Int J Adv Manuf Technol (2000) 16:791–802
© 2000 Springer-Verlag London Limited

The International Journal of
Advanced
Manufacturing
Technology

Exact Tool Sizing for Feature Accessibility

T. Lim[1], J. Corney[1] and D. E. R. Clark[2]

[1]Department of Mechanical and Chemical Engineering, Heriot-Watt University, Edinburgh, UK; and [2]Department of Mathematics, Heriot-Watt University, Edinburgh, UK

This paper presents an algorithm for calculating the volume of a 2D-profile, accessible by a given diameter of milling cutter. The method is independent of the generation of cutter tool paths, and exploits facilities commonly found in kernel modellers. Exact results are obtained despite the simplicity of the procedure. As a proof of the concept, the algorithms have been implemented in the Heriot-Watt University feature recogniser as a pre-processor for a part programming system. The aim of these algorithms is to assist and optimise the selection of multiple tools for the machining of complex components. The methodology has applications in process planning research where it is currently common to assume that a single tool will machine each feature.

Although cutter selection is a critical step in planning the manufacture of components, computer-aided process planning (CAPP) systems rarely make any attempt to analyse the trade-offs involved. Perhaps this is because, traditionally, exact tool accessibility calculations have been viewed as a side effect of generating a cutter tool path. Consequently, accessibility calculations are not carried out explicitly but they appear implicitly in the results of a complex geometric algorithm (i.e. cutter path generation). Because this implicit checking of tool accessibility is carried out, downstream from the higher-level reasoning about set-up and sequencing, the results are generally available only after a detailed process plan has been generated.

Keywords: Feature recognition; Offsetting; Operation sequencing; Process planning; Tool sizing

1. Introduction

Many feature recognition systems have been described in the literature which identify machine removal volumes. However, all of these (including the few commercially available CAM packages that have feature recognition incorporated) require human interaction for tool assignment. Consequently, the process of selecting optimal tooling is a tedious, time-consuming, error prone task, especially if it involves numerous features. By automating the process of associating tools with features, the efficacy of current CAD/CAM systems can be increased considerably.

1.1 Tool Selection and Tool Sizing

Tool selection lies at the heart of manufacturing processes and is one of the key steps in process planning, allowing increased productivity and decreased machining cost [1]. Much work on tool selection is concerned with the effect on different production criteria (i.e. required surface finish, optimal cutting speeds, feeds and depth of cut, for prolonged tool life or low operating costs, etc.).

In contrast, tool sizing, as defined here, concerns tool-to-feature association. This process is a form of manufacturability analysis, which can also assist process planning. Our objective is to study the geometric constraints imposed on tool selection, enabling both a qualitative (i.e. visual) and quantitative indication of their effectiveness under actual machining operation. Such tests also enable a planner to discard or modify features that present unrealistic machining requirements. In this work, material removed by a tool during a machining operation is termed TRV (or tool removal volume).

Consider the example shown in Fig. 1. This component consists of a "butterfly" shaped pocket. Here, a planner must decide which tool (or combination of tools) would be the most effective (i.e. most efficient). Assuming a "one-off" component is required, and that a limited variety of tools are currently available the planner might choose a single tool approach (as in Fig. 1(*b*)(iii)).

However, when several pocket clearance operations are required the situation changes (Fig. 1(*b*)(iv)):

1. The single tool approach is frequently grossly inefficient – the smallest tool being required for all pockets [1,2].
2. Selecting a good (or optimum) combination of tools for roughing and finishing is a difficult task [1,3].
3. Every tool change carries a potentially costly time penalty [3,4].

Correspondence and offprint requests to: Dr J. R. Corney, Department of Mechanical and Chemical Engineering, Heriot-Watt University, Edinburgh, EH14 4AS, UK. E-mail: j.r.Corney@hw.ac.uk

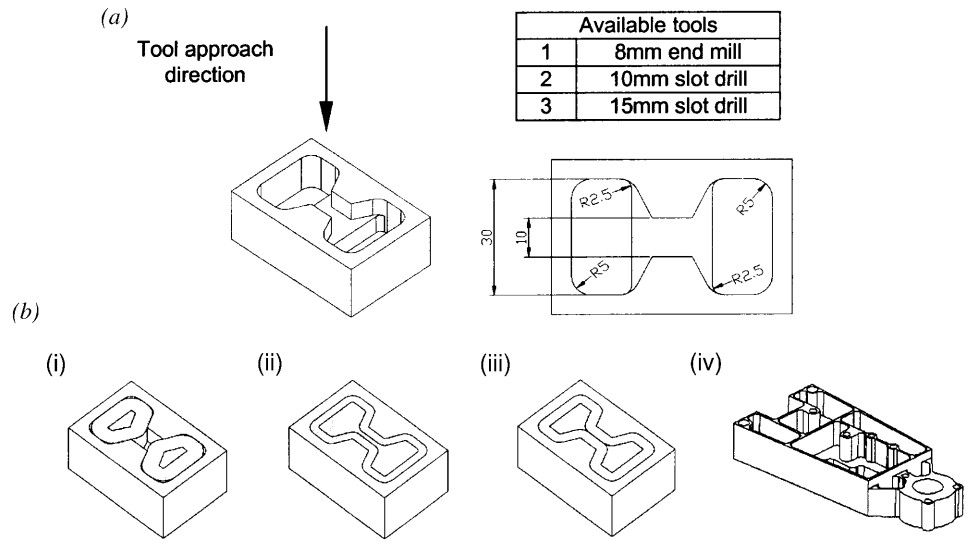

Fig. 1. Tool selection dilemma – which is/are optimal? (*a*) Example component (note the dimensions of the depression feature). (*b*) Volume clearance using two tools: in (i) a 15 mm slot drill is used, followed by an 8 mm end mill for finishing (ii). In (iii) volume clearance with a single 10 mm slot drill is used for material removal. (Note the cutter path within the volume representing the TRV.)

This paper presents an algorithm for computing the exact volumes cleared by a given tool for a selected machining feature. The objective is to maximise a tool's usage by identifying all the features it is capable of machining (either partially or fully) in a single set-up. The paper is structured in the following manner. Section 2 provides a brief overview of past work in the area of feature driven tool selection. This section also describes the capabilities of the Heriot-Watt feature finder, its geometric routines and its implementation. Section 3 presents the tool sizing utility known as QuickTest, and Section 4 illustrates the result of its implementation. The paper concludes with a brief discussion on future developments and considers how this approach could be used to support the automated selection of multiple tools for machining sets of features.

2. Literature Review

The documentation on feature recognition and intelligent/automated tool selection is extensive. However, few papers appear to have addressed the issue of optimising tool selection through the amalgamation of feature recognition, tool sizing and operation sequencing.

Preiss and Kaplanski [5] report some of the earliest work in this area. Based on heuristic condition-action rules, their program performs 3D-pattern recognition and evaluates localised features of interest. A "collide" utility written for straight or circular tool paths on faces that are planar or cylindrical (as formed by a 3-axis mill) was included. A rectangular shape approximates the selected cutter and its start and end trajectory is input manually. Boxes are created for the proposed cutter path and around the chosen face. If the boxes do not overlap, no collision occurs. Although the program provides a tool collision utility, it gives no indication as to the amount of material removed by the selected tool, either visually or numerically.

In Bala and Chang [4], features (e.g. slots, steps, and pockets) were represented as generic pockets. Cutter selection for pocket machining is instantiated by determining the pocket corner fillet radius (assumed constant for all corners). A cutter is then selected to fit within the pocket's bounding polygon without gouging into the walls of the pocket. Finishing and roughing tools are selected based on a minimum cutter motion criterion. An optimisation algorithm is then used to generate NC code that uses the least machining time.

More recently, Lee et al [6,7] developed the concept of a virtual boundary (VB) with feature-composition methodology for the planning and machining of generic virtual pockets (GVP). Under the constraints of the VB (defined as a boundary that can be violated by cutters during machining) a feasible cutter size can be found. Automatic tool selection and toolpath generations were optimised subject to several constraints (i.e. geometric, global, residual material, and machining accessibility). Depending on its complexity, decomposing a GVP into machining volume features (i.e. removal volumes) may cause suboptimal manufacturing plans. These volumes interact extensively with adjacent decomposed volumes sharing common faces. To resolve this, a manufacturing feature refinement step was introduced. This involves identifying and segmenting the entire machining area of the GVP into regions formed by merging adjacent decomposed volumes. The largest possible tool is then selected for each merged region in order to reduce the total machining time. This process has similarities to Bala and Chang's method [4] for finding a feasible cutter size by fitting the largest circle into the bounding polygon of the pocket.

Dereli and Filiz [1] presented OPT-TOOL, a system capable of selecting the best tool among many alternatives based on a maximum production rate criterion. A feature-recognition

system called FRS-PP extracts machining features including geometrical hints (e.g. datum and approach direction) while a knowledge-based system assigns candidate tools from a selected tooling database for each feature. In order to minimise tool change, cutting parameters are dynamically optimised and a single tool requiring minimum traversal is then selected for each feature. Diameters of cutters are determined by the size of the features to be machined. For example, in face milling, the required tool diameter is calculated using a multiplication factor. If the diameter found is bigger than the largest tool in the tooling database, the largest available tool diameter is selected. For hole features, the tool diameter is equivalent to that of the hole. Data for pockets and islands are input manually. Tool selection may either be automatic (the system selects corresponding tools for the machining features using rules), or interactive (the selection process can be changed or modified in each level of the system).

Eversheim et al. [3] proposed a feature oriented approach, which enables the user to select tools and inserts for machining processes taking into account geometrical, technological and "process-strategic" influencing variables. For example, the choice of milling cutter type depends chiefly on production criteria (e.g. required geometric specifications and surface accuracy, stability and parameters of the process, and machine capability). Selection rules are developed and transformed into real tool parameters according to standardised aspects on tool characteristics. Various machining situations within turning, drilling, and milling are collated and analysed on geometrical and technological criteria. On the basis of the geometry or manufacturing features determined in this way, the parameters influencing tool selection are identified and related to the tool's descriptive parameters. The descriptive attributes of the manufacturing features provide basic prerequisites for the selection and definition of tools.

Gupta [2] emphasises the importance of features in manufacturability analysis and also presents a method for sizing tools. In his work, sweeping a milling tool along a cutting trajectory creates a milling feature. The length of cutting trajectory depends on the tool size. To facilitate faster machining, tools that require the smallest cutting trajectory are selected. Drilling features are recognised and created by applying similar concepts. In effect, tool sizing is used to identify machining features.

A cutter selection methodology, known as process modelling was introduced by Wu and Liu [8]. Based on production criterion and available manufacturing facilities, their technique provides an optimum selection of cutters for machining a component. They interpret the feature recognition problem as a cutter selection problem and consider six predetermined cutter approach directions. Their technique emphasises modelling the manufacturability of a component in order to select a common cutter for machining multiple faces. A machining process can, therefore, be represented by a combination of tool approach direction and cutter. Multiple faces that share a common cutter are then interpreted as a machine feature.

In summary, it would appear that work on automated tool selection (or sizing) has tended to favour a single tool approach (e.g. assuming a single tool would be used to clear an entire pocket). Furthermore, the few researchers investigating multi-tool machining of pockets have tended to limit their studies to polyhedral objects, or only implemented procedures in 2D systems [9].

2.1 Offsetting

Offsetting has important applications in NC tool-path generation, clearance checking, and so on [10]. Bala and Chang [4] use offsetting to identify feasible cutter motion regions, whereas Gupta [2] uses offsetting to generate the cross-section of a tool's swept volume. In this work, given a tool, offsetting deduces both a feature's effective removal volume and its maximum tool-path boundary. [Furthermore, we distinguish between tool paths (i.e. cutter paths generated for area clearance) and a tool path boundary (i.e. a path/s whereby a cutter can circumnavigate within or around a feature without interference. It also describes the region that allows maximum cutter travel).] Since a feature's 2D profile consists of lines and arcs, offsetting can easily be carried out analytically, as lines and circular arcs are invariant under offsets.

More importantly, the offsetting routines used in this work (part of the API of a commercial kernel modeller [11]) carry out both trimming of overlapping profiles and filleting (i.e. filling gaps in the offset profile with arcs) automatically (Fig. 2). See [10] for a description of the offsetting operation used for this work. See also [12–15] for further information.

2.2 The Heriot-Watt Feature Finder (HWFF)

The work on tool sizing reported here uses as its starting point the output of the Heriot-Watt feature finder (HWFF) [16–19]. The algorithm operates in three stages:

1. Given a tool approach direction, a path generation function is called which considers each vertical face (relative to the machining direction) in isolation. Traversing from left to right, possible tool paths across the surfaces of the component are created. Since the start and endpoints of the paths share vertical edges, a directed graph is formed. Paths lying outside the boundary of a face are known as void-paths and are used to identify "open" features (i.e. slots and open pockets), while paths lying on a face are known as face-paths.
2. A cycle identification function analyses this directed graph. Using the orientation of the faces, closed non-intersecting cycles of paths found during the search are identified as protrusions, depressions, or open features.
3. A feature volume generation function constructs a 2D profile of individual cycles and sweeps it with its defined range.

The feature finding algorithm employs heuristics at each step of the process, thus reducing the number of paths and cycles generated. This results in an efficient and robust recognition system. Features are found and feature volumes are created directly from the B-Rep data structure without the complexities involved in a rule- or hint-based approach. The algorithm requires no feature libraries, consequently no pattern matching or subgraph comparisons are required.

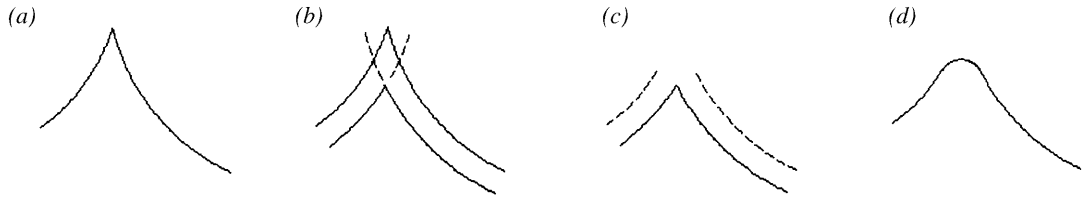

Fig. 2. Profile offsetting. (*a*) Original profile. (*b*) Offset inward and trimmed. (*c*) Offset outward. (*d*) Filleting offset profile.

3. Associating Tools with Features

After the recognition process is completed, all valid features (i.e. features accessible with respect to the tool approach direction) are placed in a list and sorted into respective levels. Levels represent precedence relationships between features. Implicit in the level structure is information such as the maximum depth/height of the features.

Figure 3(*c*) shows the HWFF user interface after features have been sorted into levels to assist operations planning. The tree control and the associated menus provide utilities for feature modification (e.g. change in *z*-depth range), removal and tool sizing. Any changes made are immediately echoed in a summary window (i.e. in the left column of the MLS user interface in Fig. 3(*c*)).

3.1 QuickTest – a Utility for Tool Sizing

The tree control of the MLS (Fig. 3(*c*)) allows individual features to be selected for analysis. Each feature is associated with a 2D profile, a *z*-depth range and a volume (representing its swept area, as shown in Fig. 3(*d*)).

Often, in feature assisted (or automated) tool selection for machining depressions, the feature's profile is analysed to

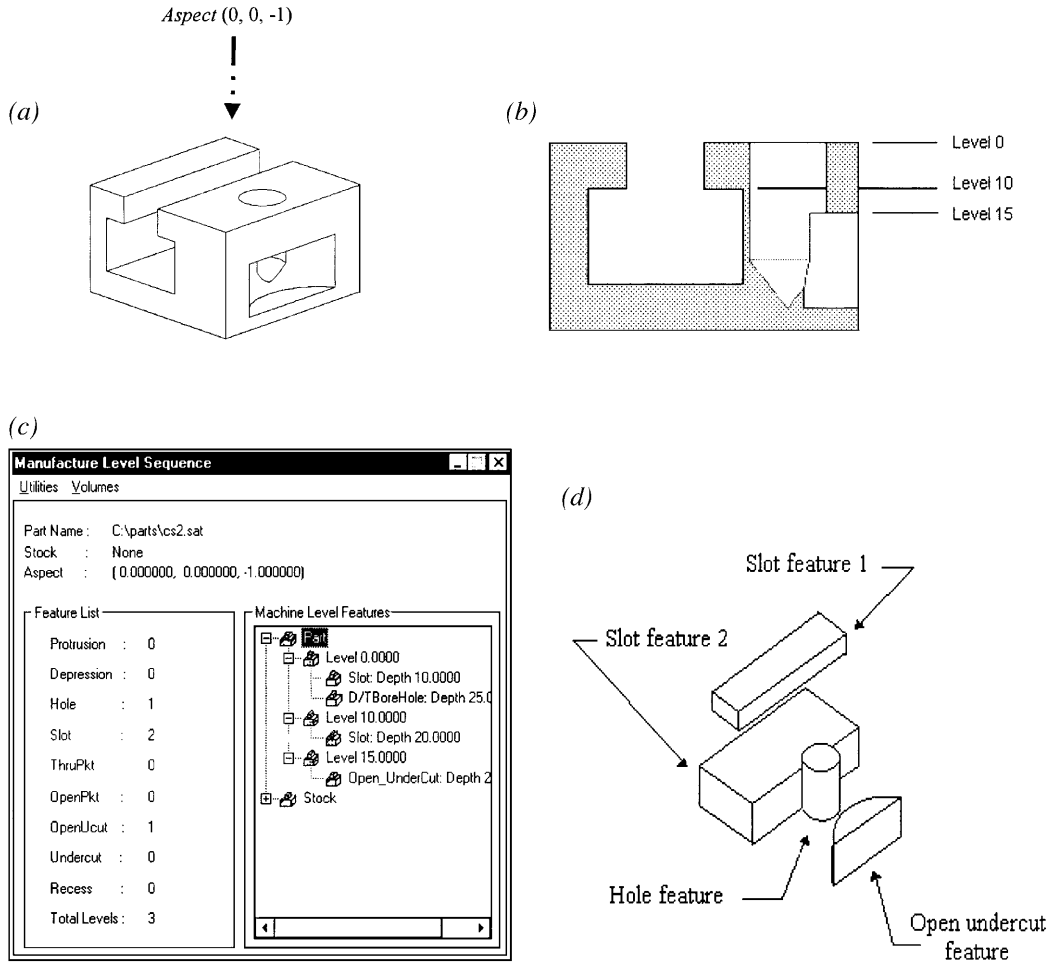

Fig. 3. Sorting features into precedence levels for post-processing. (*a*) Component 1. (*b*) Cross-section of component and respective machining levels. (*c*) HWFF post-processing user interface – manufacture level sequence (MLS). (*d*) Machining volumes returned by the feature recognition algorithms.

determine its corner radii [2,4]. The tool corresponding to the smallest radius in the pocket profile is usually selected since it guarantees complete material removal. In other words, the nominal volume of the feature will only match the machining volume if the smallest tool is used. Similarly, for drilling features, a drill is selected based on the profile's diameter and/or the conical tip angle formed at the feature's base. Systems developed using these geometric constraints favour a single tool approach and tend to ignore (or discard) those features for which no correct tools are available.

In contrast, QuickTest attempts to size a given tool to a selected feature regardless of its geometry (i.e. corner radius) and can handle pockets with multiple islands and dissimilar heights. The desired output of the QuickTest algorithm is a tool boundary path describing the location of the tool end, subject to the constraint that the tool must never interfere with the boundary of the selected feature. Figure 4 exemplifies the process of tool sizing:

1. An initial offset (Fig. 4(i)) is performed on the feature's profile (pocket profiles are offset inwards and islands outwards) using the selected tool's diameter. The purpose is twofold; initial offsetting determines whether the tool can access the feature for material removal; secondly, it determines the area in which the tool can move freely without interference. If an error occurs during the offsetting (i.e. the profile offsets into itself so that it returns nothing), the tool is deemed oversized.
2. If the offset island profiles are found to intersect they are united, Fig. 4(ii).
3. Profiles are offset by the tool radius but in an opposite sense (i.e. pocket outwards, islands inwards) as in Fig. 4(iii).
4. To obtain the maximum tool-path boundary, an intersection operation is performed between the offset pocket profile and the united island profile/s (Fig. 4(iv)).

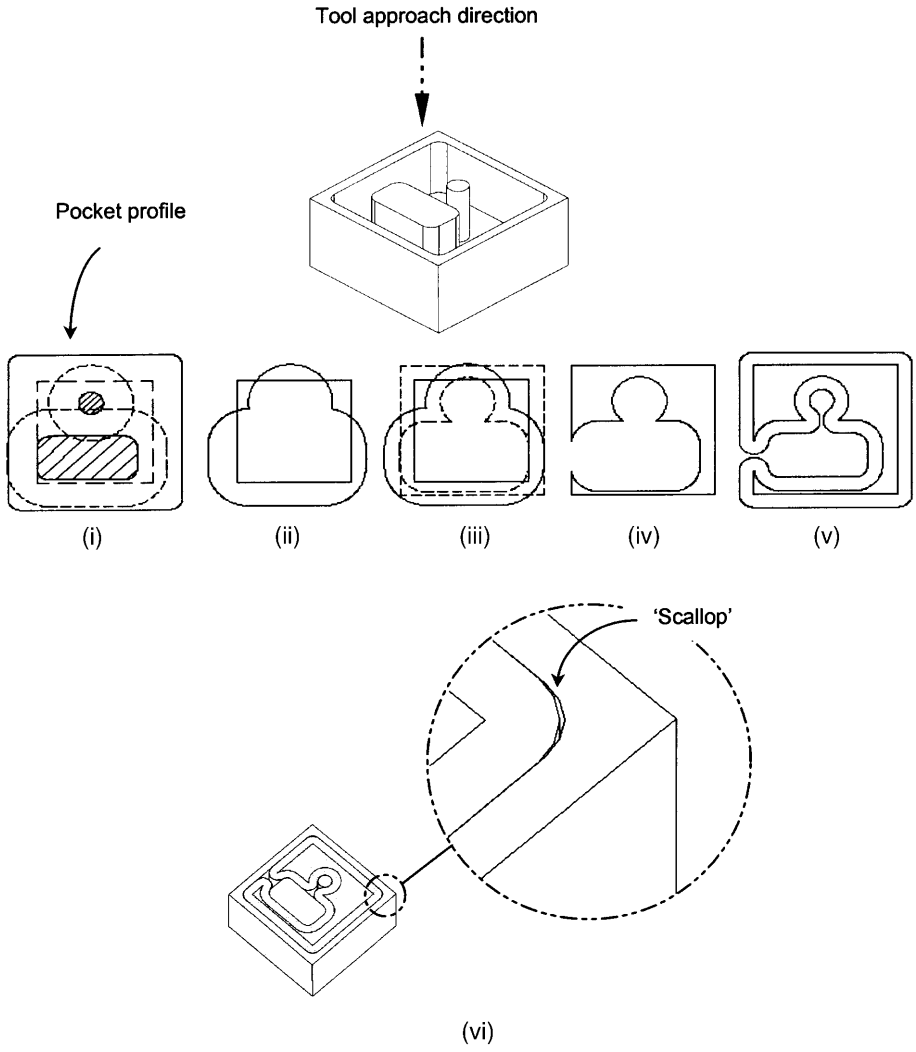

Fig. 4. Offsetting in QuickTest. (i) Offset profiles (dashed lines) of the islands and pocket by tool diameter. (ii) Intersecting island profiles are united. (iii) Offset united islands and pocket profile by tool radius. (iv) Unite offset united profiles with sheet plane. The result is the exact boundary path accessible by the tool. (v) By offsetting this tool path by its radius and sweeping it by the feature's range, the TRV is generated. (vi) Screen dump showing the TRV resulting from the use of an oversized tool (note the "scallop" caused by an oversized tool).

5. Lastly, the tool path boundary is offset outward by the tool radius (Fig. 4(v)) and swept to generate the TRV.

Within the above procedure the following check is made to determine if the pocket contains islands of differing heights (referred to as **LevelSort** in the pseudo-code):

1. Two sheet planes are created and positioned at the upper and lower bounds of the depression feature and imprinted onto the part model to identify the wire profiles of any islands within the pocket.
2. A ray-firing procedure similar to the visibility checks used in [16,17] then establishes the height of each island (which may not extend to the top of the pocket). Islands are ordered by range, and pocket profiles generated (by imprinting) at the top of each island.

The resulting output from the **LevelSort** function is a list of profiles representing feature boundaries at different *z*-depths.

Pseudo-code for the implementation is presented in Fig. 5 and the terminology adopted loosely corresponds to that of the ACIS® kernel modeller used [11]. For example, the offsetting of a single planar wire (i.e. a wire body) might result in several disjoint profiles (represented as separate shells in the wire body returned), as seen in Fig. 1(*b*)(i). Furthermore, the wire body used to represent the feature profile orientates pocket boundaries in a different sense to island boundaries. The function **wirebody_offset** then offsets each profile in a direction depending on its sense. The logic of the algorithm is also represented in flowchart form in Fig. 6.

4. Implementation and Results

This section presents a "proof-of-concept" in the form of the implementation and results obtained for three components.

Component 1

Figure 7 illustrates the tool sizing effects of the component shown in Fig. 3.

Volumes associated with open features (e.g. slots) have one or more faces through which a cutter must pass. TRVs are

(a)

```
Function: QuickTest_Accessibility
//global variables initialised in main body of Function: QuickTest…
Bod = body (i.e. the part); D = aspect vector; tol = 10 * res. of modeller;
CPList = list to hold cutter paths; TRVList = list to hold TRVs;
fp = profile of selected feature; r = depth/height range of fp;
Dia = tool diameter;  Rad = tool radius;

//… initialise local variables…
Ofp = wirebody_offset (fp, tol, D);
PL = cover_profile (Ofp); PL1 = copy_entity (PL);
//create lists to hold wire bodies and shells
wire_list = ceate_list(); shell_list = create_list();
trans_entity (PL1, (D * tol));          //translate profile
imprint (PL1, Bod, wire_list);          //and check for islands…

If wire_list.count() == 0 then          //no islands
 temp_CP = wirebody_offset (fp, Dia, D)   //initial offset…
 //check if offset procedure returns a wire body
 if is_wirebody (temp_CP) == TRUE then  //feature accessible with
 // tool …
  get_shells (Ofp, shell_list);
   for each shell, shell, in shell_list do
    Temp_prof = get_profile (shell);   //return wire body
    CP = wirebody_offset (Temp_prof, Rad, –D); //final offset
    CPList.add (CP);  //and add to CPList
   end for
 end if
```

```
else          //islands present…
  //sweep profile to create a temporary volume
  Temp_vol = create_volume (fp, r, D);
  //compute Boolean intersection withTemp_vol and Bod
  Temp_vol = Temp_vol ∩' Bod;
  Call Function: LevelSort (Temp_vol, level_list);
  for each profile, profile, in level_list do
   temp_CP = wirebody_offset (profile, Dia, D);   //1st offset…
    //check if offset procedure returns a wire body
    if is_wirebody (temp_CP) == TRUE then
     get_shells (temp_CP, shell_list);
      for each shell, shell, in shell_list do
       for each shell, next_shell, shell_list.next() do
         //check for intersection…
         if inters (shell, next_shell) == TRUE then
          shell = shell ∪' next_shell;  //unite shells
        end if
       end for
        CP = wirebody_offset (shell, Rad, -D);   //final offset
        CPList.add (CP);
      end for
    end if
  end for
end if
End Function: QuickTest_Accessibility
Call Function: Generate_TRV
```

(b)

```
Function: Generate_TRV
lr = thickness of individual layers (i.e. levels)
for each CP, cp, in CPList do
 temp_tvr = wirebody_offset (cp, Rad, D)
 //create TVR
 temp_tvr = sweep (temp_tvr, lr, D)
 TVRList.add (temp_tvr)
end for
End Function: Generate_TRV
```

Fig. 5. QuickTest tool sizing algorithm. (*a*) Function: QuickTest_accessibility. (*b*) Function: Generate_TRV.

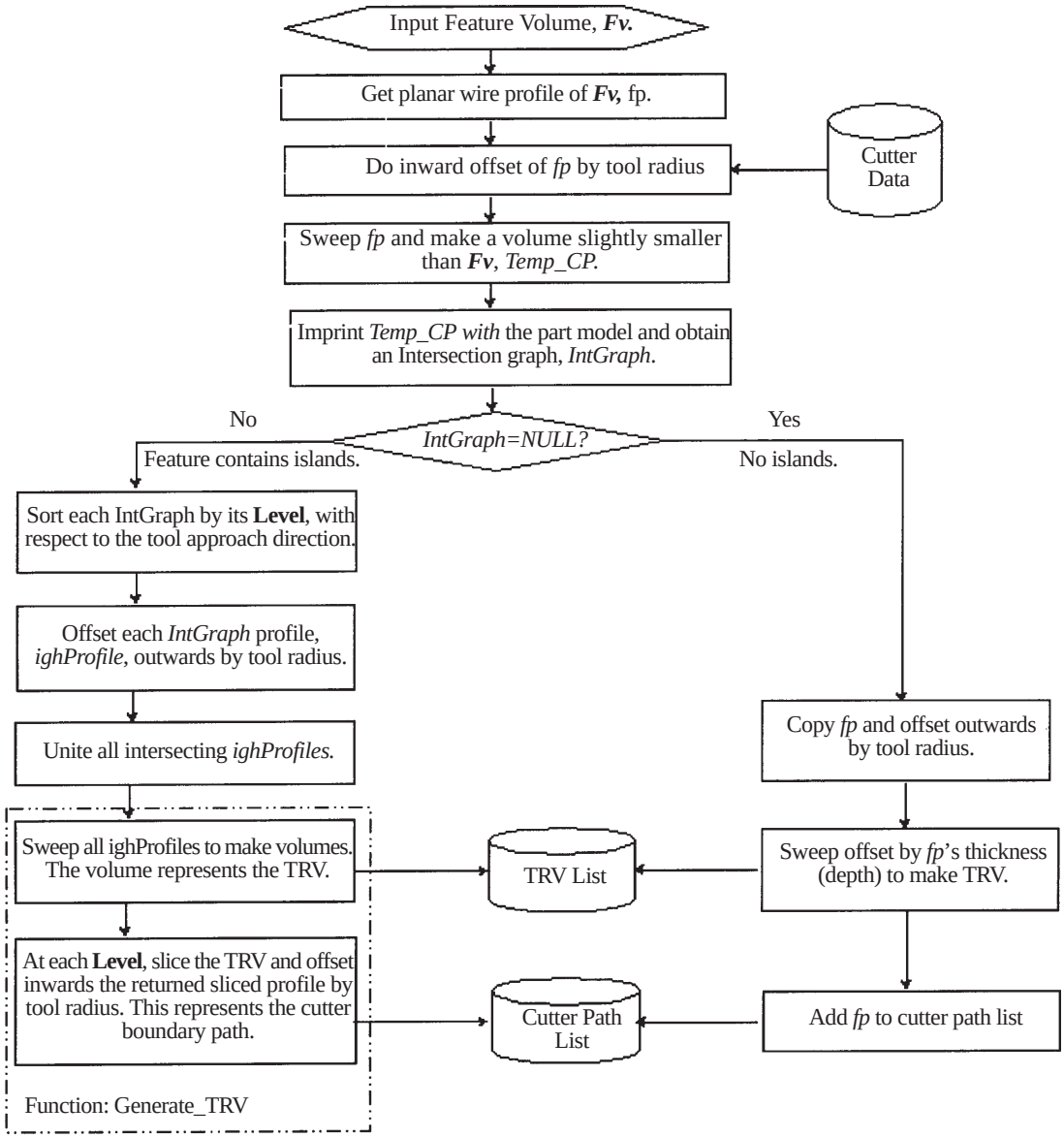

Fig. 6. Flowchart giving an overview of the QuickTest tool sizing algorithm.

generated similar to that of closed features, except that here initially the "open" faces (i.e. void paths in the feature profile) are offset outwards by a specified distance (usually a tool radius, Fig. 7(*a*)). However in cases of "open", overhung features such as the T-slot and side pocket in Figs. 7(*b*), 7(*d*) and 7(*e*) the tool diameter is used to ensure accessibility. The QuickTest algorithm is then applied to the enlarged feature.

Component 2

This is a test component from the NIST repository (http://www.parts.nist.gov) entitled "part03". The geometry of the central depression on this part is detailed in the Appendix (Fig. A1(*a*)). Figure 8(*d*)(i) presents the result of the tool-sizing algorithm for a 20 mm diameter cutter (enabling 100% material removal). The resulting TRV has a volume of 339 005 mm^3 (feature depth of 14 mm). Given that the depression has such a large basal area, a combination of tools for greater efficiency rather than this single-tool approach might be considered. Figure 8(*d*)(ii) presents the result of the tool-sizing algorithm for an 80 mm diameter cutter (enabling 88% material removal). The resulting TRV has a volume of 297 815 mm^3 (feature depth of 14 mm).

Component 3

Figure 9(*a*) shows a component having several complex isolated pocket features, two of which are tested against three different tool sizes (the geometry of these depressions is detailed in Fig. A1(*b*) in the Appendix. Figures 10(*a*)(i) to 10(*a*)(iii) show

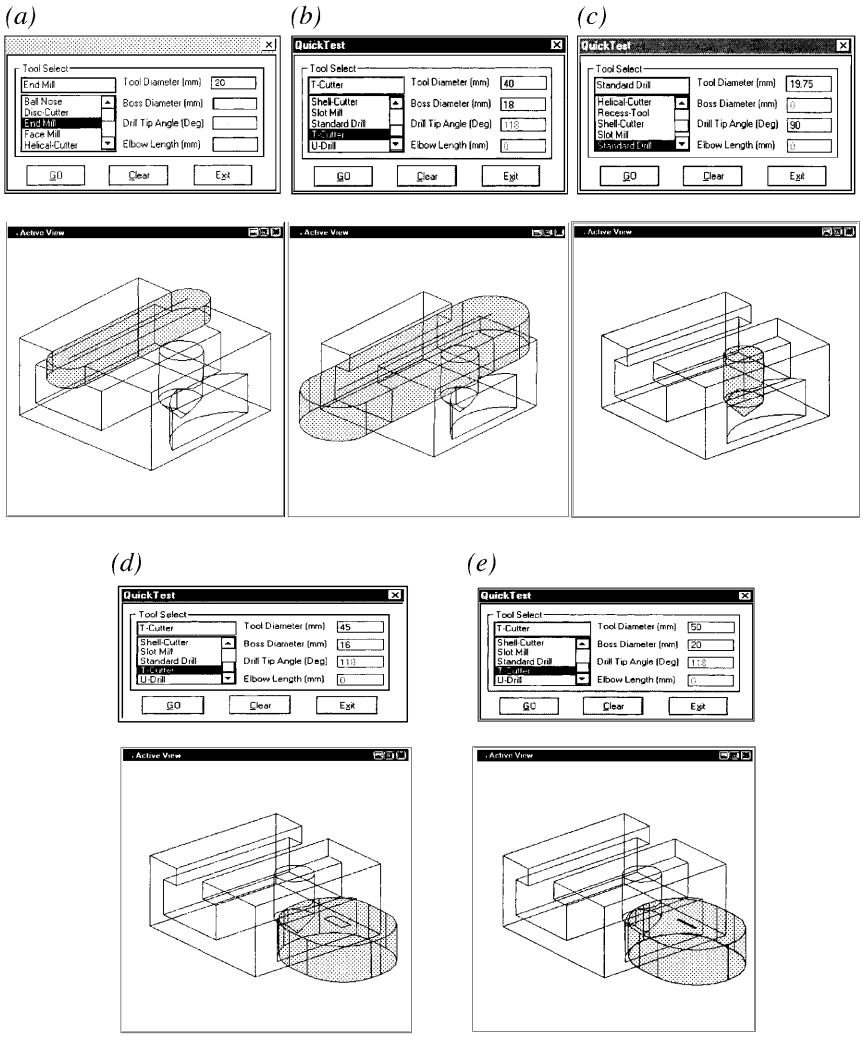

Fig. 7. Results of QuickTest with selected tools. The TRVs are the tinted volumes and the cutter paths are indicated as lines within each TRV.

Table 1. The TRV volumes resulting from Fig. 10.

Tool size (end mill) (mm)	Illustration number	Depression TRV (depth = 20 mm) (mm^3)	Open pocket TRV (depth = 20 mm) (mm^3)
10	i	111922	47614
35	ii	96197	44713
40	iii	43348	43708

the result of the tool sizing algorithm for 10, 35 and 40 mm diameter cutters. These enable 100%, 86%, and 39% material removal, respectively. Figures 10(*b*)(i) to 10(*b*)(iii) show the result of the tool sizing algorithm for 10, 35 and 40 mm diameter cutters when applied to the open pocket feature, these enable 100%, 94%, and 92% material removal, respectively. The resulting TRV volumes are detailed in Table 1.

5. Determination of Accessible Depth

The algorithm described in Section 3.1 also supports the machining of $2\frac{1}{2}$-D protrusions and depressions embedded in complex (i.e. non-$2\frac{1}{2}$D) geometry. The increased complexity of these situations can be understood by noting that all the objects shown in the previous section have features whose depth of cut does not vary with the tool size. However, Fig. 11 illustrate a tool path associated with a protrusion that sits on a concave face. The resulting TRVs generated for this feature show how the choices of tool size have consequences for both 2D and 3D accessibility.

The basic tool-sizing algorithm handles these cases with a procedure that samples the accessible depth around the TRV profile prior to sweeping (see Section 3.1, step 5). Since the feature might lie either on a convex or a concave face, the accessible depth is checked using two offset feature profiles. The first profile is offset by a small amount (ten times the kernel modeller's tolerance) and is used to establish the accessible depth around the bottom of the feature. The second profile

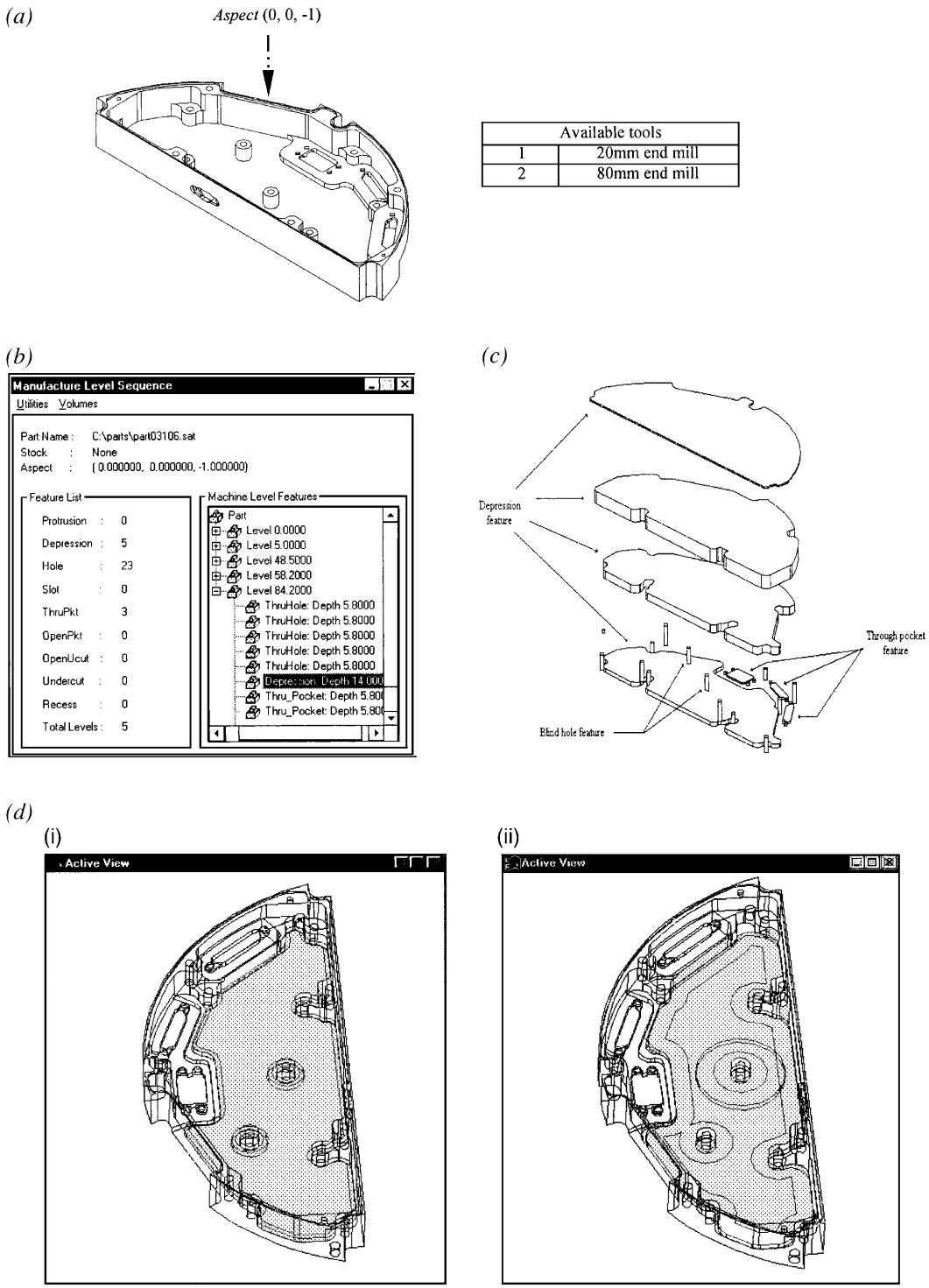

Fig. 8. Visualisation of TRVs and cutter paths for a depression feature of part03. (*a*) Component 2. (*b*) Manufacture level sequence of features for "part03". (*c*) Extracted features and their corresponding delta volumes. (*d*) QuickTest tool sizing results for a selected depression machining feature.

is offset by the diameter of the cutter and is used to find the depth of cut accessible at the edge of the tool.

In both cases depth is found by firing rays in the direction of the tool approach vector from each vertex on the offset feature profile. Functions in the kernel modeller return the point at which each ray first intersects the model, the shortest value is returned. All the depth values, from the offset profiles, are compared and the smallest used for the final sweeping operation. Although

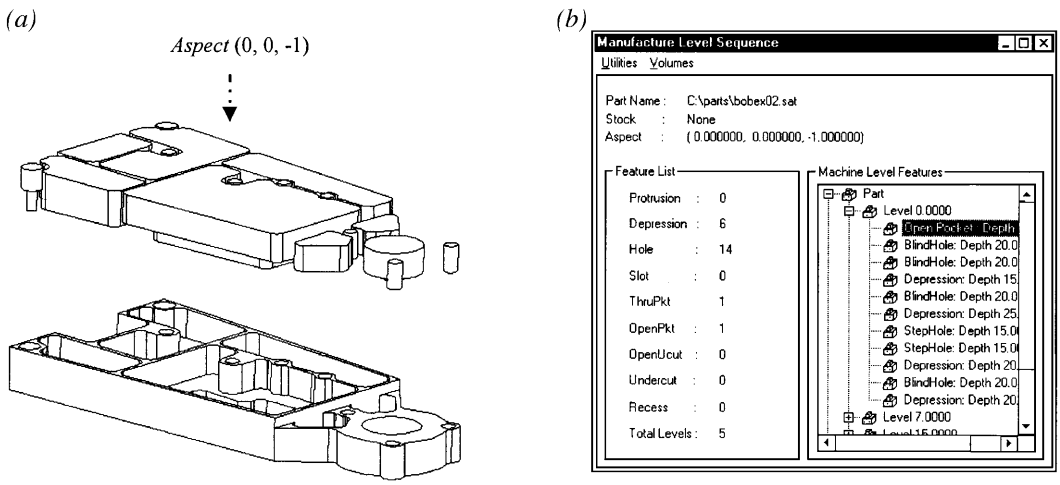

Fig. 9. Component 3 and it's machining levels. (*a*) Component 3 and its machining features. (*b*) Manufacture level sequence of component 3.

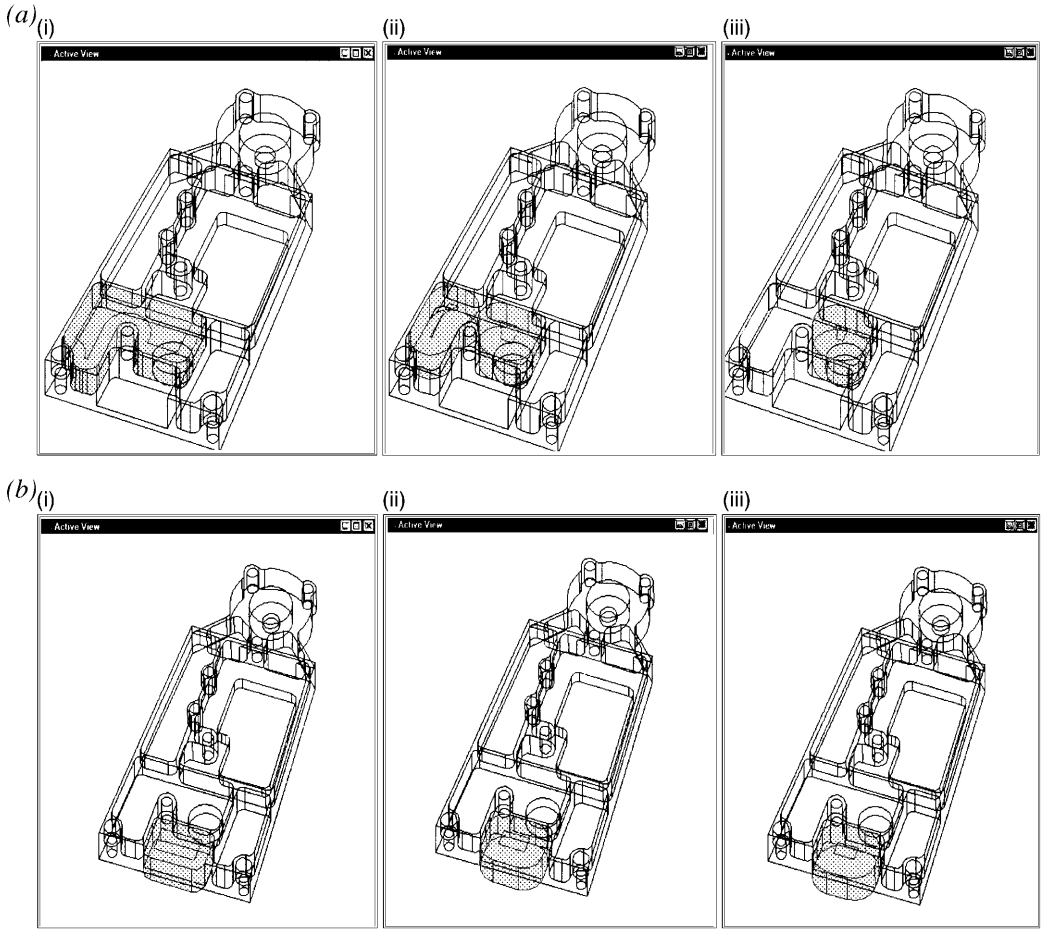

Fig. 10. Tool sizing for two selected features of component 3. (Note the varied volume/s accessible by each distinct diameter of cutter.) (*a*) Results of QuickTest tool sizing for a depression feature. (*b*) Results of QuickTest tool sizing for the "open pocket" machining feature.

qualitative in nature, the results shown in Fig. 11 illustrate how the algorithm deals with more complex geometry.

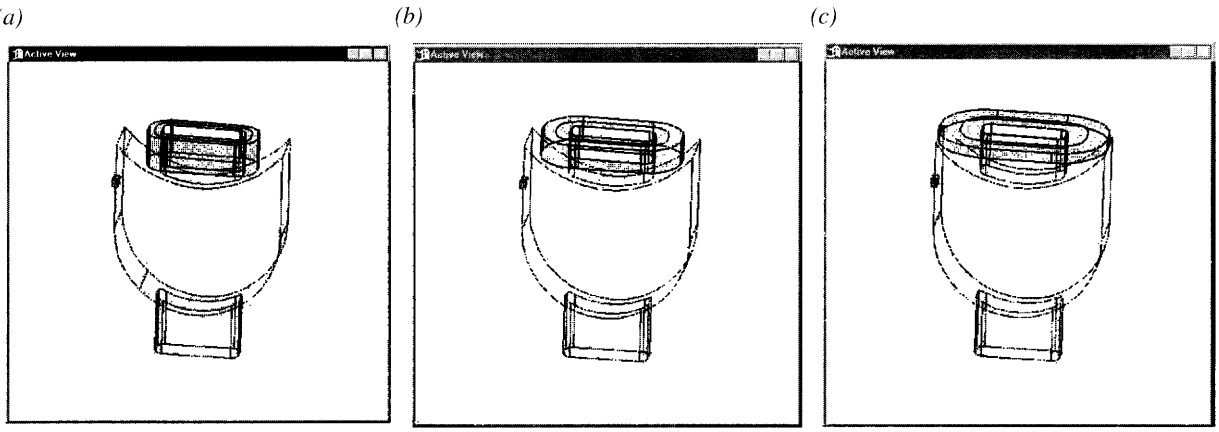

Fig. 11. Tool sizing for a non-2½-D feature.

6. Conclusions and Future Work

This work arose form two observations. First, that commercial CAM packages are unable to assign tools automatically or subdivide delta-volumes into manageable layers. Secondly, that data about pockets, and islands require manual input. The algorithms presented here automatically adjust a set of machining feature volumes to account for tooling and accessibility. A stand-alone tooling database is currently under development. This database will support future work on:

1. Alternative tool suggestion either automatically or interactively. Optimum tools can be decided by calculating the percentage of removed volume against original feature volume in addition to machinability feedback in (e.g. elapsed time needed for selected tool to clear delta-volume).
2. Automate tool-to-feature association catering for both single-tool and multiple-tool approaches given an allotted tool-change cycle (i.e. a fixed number of tool changes). In order to produce the required features, the program could scan through each machining level and also the set of cutting tools currently available on the carousel. The algorithm could then automatically associate the cutting tool/s for finishing in order to produce the complete feature. However, if current tools on the carousel are inapplicable, the algorithm attempts to suggest a more appropriate tool by searching the tooling database.

The significance of this research lies in the development of a methodology for tool sizing to assist tool optimisation and selection. It is envisaged that the incorporation of the algorithms presented in this paper will help advance automated functions within other computer integrated manufacturing (CIM) applications such as process planning.

Acknowledgements

The authors would like to acknowledge the support of EPSRC (grant no. GK/K48020), Pathtrace Engineering Systems and MacTaggart, Scott & Co. We would also like to express our gratitude to Gordon Little and Bob Tuttle for their suggestions and comments during the course of this research.

References

1. T. Dereli and I. H. Filiz, “Selection of alternative cutting tools for part features of prismatic components and their effects on operation sequence”, 5th International Conference on FACTORY 2000, IEE Conference Publications, 2–4 April, vol. 435, pp. 170–177, 1997.
2. Satyandra K. Gupta, “Automated manufacturability analysis of machined parts”, PhD Dissertation, The University of Maryland, College Park, MD, 1994.
3. W. Eversheim, M. Lenhart and B. Katzy, “Information modelling for technology-oriented tool selection”, Annals CIRP, 43(1), pp. 429–432, 1994.
4. M. Bala and T. C. Chang, “Automatic cutter selection and optimal path generation for prismatic parts”, International Journal of Production Research, 29(11), pp. 2163–2176, 1991.
5. K. Preiss and E. Kaplanski, “Automatic mill routing from solid geometry information”, in E. Warman (ed.), Computer Applications in Production and Engineering, North-Holland, pp. 773–784, 1983.
6. Y. S. Lee and D. Daftari, “Feature-composition approach to planning and machining of generic virtual pockets”, Computers in Industry, 31, pp. 99–128, 1996.
7. Y. S. Lee and T. C. Chang, “Using virtual boundaries for the planning of protrusion free-form features”, Computers in Industry, 25, pp. 173–187, 1994.
8. M. C. Wu and C. R. Liu, “Analysis on machined feature recognition techniques based on B-rep”, Computer-Aided Design, 28(8), pp. 603–616, 1995.
9. W. W. Charlesworth and D. C. Anderson, “Applications of non-manifold topology”, Proceedings of the Computers in Engineering Conference ASME, pp. 103–112, 1995.
10. M. Forsyth, “Shelling and offsetting bodies”, Proc. Third Symposium on Solid Modeling and Applications, Salt Lake City, Utah, 17–19 May, pp. 373–381, 1995.
11. Spatial Technology Inc., 2425 55th Street Building A, Boulder, CO 80301–5740. ACIS® 3D Toolkit API Reference, Version 4.0, 1997.
12. J. R. Rossignac, “Blending and offsetting solid models”, Tech. Mem. 54, PhD Dissertation, Production Automation Project, University of Rochester, Rochester, NY, 1985.
13. S. E. O. Saaed, A. de Pennington and J. R. Dodsworth, “Offsetting in geometric modelling”, Computer-Aided Design, 20, pp. 67–74, 1988.
14. R. T. Farouki and C. A. Neff, “Analytic properties of plane offset curves”, Computer-Aided Geometric Design, 7, pp. 83–99, 1990.
15. R. T. Farouki and C. A. Neff, “Analytic properties of plane offset curves”, Computer-Aided Geometric Design, 7, pp. 101–127, 1990.

16. G. Little, R. Tuttle, J. Corney and D. E. R. Clark, "The Heriot-Watt FeatureFinder: A graph-based approach to recognition", Proceedings of DETC97 1997 ASME Design Engineering Technical Conferences, pp. 1–9, 14–17 September 1997.
17. G. Little, D. E. R. Clark, R. Tuttle and J. Corney, "Delta-volume decomposition for multi-sided components", Journal of Computer-Aided Design, 30(9), pp. 695–705, 1998.
18. R. Tuttle, G. Little, D. E. R. Clark and J. Corney, "Feature recognition for NC part programming", International Conference on Manufacturing Automation, vol. 2, pp. 797–802, 1997.
19. N. Sormaz, D. E. R. Clark, J. Corney and J. R. Tuttle, "A feature recognition algorithm for NC-machining", Proceedings from the 5th IFIP Workshop, WG5.2, Workshop on Geometric Modeling in CAD, Airlee, Virginia, May 1996.

Appendix

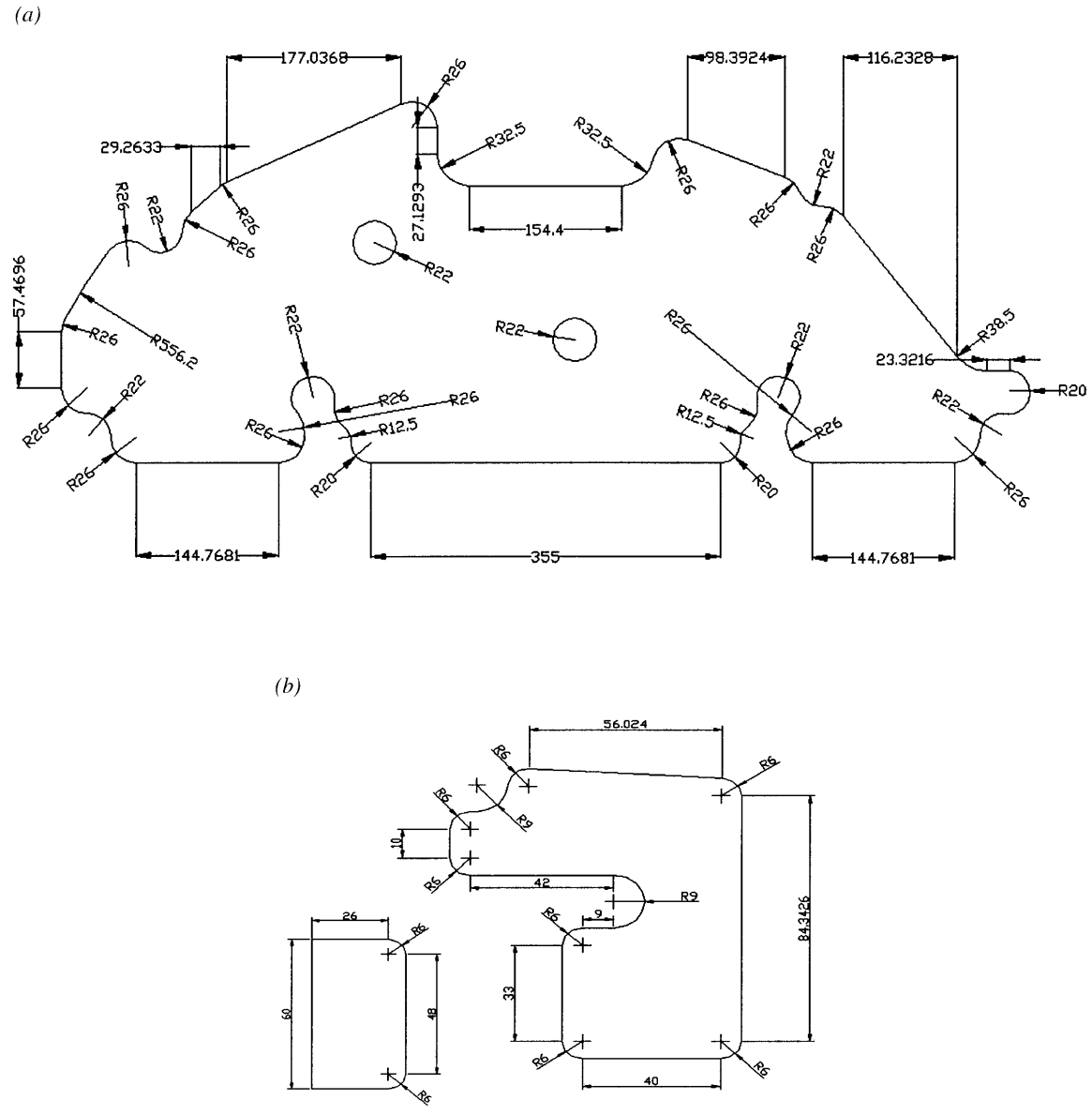

Fig. A1. Feature dimensions.