vision systems. Most of the CAD-based vision systems have been for part recognition and pose estimation, not for inspection, but a few inspection systems ([7], [15]) have been built.

There are two major problems involved in automating the inspection process. The first problem is technique. The techniques to be used in the inspection system must satisfy a set of standards. But the standards for conventional inspection tasks were not designed for machine vision and are, in most cases, unsuitable. For example, consider the task of determining if a planar surface is flat enough. There are several methods used in industry to do this. One method is to dye (with ink) the irregular surface and measure the area of the dyed portion imprinted on a known planar surface. A second method is to use a mechanical stylus. The stylus is run all over the irregular surface and the highest and lowest points are measured. This gives an idea of the irregularity of the surface. These procedures are not immediately adaptable to a machine vision system, because the criteria for success are not standardized. The second problem is interpretation of results. The error of the final measurement depends on the error at each step of the processing. Most machine vision work has not seriously considered the propagation of error.

While it is easy to define *ad hoc* algorithms for inspection, it is much more difficult to justify the algorithms with solid theory. Our goal in this work is to develop suitable definitions for a visual inspection system, to develop the appropriate theory for analyzing the errror of our measurements, and to design and implement an experimental CAD-based vision system for automated inspection of machined parts that employs the definitions and theory we have developed. In this

*This research was supported by the National Science Foundation under grants DMC-8714809 and IRI-9023977 and by the Washington Technology Center.

0-7803-0737-2/92$03.00 1992©IEEE

paper, we describe the experimental system we are building as a whole, with emphasis on the theory behind the measurements and the experimental results so far. In Section 2, we briefly describe the related literature. Section 3 gives the appropriate definitions for tolerances and for the measurement tasks that use these tolerances. Section 4 derives the formulas for uncertainty propagation for use in decision making. Section 5 describes the design of our experimental system. Section 6 discusses the image processing required for the straightness-of-edge task, and Section 7 describes the experiments and results.

2 Related Literature

Though there is an abundance of literature in the area of automatic visual inspection for specific domains (e.g: solder joints, printed circuit boards, light bulb filaments, etc.), literature on inspection systems (specifically CAD-based) for general machine parts is hard to come by.

Requicha [14] was the first to lay down a formal theory of tolerancing. We draw on the ideas in this seminal work and the guidelines prescribed in the ANSI standards [1] to set up formal definitions of the various tolerances. Park *et al* [12] discussed issues in developing an automated inspection system with emphasis on achieving an integrated CAD-Vision model, not on the tolerance theory or the measurement tasks. Other inspection systems have been described by [15] and [5]. Although a number of coordinate measurement machine (CMM) vendors claim to have solved the inspection problem, they work by fitting surfaces and curves to image data and inspecting the parameters of the fit. This, however, is not what the industrial standards prescribe.

3 Measurement Definitions

We provide several different types of measurements in our system: 1) the straightness of an edge, 2) the angle between two edges, 3) the position of a corner, 4) the position of circular hole or slot, 5) the size of a circular hole or slot, and 6) the form of a circular hole or slot.

In order to implement these tasks, we must first define them in a way that naturally leads to computational vision algorithms. Since machine vision relies heavily on extracting features from the image data, we will employ the methodology of *geometric tolerancing*, which is widely used in industry, is well-defined in terms of ANSI standards, and makes explicit use of features.

The major problem with feature-based measurement is that if the features are imperfect, the measurement are not well-defined. For instance, the distance between two jagged edges is not well-defined. The angle between two broken line segments is not well-defined. The coordinates of a vertex with respect to a coordinate system formed by two imperfect lines are not well-defined. Geometric tolerancing solves this problem with the introduction of the *simulated datum feature*, a perfectly-formed geometric entity that is associated with an imperfectly shaped, manufactured object. Thus, our inspection system associates with imperfect, real features, corresponding perfectly-formed features so that measurements can be made.
We consider five geometric features in our inspection tasks. They are: straight lines, circles with the material side external to the circle, circles with the material side internal to the circle, rectangles with the material side external to the rectangle, and rectangles with the material side internal to the rectangle. These correspond to the two-dimensional counterparts of a planar feature, a cylindrical slot, a cylindrical part, a rectangular slot, and a rectangular part, respectively. In order to implement our measurement tasks, datum features have to be associated with these five geometric features.
In the remainder of this section, we define the simulated datum features for the important object features that are required for our inspection tasks and then specify the measurements that are required for a few of the tasks.

3.1 Datum Features - Definitions and Construction

Planar Features Planar features in 3D are the planar surfaces of the object. Because we do not have 3D sensors, the only planar features we inspect are edges. The simulated datum feature for a straight edge is a straight line positioned so as to minimize the integral sum of distances between points on the edge to the line. A digital edge is just a sequence of N points. This gives us the problem of how to construct a supporting line L to the N points such that the sum of perpendicular distances from the points to the line is a minimum. We have proven that the supporting line must pass through an edge of the convex hull of the N points [11]. So our procedure is to compute the sum of the perpendicular distances to the given points for all the edges of the convex hull of the points and choose the edge yielding the lowest sum of distances. This edge is the required straight line datum. Figure 1 illustrates the construction of the simulated datum

feature.

Cylindrical Parts and Slots The 2D profile of a cylindrical part is a circle with the material side internal to the circle. The associated simulated datum is the smallest circumscribing circle. The 2D counterpart of a cylindrical slot is a circle with the material side external to the circle. The associated simulated datum is the largest inscribing circle. Constructing the smallest circumscribing circle and the largest inscribing circle for a set of 2D data points is a well-defined computational geometry problem [13].

3.2 Tolerance Definitions and Measurements

Our experimental work has so far been with straight edges. We have worked with three different tasks: measuring the straightness of an edge, measuring the angle between two edges, and determining corner point position. We define these straight edge measurement tasks as follows.

Straightness of an Edge
An edge with a straightness tolerance T_s conforms to the specification if it can be enclosed completely by two parallel lines at a separation less than T_s. We check for the straightness in the following way. The required simulated datum feature (in this case, a straight line) is first constructed. This datum line is then translated until all the edge points are on or between the simulated datum and this translated version. If the distance between these two parallel lines is less than T_s, the edge conforms to the straightness specification. This is illustrated in Figure 1.

Angle Between Two Edges
Let the ideal angle between two edges be θ_{id} and let the angular tolerance be specified by T_a. The two edges satisfy the angular tolerance, if the angle between the two associated simulated datum features is θ_{obs}, satisfying $\theta_{id} - T_a/2 \leq \theta_{obs} \leq \theta_{id} + T_a/2$. We construct the simulated datum features associated with the two edges as outlined before. Then we measure the angle between the two resultant straight lines. The measured angle has to lie within the interval $[\theta_{id} - T_a/2, \theta_{id} + T_a/2]$ for the edges to conform to the angularity tolerance.

4 Uncertainty Propagation in Measurements

A real image is seldom absolutely noise free. Noise in the image leads to uncertainties in the attributes of the entities output by image processing algorithms. Uncertainty in the lower-level image entities leads to uncertainty in the tolerance measurement tasks. To illustrate this, we will take the reader through one task, namely measurement of straightness of an edge. The derivations for variances of other measurements are similar in principle.

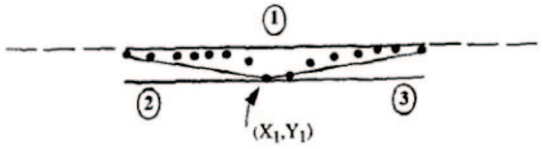

Figure 1: Line segments (1), (2), and (3) form the convex hull of the set of edge points. Edge (1), since it minimizes the sum of the perpendicular distances to the edge points, is the required datum line. Point (X_1, Y_1) is the farthest from the datum line in terms of perpendicular distance. The perpendicular distance of (X_1, Y_1) from datum line (1) should conform to the straightness tolerance.

4.1 Noise Model

Let the true edge pixel position be denoted by (x_i, y_i). Let the observed edge pixel position be denoted by $(\hat{x_i}, \hat{y_i})$. Our model for the noisy, observed edge pixel is, $\hat{x_i} = x_i + \epsilon_i$, $\hat{y_i} = y_i + \xi_i$. where, ϵ_i and ξ_i are samples from independent distributions that are even functions [16], with mean zero and variances σ_{ϵ_i} and σ_{ξ_i}, respectively.

4.2 Datum Line Uncertainty

The simulated datum line for straight edges is the nearest-supporting line that passes through an edge L_1 of the convex hull of the edge pixels. Let us denote the end points of this hull edge by (x_1, y_1) and (x_2, y_2). We can write the line equation as $L_1 : \alpha x + \beta y + \gamma = 0$. Since, we only have noisy observations $(\hat{x_1}, \hat{y_1})$ and $(\hat{x_2}, \hat{y_2})$, the observed line parameters would be expressed as functions of $(\hat{x_2}, \hat{y_2})$.

Let us estimate the behavior of one of the line parameters α as a result of noise on the edge pixels. We will represent α by a Taylor series expansion around the true edge points (x_1, y_1) and (x_2, y_2). We can then truncate the Taylor series as an approximation and in-

clude only the linear terms. As a result

$$\begin{aligned}\hat{\alpha} &= \alpha + \frac{(y_2-y_1)^2}{d^3}[(\hat{y}_1-y_1)-(\hat{y}_2-y_2)] \\ &+\frac{(x_2-x_1)(y_2-y_1)}{d^3}[(\hat{x}_1-x_1)-(\hat{x}_2-x_2)]+ \\ &\frac{1}{d}[(\hat{y}_2-y_2)-(\hat{y}_1-y_1)]\end{aligned}$$

Squaring the above equation and taking expectations on both sides results in $V[\hat{\alpha}] =$

$$\begin{aligned}&[V[\hat{y}_1]+V[\hat{y}_2]]\left(\frac{(y_2-y_1)^4}{d^6}-\frac{(y_2-y_1)^2}{d^4}+\frac{1}{d^2}\right) \\ &+[V[\hat{x}_1]+V[\hat{x}_2]]\left(\frac{(x_2-x_1)^2(y_2-y_1)^2}{d^6}\right) \\ &+2\,[Cov[\hat{x}_1,\hat{y}_1]+Cov[\hat{x}_2,\hat{y}_2]] \\ &\left(\frac{(y_2-y_1)^3(x_2-x_1)}{d^6}-\frac{(x_2-x_1)(y_2-y_1)}{d^4}\right) \quad (1)\end{aligned}$$

In a similar way, we can estimate the variances of the parameters $\hat{\beta}$ and $\hat{\gamma}$ in terms of the coordinates (x_1, y_1) and (x_2, y_2). To measure the straightness of this edge, L_1, we have to find another line parallel to this datum line and such that all the edge pixels are on or between the two lines. Assume that this parallel line, L_2, passes through (x_3, y_3). The equation for L_2 is $L_2 : \hat{\alpha}x + \hat{\beta}y - (\hat{\alpha}x_3 + \hat{\beta}y_3) = 0$. The distance between the two parallel lines that should conform to the straightness tolerance is $\hat{d} = \hat{\alpha}x_3 + \hat{\beta}y_3 + \hat{\gamma}$. Since α and β are themselves functions of θ which is the angle that the line makes with the X-axis, we can rewrite d as a function of θ and γ. Proceeding with the truncated Taylor series expansion as before,

$$\begin{aligned}V[\hat{d}] &= V[\hat{\theta}][x_3\cos\theta + y_3\sin\theta]^2 + V[\hat{\gamma}] + \\ &\quad 2[x_3\cos\theta + y_3\sin\theta]Cov[\hat{\theta},\hat{\gamma}] \quad (2)\end{aligned}$$

Equation 2 expresses the uncertainty of the straightness measurement, given the observed coordinates of the two edge pixels that support the datum line ($(\hat{x}_1, \hat{y}_1)$ and $(\hat{x}_2, \hat{y}_2)$) and the third edge pixel $(\hat{x}_3, \hat{y}_3)$ that is farthest away from this datum line, and the variances of the observed edge pixel positions. Thus, the uncertainties in the edge pixel positions have been propagated all the way up to the measurement task.

5 System Design

We are building a system called ICIS (Interactive CAD-based Inspection System) to test our concepts and to illustrate how we think automated inspection should be carried out. ICIS consists of two parts: an interactive front end and a set of computer programs to handle the actual inspection tasks. The interactive front end allows the user of the system to select the inspection tasks required for the part to be inspected. The user first selects the part from a database of CAD models. A wire-frame view of the object is then displayed on the screen. The user employs a virtual sphere device (see [4]) to interactively rotate the object so that he/she can view the various features to be inspected. A combination of menus and mouse clicks on the wire-frame image is used to select the actual tasks to be performed. Figure 2 gives an example of a typical screen displayed to the user by ICIS.

The interactive system works with a 3D vision model of the object, derived from the original CAD model. The vision model is a hierarchical, relational model of the part in terms of its surfaces, its edges, and their interrelationships. This vision model has been used in our automatic pose estimation work [3]. For use in the inspection system, we have augmented the model to include inspection specifications and tolerances. For example, each edge has an associated straightness tolerance. The system derives this information from its interaction with the user.

The second part of the system is the set of inspection procedures. We are initially focusing on measurement tasks. The measurement procedures call on image processing procedures to extract the required information from the image(s) of the object and then make the decisions as to the acceptability of the part.

6 Image Processing for Inspection Tasks

The transformation matrix from the object recognition stage yields information, with a degree of uncertainty, about approximately where to expect features of the object to appear on the image. The image processing operators, instead of operating on the entire image, operate only in specific regions where we expect features (such as edges) to lie. We call these specific regions "search windows". Given the uncertain transformation matrix and an estimate of the noise in the image, an accurate estimate of the size and position of these search windows can be obtained. We have developed an adaptive Kalman-filter-based algorithm [11] to fuse these two uncertainties to get an estimate of the size and position of the search windows. All the measurement tasks in this work require just two basic low-level entities. They are (1) edge pixels and (2) corner pixels. The sequence of image processing operations employed is geared toward extracting these two

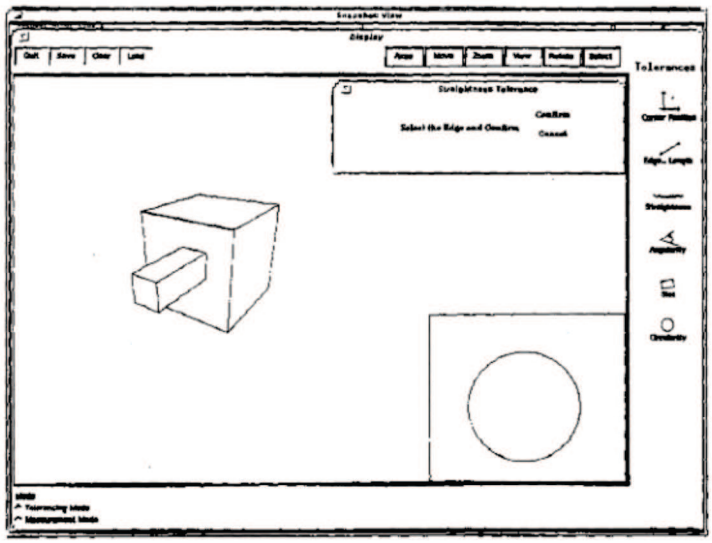

Figure 2: A typical screen displayed by ICIS. An edge has been selected from the hidden-line-removed object. Straightness tolerance has been selected. On the bottom right is the "virtual sphere" with the current viewpoint highlighted.

primitive features from selected regions in the image that are defined by the search windows.

The first step in our image processing sequence is edge detection. We employ the Haralick edge operator [6], which uses the zero crossings of the second directional derivative to classify edge pixels. The edge detector output is contaminated with stray noise specks, spurious edge pixels and small segments that cannot be grouped with other segments on the basis of adjacency or orientation. Thus, a necessary step after edge detection is symbolic grouping of edge pixels to form a higher-level entity. As a preliminary step we perform a connected shrink operation. The output of the shrink operator is a symbolic image of arc segments that are one pixel wide. After very small segments are removed, the remaining segments are grouped according to adjacency and orientation. The adjacency condition can be relaxed to group edge segments that are broken because of noise. The set of edge segments output by the grouping process is directly used by the measurement procedures. Corner pixels are obtained as a by-product of the grouping process described above. Points of high curvature on the arc segments are classified as corners.

7 Experiments and Results

Our experiments consisted of three phases. In the first phase, the accuracy of the variance formulas derived for the various measurement tasks were checked. In particular, we were interested in the conditions under which the first order Taylor series approximation would fail. We employed a statistical testing procedure for this purpose, taking the straightness measurement as an example task. In the second phase, the performance of the measurement algorithms was determined with and without error propagation. As in the first phase, the straightness measurement task was utilized as a representative task. The goal of the third phase was to test the inspection system with machined objects at varying degrees of deformations. To do this for the straightness measurement task would require a method of progressively deforming the straightness of an edge. Since this type of deformation is not well-defined, we chose the angle-measurement task, for which it was possible to introduce systematic deformations to the machined object. By repeated machining of the object, we progressively varied the angle of one edge with respect to another.

Phase 1

As a first step in our experiments, we checked for the accuracy of the variance formulas derived for the various measurement tasks. We employed a statistical testing procedure for this purpose, taking the straightness measurement as an example task.

In order to test whether $\sigma^2_{\hat{d}}$, the variance of the

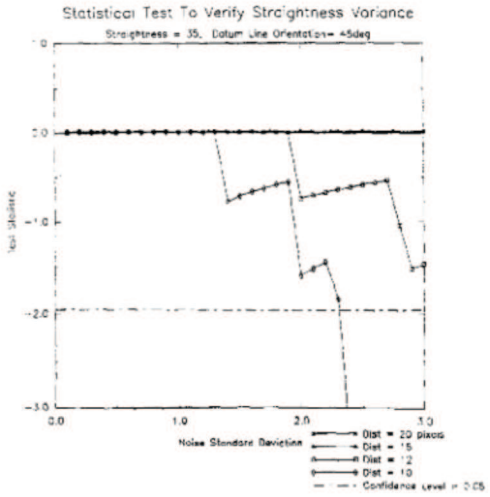

Figure 3: Results of the statistical test to verify straightness variance. The datum line used was oriented at 45°. The straightness of the synthetic edge used in this test was 35 pixels. The distance between the two edge points that support the datum line was varied from 20 pixels down to 10 pixels. The test fails when the absolute value of the test statistic becomes higher than 1.96 (significance level $\alpha = 0.05$). The test comes close to a breakdown when the distance is 12 pixels (4 times σ) and breaks down when the distance is 10 pixels.

straightness measure $\hat{d}$ is equal to $V[\hat{d}]$, the analytic formula derived, the null and alternate hypotheses were formulated to be

$$H_0 : \sigma^2_{\hat{d}} = V[\hat{d}]$$
$$H_a : \sigma^2_{\hat{d}} \neq V[\hat{d}],$$

and the test statistic to be

$$Test = \frac{\sigma^2_{\hat{d}} - V[\hat{d}]}{\sigma^2_{\sigma^2_{\hat{d}}}}.$$

Since the distribution of $\sigma^2_{\hat{d}}$ is not known, we approximate the mean and variance of $\sigma^2_{\hat{d}}$ by the experimental mean straightness measure variance and the mean variance of the straightness measure variance, respectively. The statistical test was carried out with the significance level α=0.05, corresponding to a value of ± 1.96 for a normalized Gaussian random variable. Thus the null hypothesis would be accepted if the test statistic were between ± 1.96.

Figure 3 shows the result of the statistical test for a line oriented at 45^0, with perpendicular distance to the farthest edge point being 35 pixels. The statistical test value fell below the significance level when the distance between the two points fell below 12. Thus, for the linearization approximations made in the variance derivations to hold, the two points that support the datum line must be separated by at least four standard deviations (of the noise).

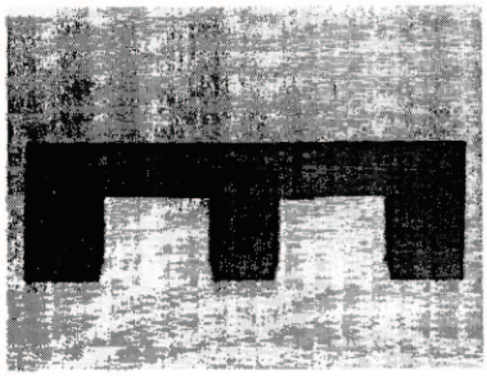

Figure 4: The machined object.

Phase 2

The second stage of our experiments focused on determining the performance of the measurement algorithms with and without error propagation. We again tested the straightness measurement algorithm. Our experimental object is shown in Figure 4. The object was modeled using PADL-2 and machined on a CAMM-3 modeling machine. We selected two edges in this image: a "good" edge (in terms of straightness) and a "bad" edge. We determined the datum lines for the good edge and for the bad edge.

An evaluation process influenced by [9] was then employed (see [11] for a detailed description of the experimental methodology). This procedure starts with the real data obtained from the image and performs 1000 different perturbations at each of several noise levels to thoroughly test the procedures. The false alarm and misdetection probabilities we obtained for various noise levels have been plotted in Figures 5 and 6. Our results show that the decision making procedure that takes error propagation into account yields

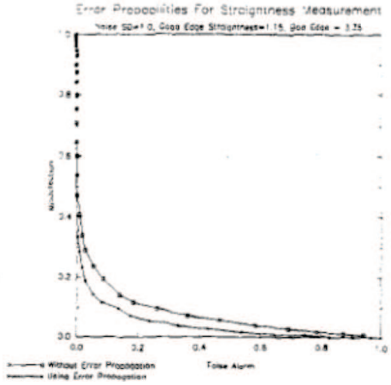

Figure 5: Error probabilities with noise $\sigma = 1.0$.

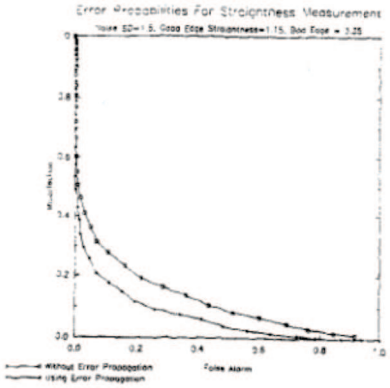

Figure 6: Error probabilities with noise $\sigma = 1.5$.

Figure 7: (a). The original machined object. (b) The same object after the fifth angular deformation. (c) The object after the tenth deformation.

lower error probabilities than the one that does not propagate errors.

Phase 3

In the third stage of experiments, the inspection system was tested with machined objects at varying degrees of deformation. Consider the visual inspection system to be a black box whose input is in the form of images of machined objects. Given a particular measurement task to perform (e.g measurement of angularity of a pair of edges), the system outputs a numerical measure along with the associated variance. The numerical measure and the associated variance are used in conjunction with the tolerance specifications in deciding the goodness of the feature(s). In our experiments, we were not directly interested in the actual value of the numerical measure, because 1) we do not possess accurate tactile measurement devices that could perform the same procedures manually for comparison, and 2) the actual values of the tolerance specifications are decided by the manufacturer and are arbitrary. In the light of these factors, what we were interested in observing was the behavior of the measure output by the inspection system as the machined object underwent deformation.

The object shown in Fig 7 was machined on our CAMM-3 modeling machine to the best possible accuracy of the machine. A pair of mutually perpendicular edges were selected. One of the edges was deformed systematically by additional machining such that the angle between the two edges increased. With this monotonic increase in the angle, the output measure of the system was observed. Each deformation produced an increase in angle of approximately two degrees and there were ten such deformations. The output of the system as shown in Figure 8 clearly indicates that the angularity measure output by the inspection system is also monotonic.

8 Conclusions

In this paper, we described a CAD-based machine vision system for dimensional inspection of machined parts. The system performs inspection by strictly ad-

Deformation #	Angular Measure Output By System
1	1.553
2	1.530
3	1.474
4	1.429
5	1.392
6	1.310
7	1.275
8	1.218
9	1.148
10	1.077

Figure 8: Progressive angular deformation of the test object produces monotonic change in the angular measure output by the inspection system.

hering to well-defined tolerance definitions. Hence all the measurements made by the system have a strong underlying theoretical basis. We extended our three-dimensional vision models originally developed for pose determination to include the tolerance information. We propagated uncertainties from lower-level edge pixels all the way up to the measurement tasks. We then incorporated these uncertainties in our decision making to rule a feature acceptable or not. Our experimental results verified the uncertainty derivations statistically, proved that the error probabilities obtained by propagating uncertainties are lower than those obtainable without uncertainty propagation, and demonstrated that the system responds in a predictable manner to progressively deformed objects.

This work sets up a theoretical and operational framework for a CAD-based inspection system. Additional measurement tasks can be included by setting up precise tolerance definitions adhering to the guidelines described. Uncertainty propagation for these additional tasks can be done by following the methodology outlined for the straightness of edge task.

References

[1] *American National Standards Institute(ANSI). 1973. Dimensioning and Tolerancing. ANSI Standard Y 14.5.* American Soc. of Mech. Engg., 1973.

[2] B. Bhanu, T. Henderson, and S. Thomas. 3-d model building using cagd techniques. In *CVPR*, pages 234–239, Jun 1985.

[3] O. Camps, L.G. Shapiro, and R.M. Haralick. Premio: An overview. In *IEEE Workshop on Directions in Automated CAD Based Vision*, pages 11–21, 1991.

[4] M. Chen, S.J. Mountford, and A. Sellen. A study in interactive 3-d rotation using 2-d control devices. *Computer Graphics*, 22:121–128, Aug 1988.

[5] F. Etesami and J. Uicker. Automatic dimensional inspection of machine part cross-sections using fourier analysis. *CVGIP*, 29:216–247, 1985.

[6] R. M. Haralick. Digital step edges from zero crossings of second directional derivatives. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Jan 1984.

[7] H.D.Park. Cavis- cad based automated visual inspection system. *PhD Thesis*, Dec 1988.

[8] K. Ikeuchi. Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks. *Int. J. Comp. Vision*, 1(2):145–165, 1987.

[9] T. Kanungo, M.Y. Jaisimha, R.M. Haralick, and R. Palmer. An experimental methodology for performance characterization of a line detection algorithm. *SPIE Opt., Illum. and Image sensing for Mach. Vision*, 1385, 1990.

[10] H. Lu and L.G. Shapiro. Model-based vision using relational summaries. In *SPIE Conference on Applications of Artificial Intelligence*, 1989.

[11] B. R. Modayur. Visual inspection of machined parts. *MSEE Thesis, Dept of EE, Univ. of Wash., Seattle*, 1991.

[12] H.D. Park and O.R. Mitchell. Cad-based planning and execution of inspection. *CVPR*, 1385, 1990.

[13] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer- Verlag N.Y. Inc, 1985.

[14] A.A.G. Requicha. Toward a theory of geometric tolerancing. *Intn. Jr. of Rob. Res.*, 2(4):45–60, 1983.

[15] G.A.W. West, T. Fernando, and P.M. Dew. Cad based inspection: Using a vision cell demonstrator. In *IEEE Workshop on Directions in Automated CAD Based Vision*, pages 155–164, 1991.

[16] S. Yi. Illumination control expert for machine vision: A goal driven approach. *PhD Thesis, Dept. of EE, Univ. of Wash.,Seattle*, 1990.