J Intell Manuf (2013) 24:71–86
DOI 10.1007/s10845-011-0540-6

A study on automatic on-machine inspection system for 3D modeling and measurement of cutting tools

Xi Zhang · Wai-Ming Tsang · Kazuo Yamazaki · Masahiko Mori

Received: 17 November 2010 / Accepted: 9 May 2011 / Published online: 24 May 2011

Abstract A 3D model of the maximum rotating envelope of a milling cutter with tool holder is required for Computer Aided Manufacturing (CAM) process design and machining simulation. The user may define the 3D model of the whole tool assembly in the tool library of CAM software. However, it is not convenient and reliable. Considering these problems, a new method based on single view 3D reconstruction algorithm has been proposed in previous research work, which is able to quickly reconstruct the 3D model of a cutter with tool holder while they are installed onto the spindle. As the extension of this work, this paper focuses on the recent progresses in order to improve the automation, accuracy, efficiency and reliability of tool modeling system. First, an improved flexible on-machine camera calibration procedure is proposed. The accurate motion of machine tool axis is used to calibrate the camera on machine tool instead of a physical calibration board. The whole procedure of calibration can be conducted automatically by running NC code. Therefore, the automation of vision system can be guaranteed. Second, the contour extraction module is improved by using a method of silhouette image composition. This method is applied to solve the problem of translucent and fuzzy cutter profile induced by motion blur. Third, the new algorithm for contour partitioning and classification are proposed, which is more reliable and robust. The reliability and accuracy of the vision system can be guaranteed. Finally, the vision system with an 8 mm lens and 1 mm extensions has been tested on different type of machine tool with smaller cutters. The average measurement accuracy is about 35 microns verified by comparison with a commercial tool setting system.

X. Zhang (✉)
School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, China
e-mail: xizhang@shu.edu.cn

W.-M. Tsang · K. Yamazaki
IMS-Mechatronics Laboratory, University of California, Davis, Davis, CA, USA
e-mail: waitsang@ucdavis.edu

K. Yamazaki
e-mail: kyamazaki@ucdavis.edu

M. Mori
Mori Seiki Co., Ltd., Nagoya, Japan

Keywords Cutting tools · Modeling · Measurement · Single view reconstruction

Introduction

For the Computer Aided Manufacturing (CAM) of milling process, a 3D model of the maximum rotating envelope of a milling cutter are necessary to design and generate the Numerical Controlled (NC) tool paths. In addition, before inputting the NC program to the machine, it is desirable to verify the program and detect possible collisions between machine tool components by machining simulation in a virtual manufacturing environment before the actual cutting. One of the inputs required to make the collision detection by simulation is the 3D model of the whole tool assembly including the tool cutter and holder.

However, the tool manufacturers usually do not disclose complete 3D model of a cutter except only its 2D drawing or the important dimensions. The user may build the 3D tool model manually by specifying the values of the geometric parameters of the tool template defined in the tool library of CAM software. Therefore, the 3D model of the tool holder as well as the tool length offset can also be specified manually in this way. By doing so, there are four main drawbacks. First, the users have to find shape and dimensions of the tool and holder from the catalog provided by tool manufacturer. It can be tedious and time-consuming. It is also prone to make mistakes to input them to CAM software manually. Second, some of the cutters might be custom tailored, and some of them may not be covered by the tool templates in the CAM library. The users have to design a customized template for those cutters in order to build model. It is not convenient and need more skills for the user of CAM software. Third, the values of tool geometric parameters may not always available. The user would need to measure those parameters manually. As far as tool measurement is concerned, there are tool measurement devices such as contact gages, on-line laser tool setting systems (http://www.blum-novotest.de/index.php?id=226) and off-line tool presetters (http://www.zoller-usa.com/en/products.htm). However, such measurements are point based. Some of the important measurements that can be obtained are the geometrical features of a cutter such as diameter and length. They cannot automatically obtain the 3D tool model for the whole tool assembly. Fourth, the measured tool length offset by tool setting system cannot attach with the 3D model of tool assembly directly. The user has to input it into the CAM software manually. Due to this inconvenience, the CAM software usually doesn't consider the tool length offset precisely during simulation. There may be no problem for 3 axis machining. However, even the result of simulation is correct, collision between cutter body and workpiece still may occur in complex 5 axis machining or in the machining process with micro cutters.

Considering these drawbacks, the most reliable way is to try to obtain quickly the 3D model and tool setting information from the actual cutter while it is clamped on the spindle. All the cutters can be loaded from the tool magazine one by one. Their 3D models are reconstructed by an on-machine measuring system within several seconds. The 3D models together with tool offset information are saved and imported into the tool library of the CAM software. These 3D models can be used directly in simulations by just simply selecting it in the tool list. The measurement is needed usually only one time since the obtained 3D models can be used repeatedly. If the offset of a cutter changes for some reason, the measurement can be conducted quickly and automatically once again.

The authors have proposed a new method to support this idea in previous research work (Zhang et al. 2008, 2010). The 3D model of a rotating milling cutter can be reconstructed automatically on machine tool from a single camera. The system will improve the productivity of the automated manufacturing by eliminating the needed manual labor to build the 3D model. This method is flexible for any kinds of milling tools. In addition, this system is an economical solution since the price of camera will continuously decrease along with the improvement in performance. However, there are still some problems with this system:

- Because of the limitation of camera calibration method, full potential of the camera cannot be utilized; automation of the camera calibration cannot be realized as well.
- Contour of an imaged cutter cannot be extracted correctly for large cutters because of translucent contour from the motion blur of the flutes.
- Contour partitioning cannot be reliable using curvature analysis due to noisy points on the contour.
- The generating function of the cutter is calculated point by point for the segment of straight line, which is not efficient.

Noting these problems, this paper will improve the existing system regarding automation, accuracy, efficiency and reliability of the tool modeling system. The paper is organized as follows. In the second section, the system design for automatic 3D tool model reconstruction is described; in the third section, the improved on-machine camera calibration procedure is discussed; in the fourth section, the new designed contour analysis algorithms are explained. In the fifth section, the 3D reconstruction from self-occlusion is given. In the sixth section, the experimental verification of the vision system and measurement accuracy comparison with laser based tool setting system is presented.

On-machine automatic 3D tool modeling system

Single view reconstruction method

A rotating cutter and holder is a Surface of Revolution (SOR), which is formed by rotating a planar curve, called generating function, around an axis. Considering this constraint, a vision based method is proposed (Zhang et al. 2010). Only one camera is used for 3D modeling of cutter and holder. Since an SOR can be determined by its generating function completely, the key point here is how to obtain the generating function of a cutter and holder from an image. The overall idea is shown below.

The image of a cutter is taken by a camera (Fig. 1a, b). Then the contour of the cutter is extracted from the image (Fig. 1c). According to the camera model explained in details in "Determining the cross section of a cutter", one point on the contour can be used to determine one cross section of the 3D model, which means the center and radius of this cross section can be calculated. Using all the points on the contour, the 3D model of the cutter can be reconstructed as a stack of circular cross sections as shown in Fig. 1d. Then the generating function can be obtained by the intersection curve between a plane and the stack of circular cross sections as shown in Fig. 1e. Once the generating function is extracted, it is easy to build the 3D model and extract tool offset information. The detailed algorithm for calculation one cross section is explained in next section.

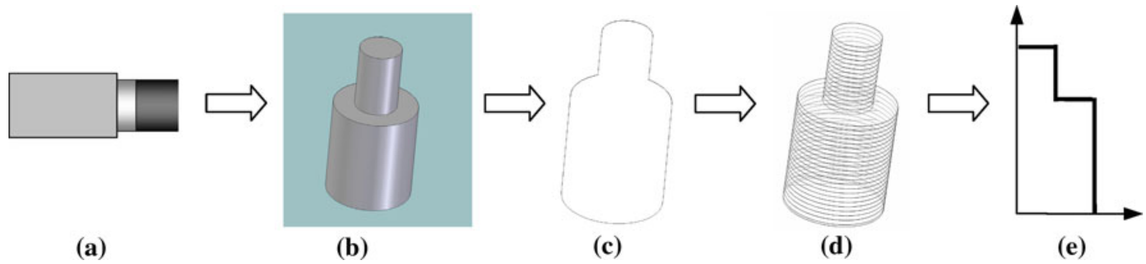


Fig. 1 Vision-based solution for 3D model reconstruction of a milling tool

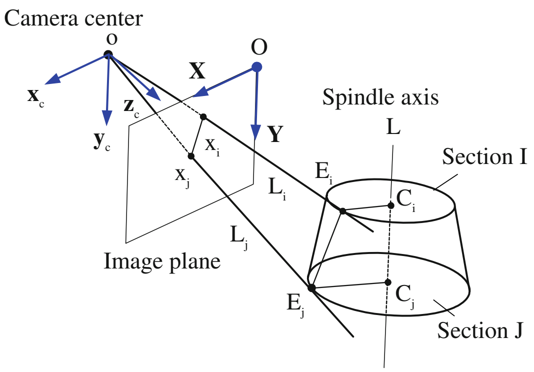


Fig. 2 3D reconstruction from contour

Determining the cross section of a cutter

Suppose there is an image of a cutter taken by a camera and the contour of the imaged cutter is known. According to the camera model in Fig. 2, a point on the contour and a known camera center, can define a light ray, which is tangent to the surface of the cutter. If the axis of the cutter and the parameters of the camera are known, the center and radius of a cross section can be calculated. The detailed algorithm is explained below.

There are two coordinate systems: the pixel coordinate system O-XY and Camera Coordinate System (CCS) $o\text{-}x_c y_c z_c$. The image point x_i and the camera center define the light ray, L_i, which is tangent to the cutting tool at point E_i. Let the intrinsic parameter of the camera be K. The direction vector of L_i in CCS is (Hartley and Zisserman 2003)

$$L_i = K^{-1} x_i = (m_i, n_i, p_i) \tag{1}$$

The perpendicular distance between line L_i and the spindle axis equals to the radius of a circle in 3D space. Suppose the direction vector of spindle axis relative to CCS is (a, b, c) and (x, y, z) is an arbitrary point on spindle axis. The radius of the circle on cross section I is,

$$r_i = D_1/D_2, \text{ where} \tag{2}$$

$$D_1 = \begin{vmatrix} x & y & z \\ a & b & c \\ m_i & n_i & p_i \end{vmatrix} \tag{3}$$

$$D_2 = \sqrt{\begin{vmatrix} a & b \\ m_i & n_i \end{vmatrix}^2 + \begin{vmatrix} b & c \\ n_i & p_i \end{vmatrix}^2 + \begin{vmatrix} c & a \\ p_i & m_i \end{vmatrix}^2} \tag{4}$$

Next, the center of this cross section will be determined. The image of a circular cross section is an ellipse. Any image point on the ellipse can be used to calculate the center of the cross section. Here the point x_i is used. As shown in Fig. 2, the point x_i can define a light ray L_i whose direction vector of light ray L_i is calculated by (1). Since the ray passes through the camera center, the coordinate of E_i measured in CCS is (mt, nt, pt). Here t is an unknown parameter needed to be calculated.

The perpendicular distance from E_i to the spindle axis is known as r_i. The problem to determine the center equals to find a point on this light ray with direction vector L_i which fulfills:

$$\begin{vmatrix} i & j & k \\ a & b & c \\ m_i t - x & n_i t - y & p_i t - z \end{vmatrix} = r_i \tag{5}$$

where, i, j, k is the direction vector of CCS. || denotes the normal of a vector. The center of the cross section I can be determined by (6).

$$c_i = \left(\begin{vmatrix} b & nt - y \\ c & pt - z \end{vmatrix}, - \begin{vmatrix} a & mt - x \\ c & pt - z \end{vmatrix}, \begin{vmatrix} a & mt - x \\ b & nt - y \end{vmatrix} \right) + E_i \tag{6}$$

Similarly the radius and center of cross section J can be determined using image point x_j. Therefore, there exists a one to one correspondence between a cross section of an SOR and a point on the contour extracted from the image.

It is necessary to point out that not all the portions of the contour are used for 3D reconstruction. As shown in Fig. 3, the contour of a SOR consists of two kinds of curves, namely the apparent contour and the imaged cross section. The apparent contour is the image of points at which the surface is smooth and the projection rays are tangent to the surface. The imaged cross section is an elliptic arc. Apparent contour can be used for building the generating function of the tool cutter, while the imaged cross section cannot be used. Therefore the contour should be partitioned and classified to find the apparent contour first.

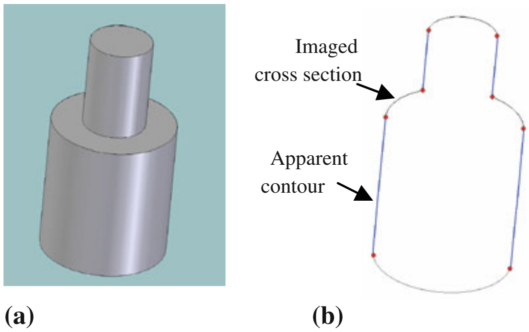


Fig. 3 The contour of an SOR. **a** Image of an SOR, **b** two kinds of contour

System design

Based on the sections "Single view reconstruction method" and "Determining the cross section of a cutter", an automatic on-machine 3D tool model reconstruction system is designed as shown in Fig. 4.

First, the parameters of the camera on the machine tool should be determined by calibration process. The position of the spindle axis relative to the camera is also identified. Second, the image of the cutter is taken and the contour of the cutter is extracted from the image. Next, the contour is divided into several segments based on curvature analysis. Then, each segment is classified as an apparent contour or

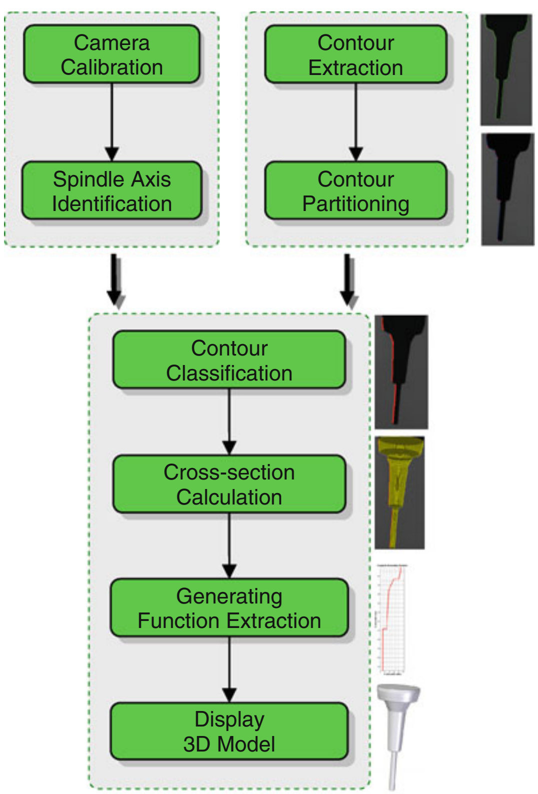


Fig. 4 On-machine automatic 3D tool model reconstruction system

an imaged cross section. The cross sections of the 3D model are calculated from apparent contours and the 3D tool model can be reconstructed as a stack of cross section along spindle axis. Then, the generating function of the 3D tool model can be extracted from the stack of cross sections. Finally, the 3D model is built by rotating the generating function along the spindle axis. The detailed algorithms for each modular will be clarified in the following sections.

On-machine camera calibration

Requirements of on-machine camera calibration

According to Eqs. (1)–(6), the intrinsic parameters of the camera as well as the position and orientation of the camera are required, which can be determined by camera calibration. The camera calibration algorithms have been studied intensively by the researchers in the societies of computer vision and photogrammetry. Usually a physical 2D or 3D calibration object is involved. The most popular method is to use the linear approximation to initialize the pinhole camera parameters, and to use the nonlinear optimization algorithm to determine the accurate pinhole camera parameters and also the distortion parameters (Salvi et al. 2002). The detailed procedures can be found in popular computer vision textbooks (Hartley and Zisserman 2003; Faugeras 1997). There are also open source calibration tools available for camera calibration (Bradski and Kaehler 2008; Bouguet 2010). From algorithm point of view, these classical methods can be used directly for the application on machine tool. However, direct implementation of classical camera calibration procedure in the machine tool environment does not work well for this specific application. For example, in the previous work (Zhang et al. 2010; Tian et al. 2010), as show in Fig. 5, a planar checkerboard is placed on the work table based on the Zhang's algorithm (Zhang 2000). There are some problems with classical camera calibration procedure regarding the application on machine tool.

First, it is desirable to reduce the human assistance and complete the entire camera calibration procedure automatically by running NC code for on-machine application. In this way, the vision system can be integrated into the manufacturing process seamlessly. However the classical camera calibration procedure is not able to automate the process. In Zhang et al. (2010), Tian et al. (2010), the user has to manually change the position and orientation of calibration board several times during camera calibration. It is inconvenient and time consuming. And the manual operation will influence the repeatability of calibration results.

Second, the classical camera calibration procedure does not consider the alignment between the calibration coordinate system with machine tool coordinate system. Usually

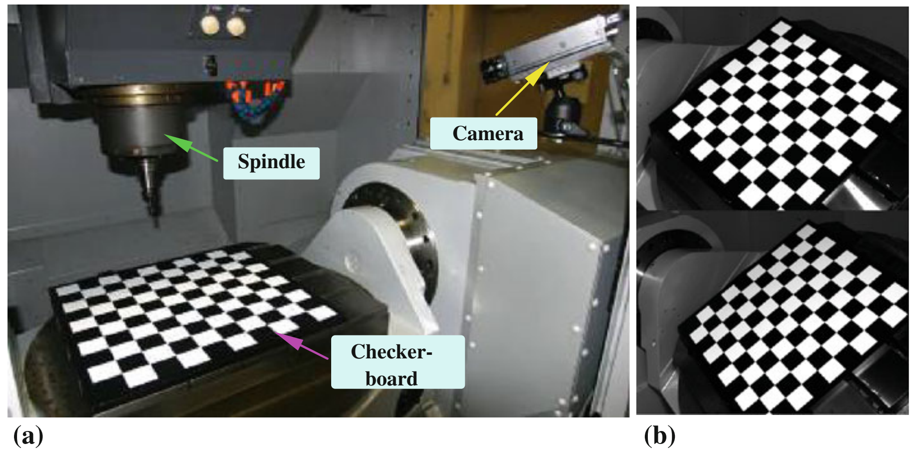


Fig. 5 Camera calibration on machine tool with planar calibration board. **a** Setup on machine tool, **b** images of checkerboard

the 3D reconstruction task in the field of computer vision is conducted relative to the camera coordinate system. Only the intrinsic parameters are concerned in the camera calibration. The position and orientation of the camera relative to the calibration target does not matter. Therefore, the calibration target can be placed in front of the camera arbitrarily. However, for on-machine applications, such as object tracking for collision detections, 3D reconstruction results should be represented relative to machine tool coordinate system. Thus, the calibration coordinate has to precisely align with the machine tool coordinate system. In Zhang et al. (2010), Tian et al. (2010), the first position of the checkerboard has to be aligned with the edge of the workable table carefully in order to establish the relation between calibration coordinate system and machine tool coordinate system. During the calibration, more efforts are made to align the calibration board precisely, which is time-consuming. However, the alignment accuracy cannot be guaranteed by this method.

Third, a fixed size calibration object is used in classical camera calibration procedure, which is not flexible for on-machine application. The method proposed in this paper is general and can be implemented on any CNC milling machine tool. For its implementation on different machine tools, the size of cutter may be different and the camera with different field of view will be used accordingly. However, following the classical camera calibration procedure as in Zhang et al. (2010), Tian et al. (2010), a special size of calibration object has to be fabricated for each application on different machine tool. It is cost consuming and not flexible.

Finally, the correlation between 2D image feature points with 3D point before calculating the camera parameters in the classical camera calibration procedure is not a trivial matter. The algorithm for feature detection and correlation is complex, which may not stable in the machine tool environment with poor illumination. In Zhang et al. (2010), Tian et al. (2010), erroneous correlation sometimes occurs due to missing detection of corners or the wrong sorting of detected corners. This will influence the stability of the calibration results.

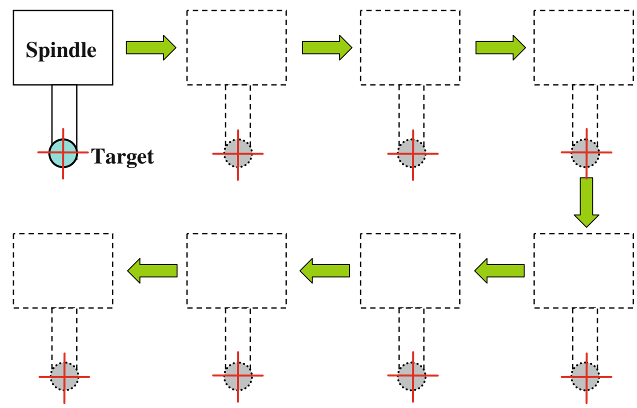


Fig. 6 Motion of machine tool and virtual 3D calibration object

In order to fix all four of these problems with camera calibration for on-machine application, an improved on-machine camera calibration procedure is presented.

Proposed camera calibration procedure

Instead of using a physical calibration object, the precise motions of CNC machine tool are used to form a virtual calibration target. The proposed camera calibration procedure is described below.

As shown in Fig. 6, install a target with marker onto the spindle of a machine tool. The center of the target can be set as a point on a virtual 3D calibration object. The absolute coordinate system is attached on the first position of the spindle. By moving the spindle to the different positions inside a cubical volume space, a virtual 3D calibration object can be established using all the center of targets. Figure 7 illustrates the image of a virtual 3D calibration object formed by moving the spindle to 125 different positions.

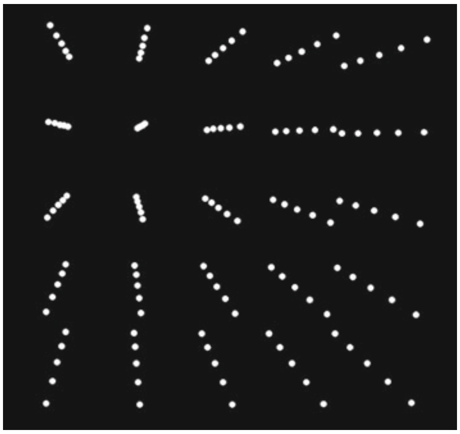


Fig. 7 Image of virtual 3D calibration object

This method has several advantages in setup convenience, flexibility, accuracy, reliability and guaranteed coordinate alignment. (1) There is only one target, which is easy to setup and can be mounted automatically onto the spindle like a cutter using the automatic tool change of the machine tool. The whole calibration process can be fully automated by running NC code. (2) By changing the motion range of machine tool, it is flexible to cover a wide range of field of view with the same camera calibration artifact. (3) The virtual calibration object is more accurate than physical calibration object. The accuracy of the calibration object is mainly determined by the motion accuracy of machine tool, which can achieve 1 micron easily. And there is less deformation comparing with physical calibration object. (4) By using one feature point, no feature matching algorithm is required to match multiple image feature points to the corresponding world coordinate points, which eliminate the complexity of designing a robust point matching algorithm. Therefore, the reliability of camera calibration is guaranteed. (5) Moreover, the calibration coordinate system is established by the motion of the machine tool. The camera calibration coordinate system and the machine tool would be aligned automatically. By placing the target feature point coincides with the spindle axis, the spindle axis can be identified automatically. Thus, the only relationship between the two coordinates is the offset.

For the calculation of camera parameters, a classical camera calibration algorithm (Hartley and Zisserman 2003) is used. First, the camera parameters are estimated using least square method. However, only the projection matrix can be obtained this way. Afterwards, all of the basic camera parameters, as well as the lens distortion coefficients can be obtained by using non-linear optimization method.

Calibration target design and feature detection

Based on the considerations above, a spindle mounted calibration target is designed as shown in Fig. 8a. This target can be mounted automatically onto the spindle using the automatic tool change of the machine tool. In order to make the system more reliable, the feature used for the camera calibration must be uniquely defined such that no other objects inside the machining environment have the same signature. The unique feature point used is the double concentric circle, which is easy to detect in the machining environment. A single circle cannot be used because there might be multiple objects on the machine with circular features, such as coolant nozzles and screws in the machine tool environment. The centers of the ellipses are used as the feature point. The feature point is designed to be aligned with the spindle axis so only the z-offset is required to relate the calibration coordinate with the machine tool coordinate. Figure 8b shows a target clamped on the spindle.

After the target is designed, the feature recognition algorithm would also need to be implemented to extract the image location of the target. The algorithm is implemented as follow:

```
1.  Extract the contours on the image
    using canny operator.
2.  Remove all of the short contours,
    the length of the contours less than
    a threshold such as 50 pixels.
3.  Remove all of the contours with high
    aspect ratios.
4.  Fit an ellipse through the remaining
    contours.
5.  Check the ellipses for which the
    distance between two ellipse centers
    is less than threshold.
6.  The minimal distance between
    the two ellipse centers passes the
    check is the center.
```

Contour extraction and analysis

Rotation induced blur and contour extraction

Extracting a complete contour of a rotating cutter is very important for 3D reconstruction. A still milling cutter (Fig. 9a) is not a real SOR due to cutting edges and flutes. When the cutter is rotating, an SOR can be formed by its maximal envelope body (Fig. 9b). Unlike a solid SOR, the contour corresponding to the rotating cutting edge is not distinct—the edge may be translucent or blurry. It is very difficult to determine the contour even with human assistance. This is called rotation induced blur, which is inevitable no matter how fast the spindle speed is, especially when the spiral flute is large.

Rotation induced blur results in the indistinctive contour corresponding to cutting edge portion, which prevents accurate extraction of the complete contour of the cutter.

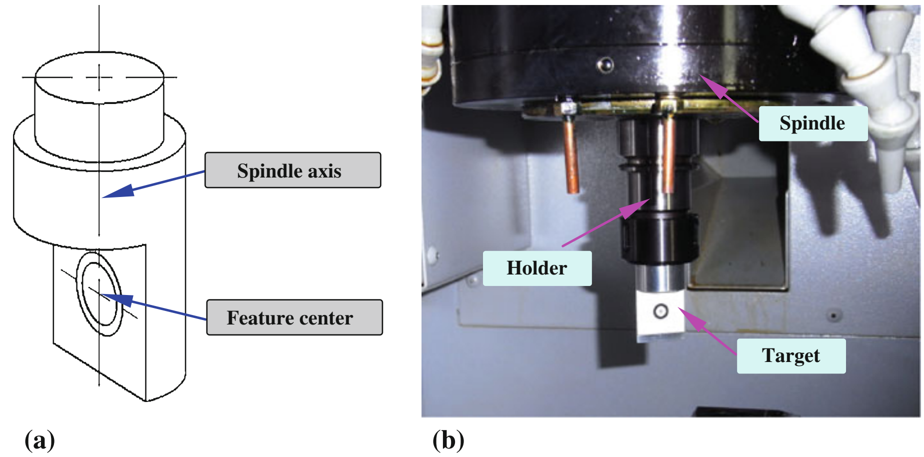


Fig. 8 **a** Camera calibration target, **b** target clamped on the spindle

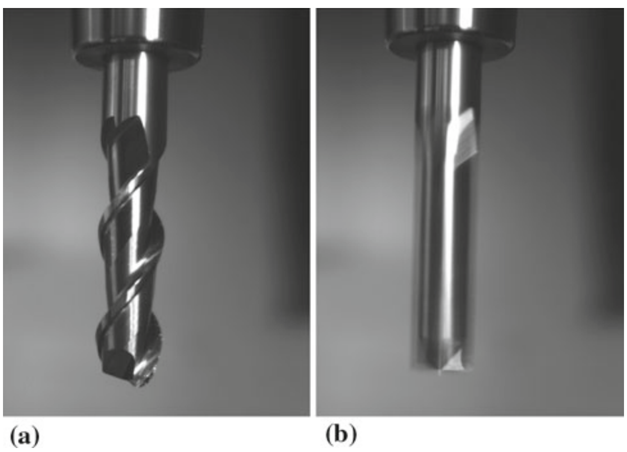


Fig. 9 Rotating induced blur. **a** Still cutter, **b** rotating cutter

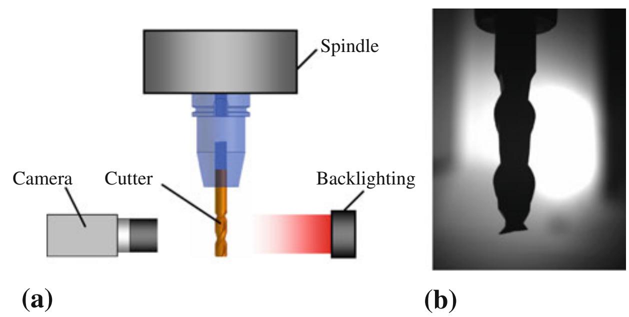


Fig. 10 The effect of backlighting. **a** Setup of lighting, **b** image of a rotating cutter

The applications of single view reconstruction in computer vision filed usually only consider solid SOR. Rotation induced blur is the challenge and special issue in the application of cutting tools.

It is very difficult to solve this problem only by image processing algorithm since it is not easy to distinguish actual edges. In order to extract the complete contour reliably, it is necessary to improve the lighting condition. Backlighting can provide sharp silhouettes of objects for edge detection, even for translucent and transparent objects. Therefore, the effect of backlighting for rotation induced blur is tested. The layout of lighting and camera is shown in Fig. 10a. With fast shutter control, the image of the cutter is captured while the cutter is rotating. The image of a rotating cutter is almost the same as the image of a still cutter. The edge is very sharp and easy to extract as shown in Fig. 10b.

Based on this configuration, the idea to obtain the complete contour is proposed. First, a sequence of images of the rotating cutter is captured. Then the images are overlapped to find the maximum region occupied by a rotating cutter.

The number of images captured at different rotation position should be enough to cover 180 degrees. So it depends on how many degrees the cutter rotates when one frame is captured. The rotating Degree per Frame (DPF) is determined by the Frame Speed (FS, unit: fps) of the camera, the Spindle Speed (SS, unit: rpm), as well as the number of flutes. The DFP can be calculated by

$$\mathrm{DPF} = \frac{(\mathrm{SS}/60)^{*}360}{\mathrm{FS}} = \frac{6^{*}\mathrm{SS}}{\mathrm{FS}} \tag{7}$$

The total different position is.

$$\mathrm{N} = \begin{cases} 180/\mathrm{DPF} & \text{if DPF} \leq 90 \\ 180/\left|\mathrm{DPF} - 180\right| & \text{if DPF} > 90 \end{cases} \tag{8}$$

The total number, n, of images to get the required different position also depends on the number of flutes. Due to the symmetrical design of the tool cutter, cutter with more flutes can get the same amount of different positions as cutters with less flutes by using less input images. The following equation shows the relationship between n and the number of flutes.

$$\mathrm{n} = \mathrm{N}\frac{2}{\text{number of flutes}} \tag{9}$$

Here is an example to illustrate the application of Eqs. 7–9, for a two flutes cutter, suppose the frame speed FS is 30 fps and spindle speed SS is 925 rpm. Therefore, about 36 images are needed to get the complete contour. This is verified by the

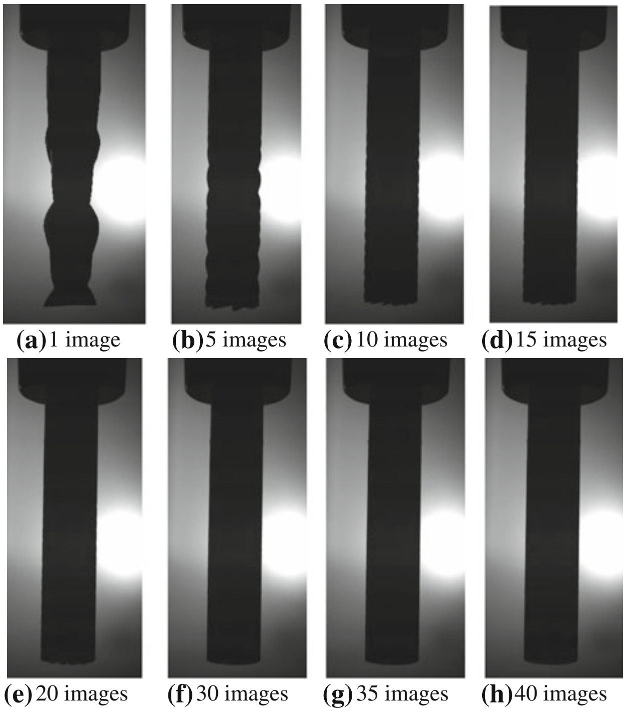


Fig. 11 Overlapping images

experiment. As shown in Fig. 11, it can be found that there is no complete contour formed until 35 images are used. As the number of images is increased, there is no obvious change.

Once the complete contour is obtained, the edge detector such as canny detector can be used. In order to remove the noise edge, the length and shape criteria are used.

```
(1)  Remove the edge which is too short

(2)  Remove the edge which meets the
     intensity criterion:
     Ratio > Threshold value

  Here, Ratio = N / Area
  N is the total number of points on the
  edge
  Area is the area of the minimal
  circumscribed rectangle of the edge
```

The results of a straight end mill and a ball end mill are shown in Fig. 12.

Contour partitioning

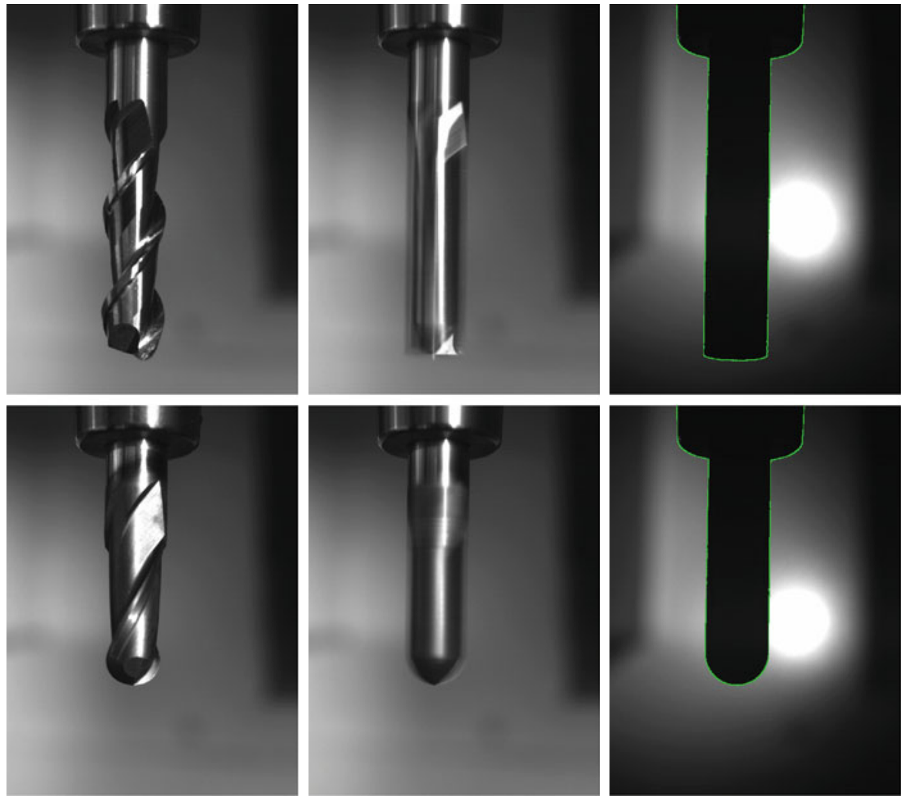
The purpose of contour partitioning is to divide the contour into several segments by Critical Point (CP), where exists a sharp change in the curvature. In order to determine the CP, curvature analysis of the extracted contour is necessary. There are two ways for curvature analysis: directly and indirectly. The direct way is to calculate the curvature of each discrete point or the curve fitted by discrete points. The other is not to calculate the discrete curvature necessarily. Instead, it uses some measure which can reflect the change of curvature. In previous research work (Zhang et al. 2010), the CP can be determined based on the direct curvature analysis. The contours are fitted into a cubic polynomial spline and find the points with local maximal curvature. However, there are too much noise points on the contour, some of which may be regarded as a point with local maximal curvature. In order to remove these false points, complex criteria are designed. Therefore, the results of determining of CP are not stable, which results in erroneous results of 3D reconstruction. In order to solve this problem, the indirect method based on Digital Straight Segment (DSS) recognition algorithm (Klette 2004) has been proposed in this paper. The concept of Forward Minimal Bounding Rectangle (FMBR) is used.

FMBR

The meaning of FMBR is explained here. For each point on the contour, use it together with some of its forward sequential points to determine the minimal-area bounding rectangle. If the width of the rectangle is less than a threshold (i.e 2 pixels), more forward points can be added to calculate bounding rectangle again until the width of rectangle exceeds a threshold. Then the final minimal-area bounding rectangle is called as a FMBR. The length of FMBR is defined by the length of rectangle. Its angle is the angle between the line pointing from FMBR center to its beginning points and the x axis. Figure 13a below shows the FMBR of two points Pi and Pj. It can be noticed that the intersection of two FMBRs may correspond to a critical point. Therefore, the FMBR can be used to partition the contour into several segments which is robust to noise point. Some of the intersection points between two FMBRs are critical points. Others are not. As shown in Fig. 13b, point A, B, D, E, G are critical points, while point C and F are not. The change of two FMBR in angle and length can be used as criterion to find CP from all the candidate points determined by FMBR.

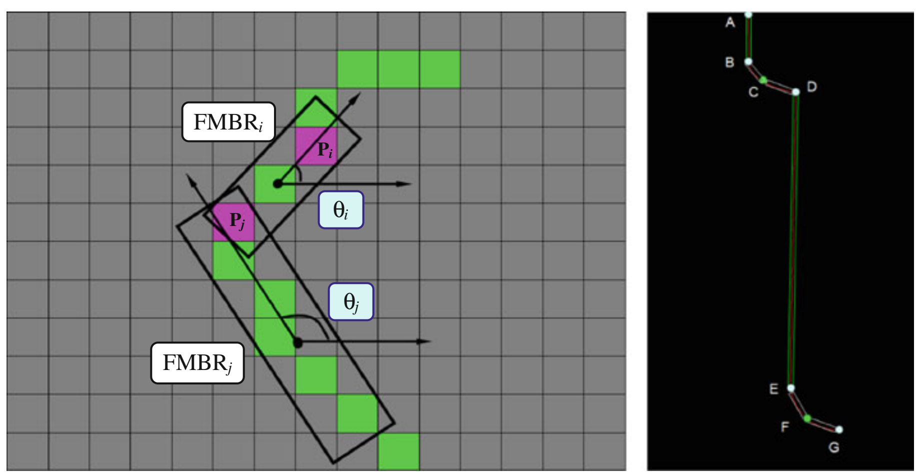
Determining the CP by FMBR

The representation of the contour can be simplified by dividing it into several continuous curves. Then end points of each curve are the CPs in this case. First, the contour curve is divided into several segments by FMBR. The intersection points are candidate points to find critical points. Next, the similar segments are merged by analyzing the change of FMBR. Two criteria are used to merge the FMBR: the angle between two FMBRs and the length ratio between FMBR.



(a) (b) (c)

Fig. 12 Contour extraction. **a** Still cutter, **b** rotating cutter under normal lighting, **c** composed image with back-lighting and contour extraction



(a) (b)

Fig. 13 Concept of FMBR. **a** Definition of FMBR, **b** an example of FMBR

If the angle is small and length ratio is close to 1, then they should be considered to be merged. Then, some candidate points will be removed, and the rest points should be CPs. Figure 14 shows an example of contour partitioning to determine CP by FMBR based algorithm.

This FMBR based algorithm is more stable than existing curvature-based algorithm. In the latter, the curvature is calculated pixel by pixel. Therefore, the algorithm is sensitive to noise points on the contour. While using this algorithm, the influence of noise points can be depressed effectively by setting the width of FMBR.

Contour classification

The objective of contour classification is to identify the type of partitioned segments: apparent contour or imaged cross section (see Fig. 3). In order to improve the calculating efficiency, the straight line will be found since only two endpoints of the

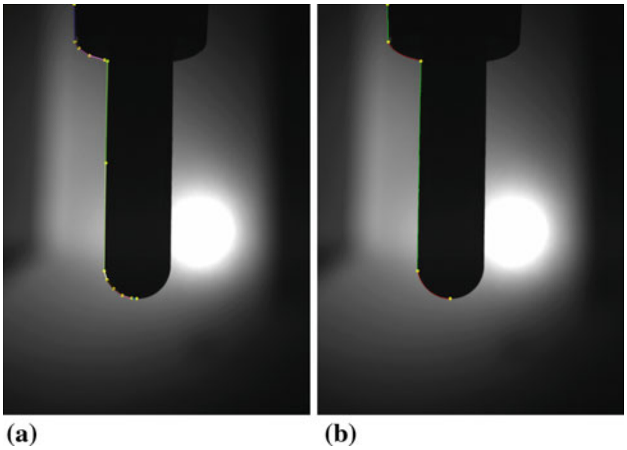


Fig. 14 Contour partitioning based on FMBR. **a** Candidate points, **b** CP by merging candidate points

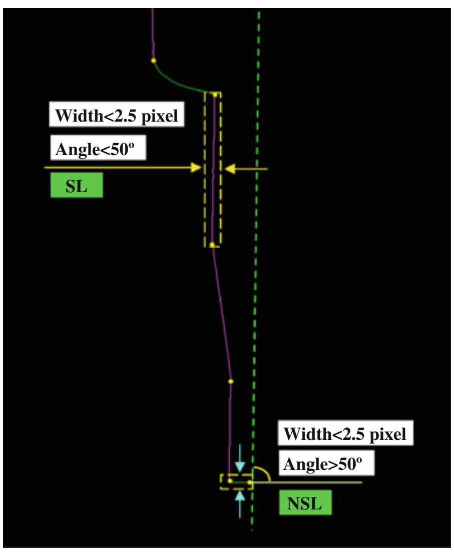


Fig. 15 Distinguish straight line (SL) and non straight line (NSL)

straight line segment are needed for 3D reconstruction. First, the segments are classified as straight line segments and non-straight line segments (Fig. 15). Then the elliptic arcs are recognized from non-straight line segments (Fig. 16).

Classification of the straight lines

In order to find the straight line segments, the Minimal-area Bounding Rectangle (MBR) of each segment is calculated. If the width of a rectangle is less than 2.5 pixels, it should be a straight line. However, for some smaller straight end mill, the last segment of the contour is a very short elliptic arc. The width of the short ellipse is usually less than 2.5 pixels. It will be classified as straight line incorrectly. Therefore, extra criterion should be added, which is the angle between MBR and image spindle axis. Since usually an elliptic arc

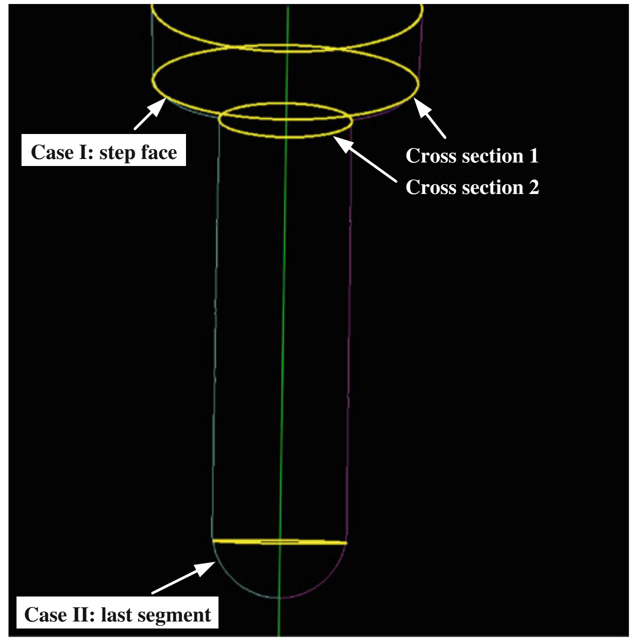


Fig. 16 Distinguish elliptic arc from non straight line

corresponds to a step face. The angle between the MBR (See Fig. 15) of an elliptic arc and imaged spindle axis is large. So the improved criteria are:

```
If Width < 2.5 pixels && Angle < 50°,
it is a straight line;
Otherwise, it is a non straight line.
```

Classification of the ellipse arcs from the non straight lines

If the contour segment is not the last segment of the contour, the elliptic arc corresponds to the step face on the cutter body. Then there is a sharp change in radius. Two endpoints of the segment can determine two cross-sections. Let their radius are R1 and R2. Their centers are C1 and C2. Calculate $E = |R1\text{-}R2|/\text{distance}\{C1, C2\}$. If the segment is an elliptic arc, E should be a large value due to a step face.

If the segment is the last segment of the contour, it should be elliptic arc (straight end mill) or circular arc (ball end mill). The angle between the MBR (See Fig. 16) of an elliptic arc and imaged spindle axis is large. So the criteria for the last segment are:

```
If Angle > 70°, it is an elliptic arc
Otherwise, it is a circular arc
```

3D reconstruction

Once the apparent contour is classified, according to the sections “Single view reconstruction method” and “Determining

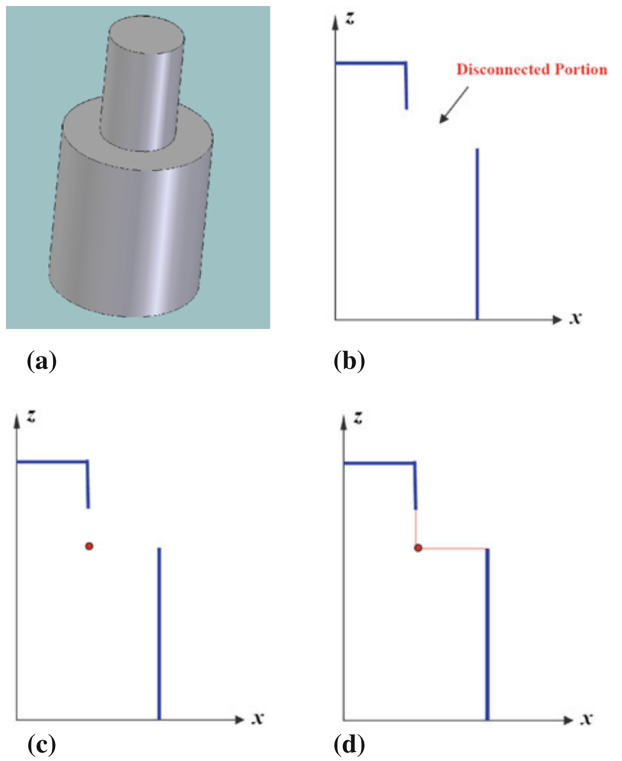


Fig. 17 Self-occlusion on an SOR and solution. **a** An SOR, **b** partial generating function, **c** determine intersection point, **d** recover generation function

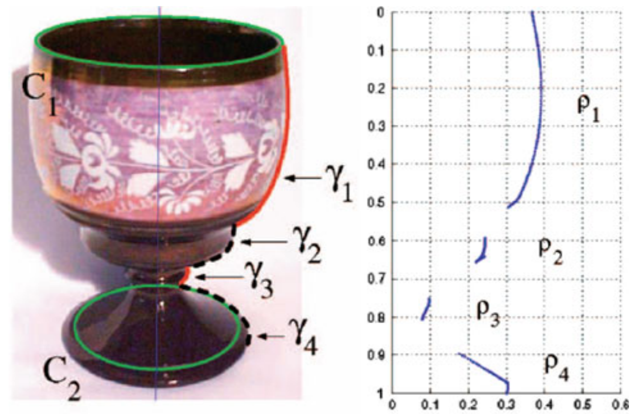


Fig. 18 A case of self-occlusion (Colombo et al. 2004)

the cross section of a cutter", one point on the apparent contour can define a cross section. If all the points on the contour are used, the SOR can be reconstructed as a stack of circular cross sections. From this model, the generating function of SOR can be extracted. This idea is implemented directly in Zhang et al. (2010). However, it is not necessary to use all the points on a straight line to calculate the cross sections but just two end points. In this paper, the designed contour classification algorithm can determine the apparent contour of straight line. Therefore the calculation efficiency of 3D reconstruction can be improved greatly.

Another consideration is self-occlusion which occurs due to discontinuities in the surface normal. For example, in the Fig. 17a there is a step face on the SOR. In this case, only the partial generating function can be obtained as shown in Fig. 17b. The partial generating function should be recovered further to get a complete one.

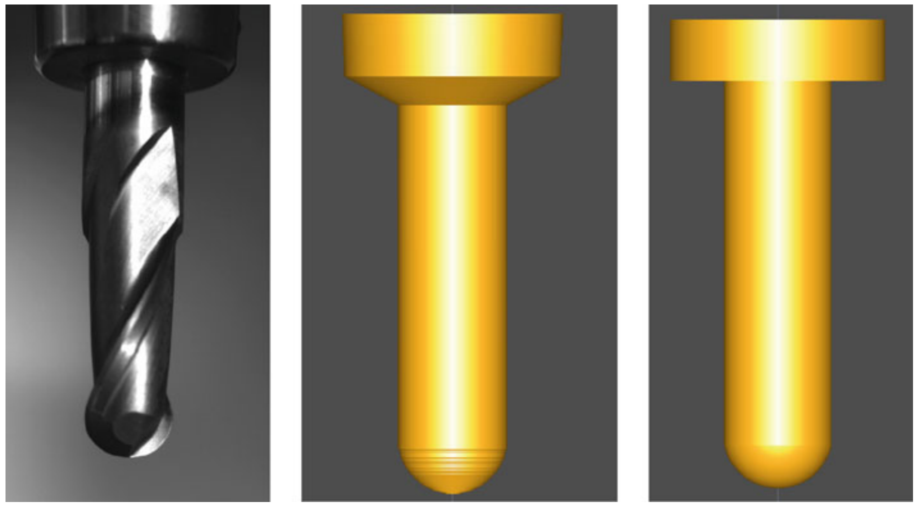
For a general SOR (Colombo et al. 2004) as shown in Fig. 18, it is difficult to repair the complete generating function because there are no enough geometrical constraints to follow to recover the disconnected curves. Fortunately, for a typical tool cutter, self-occlusion only occurs due to step face of cutter body. Therefore, it is a reasonable assumption that the missed apparent contour due to self-occlusion is straight line. So the straight line is used to recover the generation function (Zhang et al. 2010). First, find the self-occlusion. If there is an elliptic arc other than the last segment of the contour, generation function should be recovered. Then, find the possible intersection between two disconnected segments (Fig. 17c). At last, extend the body along z axis, whose diameter is small to the surface corresponding to the intersection (Fig. 17d). The example in Fig. 19 shows the result before recovery and after recovery. Notice that for the ball end mill, it is not good enough if only use the pixel to calculate the spherical part. The spherical surface is smoothed by fitting a circle and adding more points into the generation function. Once the complete generating function is obtained, its length offset and diameter offset can be extracted. For the length offset, the reference length is set by the calibration target.

Experimental verification

Simulation

The 3D reconstruction algorithm is first verified by simulation. The camera and cutter as well as backlighting are set in the virtual environment of software Autodesk 3ds Max as shown in Fig. 20. The 3D model of the cutter and the image of the cutter with backlighting are shown in Fig. 21. The parameters of the camera, cutter and lighting for simulation are listed in Table 1. First the camera is calibrated with a virtual camera calibration target following the procedure in Section 3. There are $3 \times 3 \times 3$ different locations to position the camera calibration target. Afterwards, the images of the camera calibration target are generated using 3Ds Max, and using the ellipse detection algorithm as well as the camera calibration algorithm, the calibrated camera model is obtained. The calibration results are given in Table 2.

While the cutter is rotating at 5 degrees per frame, 36 images are generated. They are overlapped to get a composed



(a) (b) (c)

Fig. 19 3D reconstruction from self-occlusion. **a** The cutter, **b** before recovery from self-occlusion and smoothing, **c** after recovery from self-occlusion and smoothing

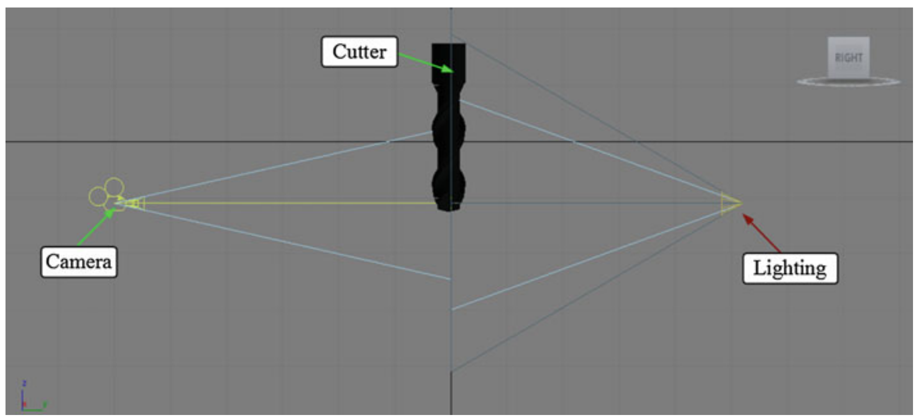
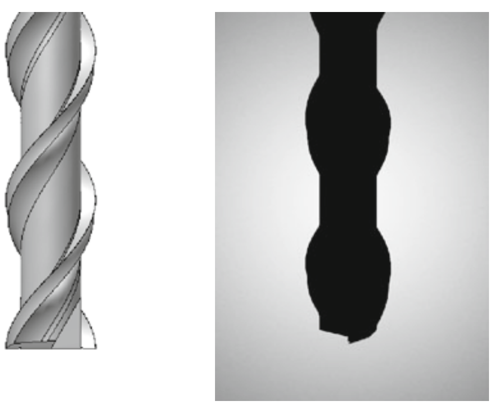


Fig. 20 Setup of simulation



(a) (b)

Fig. 21 The cutter and its image in simulation. **a** The 3D model of cutter, **b** the image of the cutter with backlighting

image (Fig. 22a). Then, the contour (Fig. 22b) is extracted and analyzed to get the apparent contour (Fig. 22c). The apparent contour is used to calculate the generating function. The 3D model (Fig. 22d) is built by rotating the generating function along spindle axis.

Table 1 Simulation parameters

Tool	Camera system	Lighting system
Type: Straight end mill	Viewing angle: 33.4 deg	Cone angle: 40 deg
Diameter: 6 mm	Shutter: 1/1000s	Luminance: 261 lx
Length: 30 mm	Aperture: 1.4	
Rake angle: 0 deg	Film speed: 200	
Relief angle: 20 deg		

The calculated radius and length offset are compared with the nominal values in Table 3. The results verify that the proposed method and algorithm are effective and accurate.

Prototype system on machine tool

The prototype of the vision system has been implemented on a Mori Seiki NVD1500/3AX CNC machining center as shown in Fig. 23. A Sony CCD camera (1024 ×

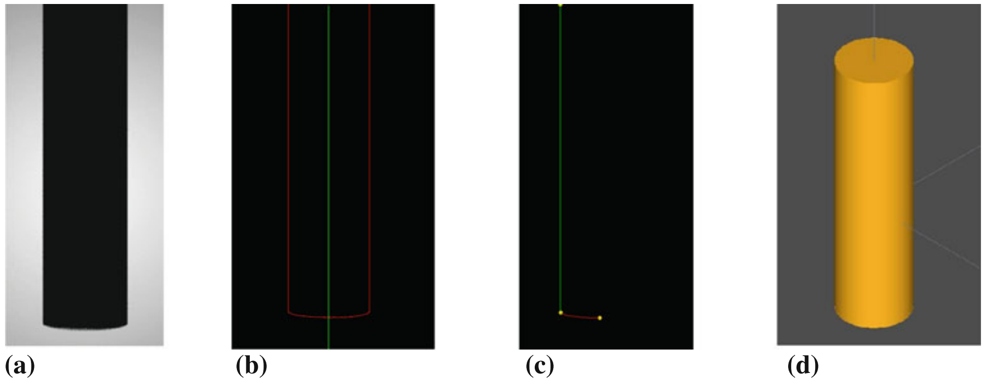


Fig. 22 3D tool reconstruction. **a** Composed image, **b** contour extraction, **c** contour classification, **d** 3D model

Table 2 Results of camera calibration for simulation

Intrinsic matrix (unit: pixel)	$K = \begin{bmatrix} 1600.009837, & 0, & 512.090147, \\ 0, & 1599.913916, & 384.019078, \\ 0, & 0, & 1 \end{bmatrix}$
Rotation matrix	$R = \begin{bmatrix} 1, & -0.000189, & 0.000006, \\ 0.000006, & -0.000276, & -1, \\ 0.000189, & 1, & -0.000276 \end{bmatrix}$
Translation vector (unit: mm)	$T = [-0.02055, -0.019162, 56.244327]$
Distortion coefficient	D=[0, 0, 0, 0]

Table 3 Results of 3D reconstruction (unit: mm)

	Radius offset	Length offset
Nominal value	3	30
Calculated value	2.966	30.023
Difference	0.034	−0.023

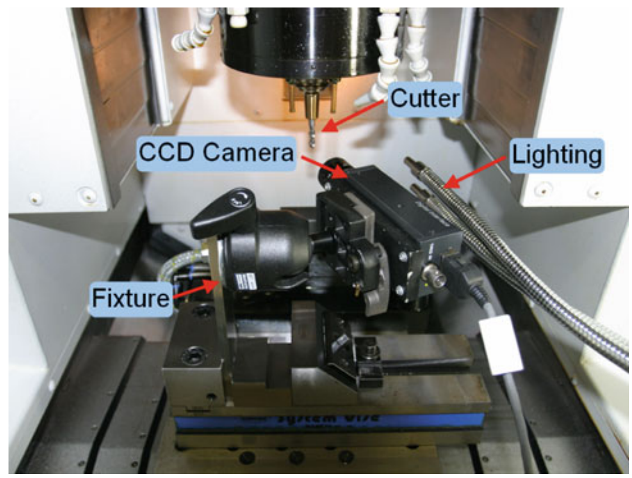


Fig. 23 Vision system on Mori Seiki NVD1500 machining center

768, 1/3 inch) is mounted on the worktable, which moves along X and Y axes. The spindle moves along Z axis. An 8 mm lens with a 1 mm extension is used in order to get better magnification. The working distance is around 60 mm. The GUI design of the prototype is shown in Fig. 24. The top-left window displays a dynamic video of the camera. The captured image as well as image processing result is shown on the bottom-left window. The cutter dimensions and 3D model are displayed in the center and right window respectively. The bottom panel is used for camera control and calibration. Ten different cutters are tested in the experiment as shown in Fig. 25. They consist of straight end mill and ball end mill. Their diameters vary from 1 to 6 mm.

First, the camera is calibrated on the machine tool. The designed calibration target is clamped onto the spindle. By moving the spindle and worktable to different positions, a virtual 3D calibration space of $18 \times 4 \times 18$mm can be established. 27 positions are used for the calibration. Figure 26 shows 9 points on one plane of the calibration space. The calibration results are given in Table 4.

Then, each cutter is loaded onto the spindle from tool magazine of the machine tool and move to the front of the camera. For each cutter, 36 images are taken while it is rotating (Frame rate: 30 fps, Spindle speed: 925 rpm). These images are processed to extract the generating function of each cutter following the algorithm in "Contour extraction and analysis" and "3D reconstruction". Figure 27 shows the 3D model built from the generating functions of cutting tools. The measurement process can be completed automatically within 10 s.

In order to verify the accuracy of 3D reconstruction of the vision system, the calculated radius of the end surface of straight end mill or sphere of ball end mill are compared with values measured from a commercial laser based tool setting system. The model of the tool setting system is BLUM Nano NT system, whose measurement repeatability is $\pm 0.2\,\mu$m. The results of comparison are shown in Table 5. The average measurement accuracy is verified around 35 microns with

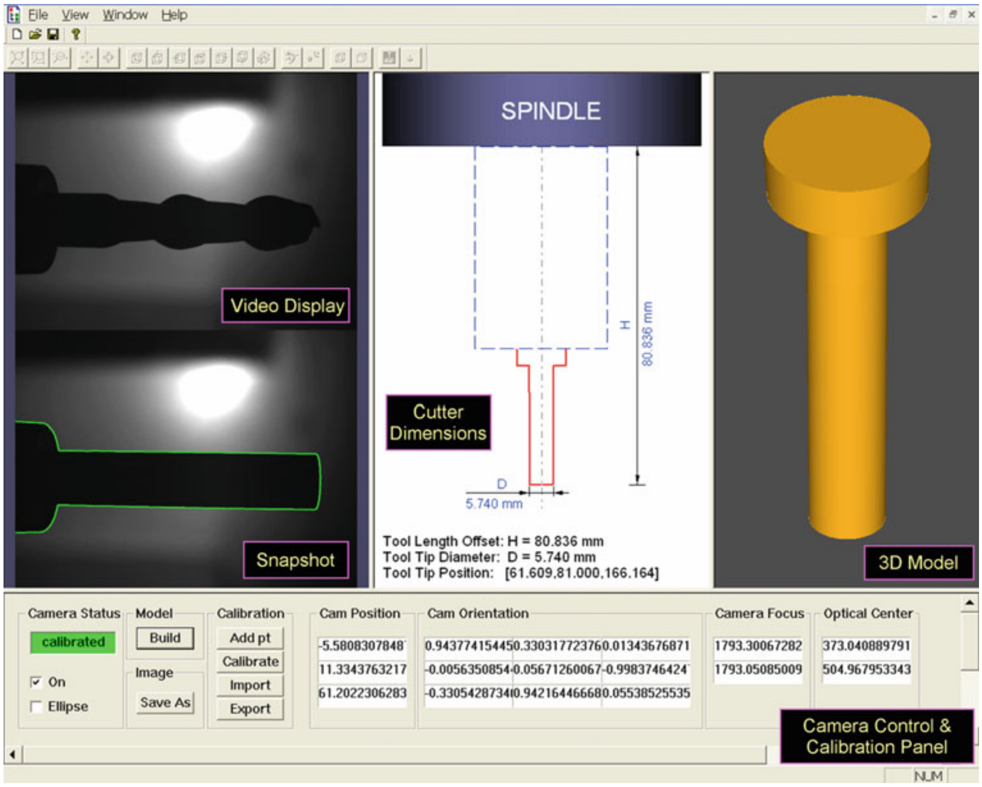


Fig. 24 GUI design of the vision system

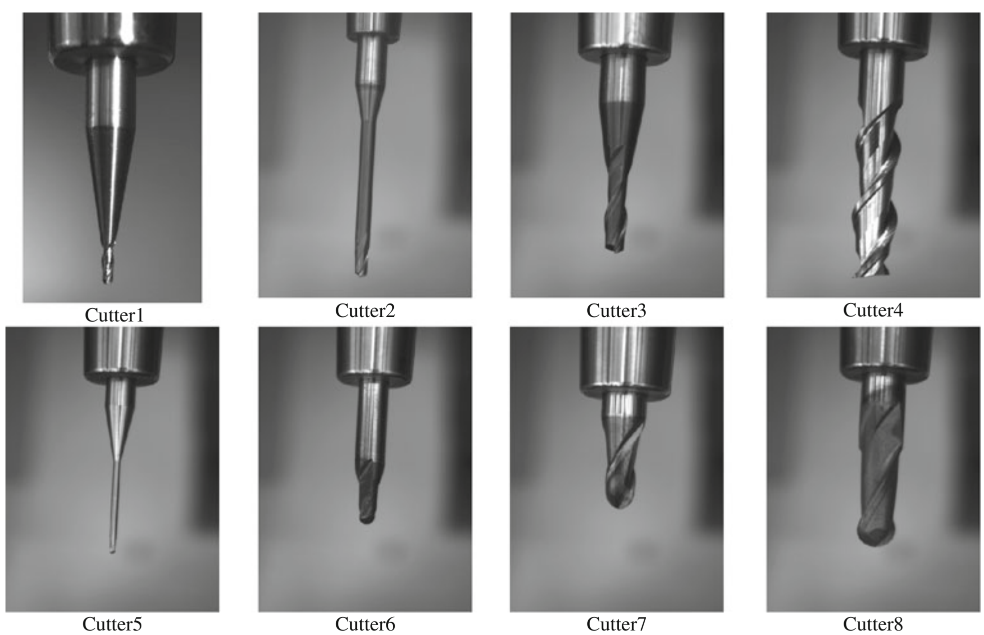


Fig. 25 Cutter used in the experiment

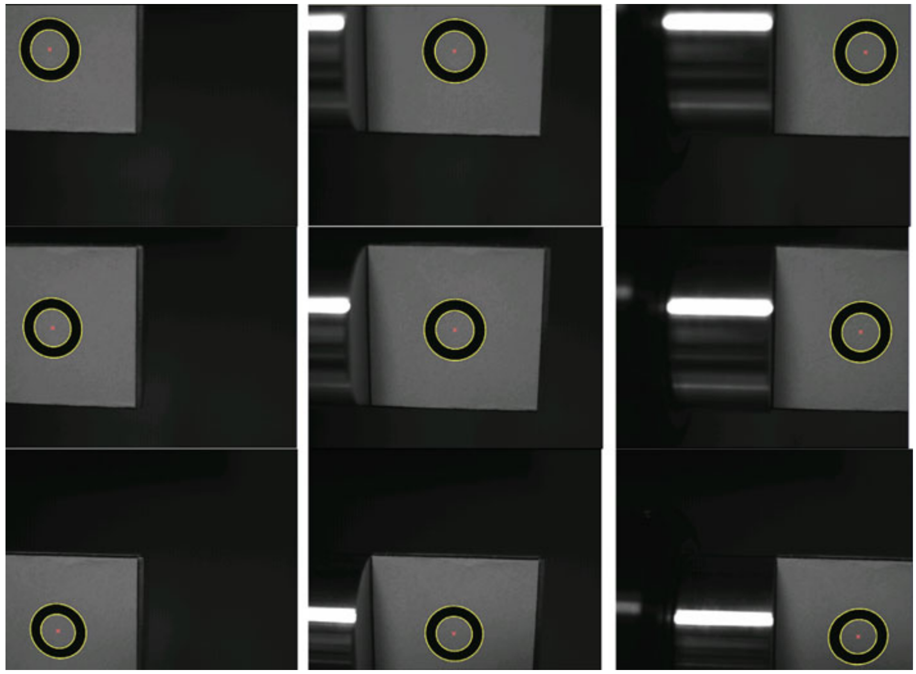


Fig. 26 The images of the calibration target at different positions

Table 4 Results of camera calibration

Intrinsic matrix (unit: pixel)	$K = \begin{bmatrix} 1,793.3007, & 0, & 373.0409, \\ 0, & 1,793.0509, & 504.9680, \\ 0, & 0, & 1 \end{bmatrix}$
Rotation matrix	$R = \begin{bmatrix} 0.9438, & 0.3303, & 0.0134, \\ -0.0056, & 0.0567, & -0.9984 \\ -0.3305, & 0.9422, & 0.0554 \end{bmatrix}$
Translation vector (unit: mm)	$T = [-5.5808, 11.3344, 61.2022]$
Distortion coefficient	$D = [-0.1984, 0.0046, 0.0001, 0]$

Table 5 Comparison of radius value with BLUM system

Cutter	Vision (mm)	Blum (mm)	Difference (mm)
1	0.544	0.511	0.033
2	1.005	0.995	0.01
3	1.531	1.488	0.043
4	3.025	2.999	0.026
5	0.508	0.463	0.045
6	1.549	1.494	0.055
7	2.034	1.994	0.04
8	3.022	2.992	0.03

Average difference: 0.0353

current optical configuration. Even better accuracy can be obtained if appropriate lens system with larger magnification is used.

Conclusions

A new method based on single view reconstruction algorithm has been proposed in previous research work to rapidly obtain the 3D model of a rotating milling cutter with tool holder. Only a single camera is mounted on the machine tool. This paper contributes to present the improved on-machine camera calibration procedure and new algorithms for contour analysis to improve the automation, accuracy, efficiency, and reliability of this system.

First, an improved camera calibration procedure is proposed. The accurate motion of machine tool axis instead of a physical calibration board is utilized to calibrate the camera on machine tool. Therefore the camera calibration can be conducted automatically by running NC code. This method has several advantages in automation, flexibility, accuracy, reliability and convenience of setup and coordinate alignment. Second, the idea to extract the complete contour is proposed. Capture the sequential images while the cutter is rotating. Then overlap them to find the maximum region occupied by a rotating cutter. This algorithm is very important to guarantee the reliability of the vision system in the actual machine tool environment. Third, the new algorithms for contour partitioning and classification are proposed, which is more reliable and robust. Straight line segments can be identified from apparent contours, which improve the calculating efficiency of 3D reconstruction. Finally, the vision system has been implemented and tested on different type machine tool with 8 kinds of cutting tools, which verified that the presented method is feasible and effective. The reconstructed results

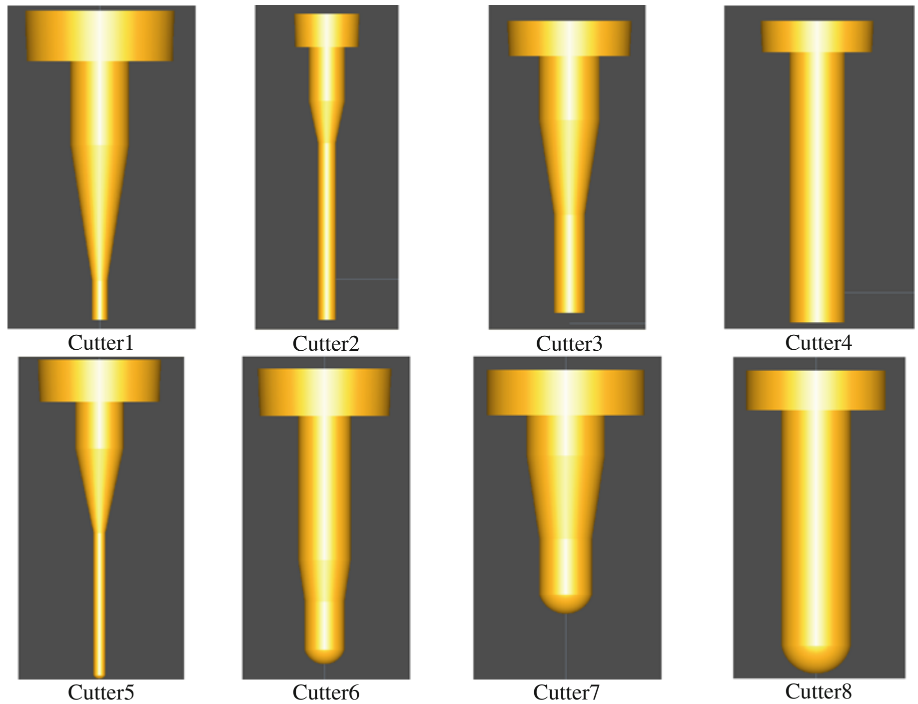


Fig. 27 3D reconstruction results

are compared with the measured values using laser based tool setting system. The measurement accuracy is around 35 micron while using an 8 mm lens and 1 mm extension. The accuracy of this system can satisfy the requirement of collision detection by machining simulation.

Acknowledgments The authors would like to express their sincere appreciation to Mori Seiki Co., Ltd, and MTTRF, Machine Tool Technology Foundation, for supporting this research.

References

Bouguet, J.-Y. (2010). Camera calibration toolbox for matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html#links.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. Cambridge, MA: O'Reilly Media Inc.

Colombo, C., Comanducci, D., Bimbo, A. D., & Pernici, F. (2004). Accurate automatic localization of surfaces of revolution for self-calibration and metric reconstruction. Paper presented at the *Proceedings of the 2004 conference on computer vision and pattern recognition workshop (CVPRW'04)* (pp. 55–60).

Faugeras, O. (1997). *Three-dimensional computer vision: A geometric viewpoint*. Cambridge, MA: MIT Press.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision* (2nd ed.). Cambridge, UK/New York: Cambridge University Press.

Klette, R. (2004). *Digital geometry: Geometric methods for digital picture analysis (The Morgan Kaufmann series in computer graphics and geometric modeling)*. Amsterdam: Elsevier.

Salvi, J., Armangué, X., & Batlle, J. (2002). A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition, 35*(7), 1617–1635.

Tian, X., Zhang, X., Yamazaki, K., & Hansel, A. (2010). A study on three-dimensional vision system for machining setup verification. *Robotics and Computer-Integrated Manufacturing, 26*, 46–55.

Zhang, Z. (2000). A flexible new technique for camera calibration. *Ieee Transactions on Pattern Analysis and Machine Intelligence, 22*(11), 1330–1334.

Zhang, X., Tsang, W. M., Mori, M., & Yamazaki, K. (2010). Automatic 3D model reconstruction of cutting tools from a single camera. *Computers in Industry, 61*(7), 711–726.

Zhang, X., Tsang, W., Tian, X., Yamazaki, K., & Mori, M. (2008). Automatic segmentation of the apparent contour for 3D modeling of cutting tools from single view. In G. B. Bebis, et al. (Eds.), *Las Vegas, NV, United states* (PART 2 ed., Vol. 5359 LNCS, pp. 772–781), Lecture Notes in Computer Science. Springer.