A Polyhedral-based Approach of Accessibility Computation for a NURBS Model in Tool-based Manufacturing

Kittichai Suthunyatanakit[1], Erik L. J. Bohez[2] and Kittinan Annanon[3]

[1]Asian Institute of Technology, skittichai@hotmail.com
[2]Asian Institute of Technology, bohez@ait.ac.th
[3]National Metal and Materials Technology Center, kittinau@mtec.or.th

ABSTRACT

Computing accessibility information from a NURBS model is an interesting aspect in design and manufacturing planning. We propose an approach how to compute the accessibility information from a NURBS model, called a *polyhedral-based approach*. In this paper, not only global point accessibility (usually used in the applications of CMM measuring and machining) but also global patch accessibility (i.e. a new term for mold design) is alternatively determined. At the high resolution, this approach runs faster than the approach for computing the facet accessibility.

Keywords: accessibility, visibility, manufacturability, mold design, machining.
DOI: 10.3722/cadaps.2010.139-151

1 MOTIVATION

In design, manufacturing, and especially manufacturing planning, accessibility information plays a significant role because it is used to decide whether a designed product is manufacturable or not. In addition, it is very helpful in process planning of the following tool-based applications.

- *Mold design*. In 2-piece mold design, the accessibility information is computed and analyzed for finding a pair of antipodal directions with the minimum number of undercuts [1-6]. In multi-piece mold design, the optimal number of parting directions without undercuts is determined from the accessibility information [7-9].
- *Machining*. In 3- and 4-axis milling, workpiece orientations can be optimally determined from the accessibility information by minimizing the number of setups [10-13]. In 5-axis milling, the accessibility information is adopted for finding tool orientations [14,15], tool selection [16], tool length [17], and toolpaths generation [18,19] by attempting in a single setup.
- *Measuring*. The accessibility information can help in determining part orientations on a CMM machine in order to approach a probe for measurement [20-26].
- *Other applications*. In the applications of scanning or reverse engineering [27], space partitioning [28], robot manipulation [29], etc., the accessibility information or its aspect can be used.

There have been various publications found in the literature. In the literature, however, we have been faced with two points of view. First, accessibility can be computed from a polyhedral model (i.e. a tessellated NURBS model), called *facet accessibility* in [30,31,32,33] and extended to be *region accessibility* in [34]. For this kind of computation, every facet of the model (or in the concave region) must be analyzed with a fixed resolution without mentioning a freeform NURBS model. Second, accessibility that is computed directly from a NURBS model is called *point accessibility* and this type of accessibility has been usually used in the applications of machining and measuring. To utilize it in the application of mold design, the point accessibility must be performed like the facet accessibility; that is, every point in the model must be analyzed. To relieve the gap of a polyhedral model and a NURBS model for accessibility computation, a polyhedral-based approach of accessibility computation for a NURBS model is proposed in this paper; to utilize the accessibility information in those applications of machining, measuring and mold design, two solutions, i.e. *point accessibility cones* and *patch accessibility cones*, can be alternatively determined in the approach.

The paper is organized into five sections. Section 2 reviews the literature on accessibility. Section 3 presents our methodology including the overview of our approach and the way how to compute accessibility from a NURBS model. Section 4 shows graphical results, computational complexity and evaluation. Finally, conclusions and future research directions are discussed in Section 5.

2 LITERATURE REVIEW

Authors	*Approach*	*Surface Type*	*Local/ Global*	*Acc. Type*	*Sol.*	*Appl.*
Spyridi & Requicha (1990) [35]	Gauss image + Minokski sum	Generic	Local& Global	Feature	LAC GAC	CMM measuring
Chen & Woo (1992) [36]	Gaussian map → Visibility map	Generic	Local	-	LAC	Machining
Kim et al. (1995) [37]	Tangent, normal & visibility map	Bezier	Local	Feature	LAC	General
Elber & Cohen (1995) [38]	Hidden surface removal technique	NURBS	Local	Point	-	Machining
Elber & Cohen (1998) [39]	Unified approach	NURBS	Local	Point	-	Machining
Kang & Suh (1997) [14]	Visibility to binary spherical map	NURBS	Global	Point	BSM	Machining
Yang et al. (1999) [40]	Inaccessibility based on control polygons	NURBS	Global	Point	GAC	Machining
Elber et al. (2004) [3]	Aspect graph	NURBS	Global	Surface	-	Mold design
Roberts & Rawat (2007) [41]	Partition-then-Evaluation	NURBS	Global	Point	BSM	Machining
Suthunyatanakit et al. This approach	Polyhedral-based approach	NURBS	Local& Global	Point Patch	LAC IAC GAC	General

Access.: Accessibility, Appl.: Application, BSM: Binary spherical map, GAC: Global accessibility cone, IAC: Inaccessibility cone, LAC: Local accessibility cone, Sol.: Solution.

Tab. 1: Summary of literature review on surface-based accessibility approaches.

Many approaches of accessibility computation have been proposed in tool-based manufac-turing applications. Balasubramaniam et al. [18] measured the accessibility information by using a graphics hardware technique for finally generating toolpaths in 5-axis machining. Bernhard and Veron [25] applied visibility theory to automatically find the orientation of a plane laser sensor in a digitizing process for a polyhedral part. The solutions of this approach are approximated, because represented on a tessellated unit sphere with a finite set of triangles. Dhaliwal et al. [30] proposed a geometric algorithm for computing global accessibility from a polyhedral model with triangles. A global accessibility cone (GACs) of this approach is represented as BSM, which is an approximate solution. To extend the scope of a polyhedral model, Li and Frank [31] proposed how to compute inaccessibility cones (IACs) for convex planar polygonal facets. Succeeding in fast computing IACs and computing exact GACs, Suthunyatanakit et al. [33] proposed a geometric algorithm for facet accessibility computation. To more extend the scope of a polyhedral model, Liu et al. [34] recently provided an algorithm using a Minkowski sums technique in order to compute global accessibility cones for regions on the boundaries of a polyhedral model. Notice that, all of these approaches are to compute the accessibility information from a polyhedral model.

However, there have been many approaches to compute accessibility from a surface model. Spyridi and Requicha [35], who firstly defined what local accessibility and global accessibility are, proposed a general method how to compute the local accessibility cone (LAC) using a Gauss image and the GAC using a Minkowski sum from a generic surface in CMM measuring. Like the LAC but in machining, Chen and Woo [36] computed a visibility map (VMap) by using the duality of a Gaussian map (GMap). To extend the scope of a surface, Kim et al. [37] developed this approach able to compute the LAC from a Bezier surface. Attempting to compute accessibility information from a NURBS model, Elber and Cohen [38,39] proposed a unified approach for 5-axis milling, as well as Elber et al. [3] proposed an aspect graph used in the application of two-piece mold design. In order to reduce the complexity of computing global accessibility from a NURBS surface, neighboring surfaces were approximately defined; for example, the control polygons of the surfaces [42], the convex hull of the surfaces [40], etc. have been used. Likewise, Kang and Suh [14] decomposed the part surface into triangular patches and computed a point visibility cone for 5-axis machining. Thus far, using a NURBS model has still been a significant aspect of accessibility computation. Recently, Roberts and Rawat [41] proposed a conservative approach, extending from the work of Kang and Suh, to compute global accessibility from a NURBS model by applying a partition-then-evaluation method in the process. However, the solutions of both approaches are approximated; i.e. a binary spherical map (BSM) for global point accessibility.

Notice that, most of the surface-based accessibility approaches in the literature are based on point accessibility information, which is normally utilized in the applications of CMM measuring and machining. In Table 1, the surface-based accessibility approaches including our approach are summarized.

3 METHODOLOGY

3.1 Overview of Polyhedral-based Approach

In this paper, our proposed approach is based on the approach of computing accessibility for a polyhedral model with convex polygonal facets in Ref.[33]. The proposed approach has four main, excluding the preliminary partitioning (see Fig.1):

Input: A set of NURBS surface patch $\{s_1, s_2, ..., s_S\}$, where S is the total number of surface patches. Additional input is point data $\{p_1^1, p_2^1, ..., p_1^2, ..., p_j^i, ...\}$, e.g. sampling points on the surface patch for computing global point accessibility, where i is the index of a surface patch s_i and j is the index of a point on the patch.

Output: Either global point accessibility cones $\{GpAC(p_1^1), GpAC(p_2^1), ..., GpAC(p_j^i), ...\}$ or global patch accessibility cones $\{GPAC(s_1), GPAC(s_2), ..., GPAC(s_S)\}$.

Input restriction: the geometric model is a NURBS model which is water-tight without internal shell and self intersection. Preliminary partitioning is done in order that (1) each surface patch has single curvature (i.e. planar, convex, concave, or saddle), (2) no crease curves exist on the surface, (3) radius of curvature in the iso-parametric curves (u and v) on the NURBS is not more than π, and (4) No inner trimming curves exist.

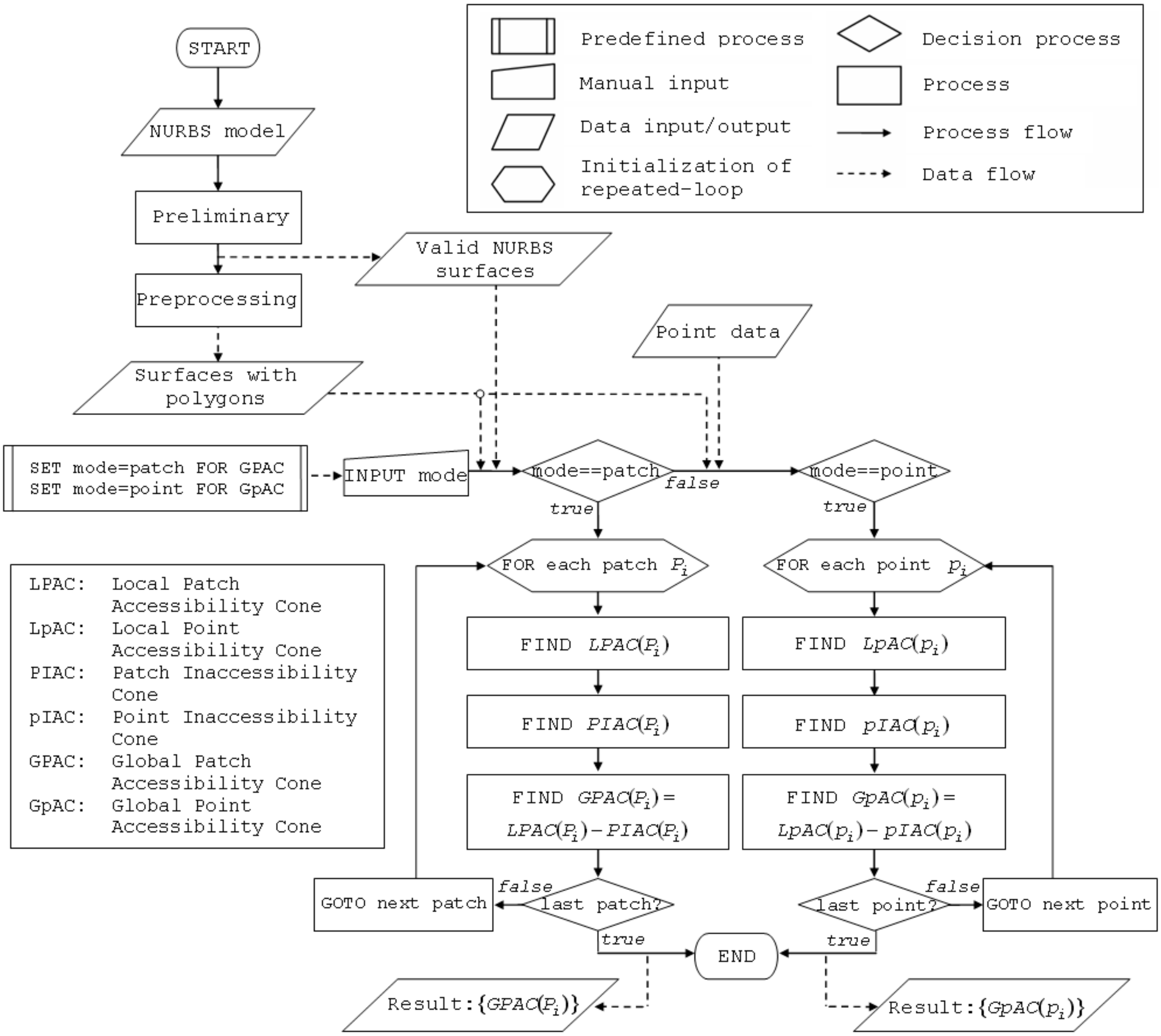

Fig. 1: Flowchart of point accessibility and patch accessibility computation for NURBS surfaces.

Procedure:

(1) *Preprocessing process.* This step is done as the preparation of the geometric model for accessibility computation. Because of extending the polyhedral-based accessibility computation, each NURBS surface patch is tessellated as geometric polygons; however, the mesh of the polygons can be tessellated adaptively depending on the resolution of the surface. In this paper, we use the simple tessellation for each surface patch.

(2) *Local accessibility determination.* This step does not happen in the accessibility computation for a polyhedral model because the facets of the model are planar. For a NURBS model, each surface patch may not be planar and its local accessibility cone (LAC) is not always the hemisphere. Therefore, the local accessibility for the non-planar patch must be determined. Section 3.2 describes how to determine both the local point accessibility cone (LpAC) and the local patch accessibility cone (LPAC).

(3) *Global inaccessibility determination.* In this step, either a point inaccessibility cone (pIAC) for each point on a surface or a patch inaccessibility cone (PIAC) for each patch is determined. The determinations of the pIAC and the PIAC are developed by extending the polyhedral-based accessibility computation. They are described in Section 3.3.

(4) *Global accessibility determination.* Finally, the global point accessibility cone (GpAC) and the global patch accessibility cone (GPAC) can be determined in the same way of the polyhedral-based accessibility computation; i.e. the complement of the inaccessibility cone are determined. Unlike a polygonal facet, the GpAC and the GPAC can be calculated by: *GpAC* = *LpAC* – *pIAC* and *GPAC* = *LPAC* – *PIAC*, respectively, because *LpAC* and *LPAC* is not always the hemisphere. Section 3.4 discusses this step.

3.2 Local Accessibility Determination

For local point accessibility. The point on a surface is said "*local point accessibility*" when the point is accessible with the directions in which the light rays do not intersect with the surface occupying the point. Given a surface s and a point p on s, the local point accessibility cone of p (denoted as $LpAC(p)$) is determined, as follows.

If s is either a planar surface or a convex surface, $LpAC(p)$ is the hemisphere the pole of which is the endpoint of the unit normal at p. In this case, Fig.2 shows the results.

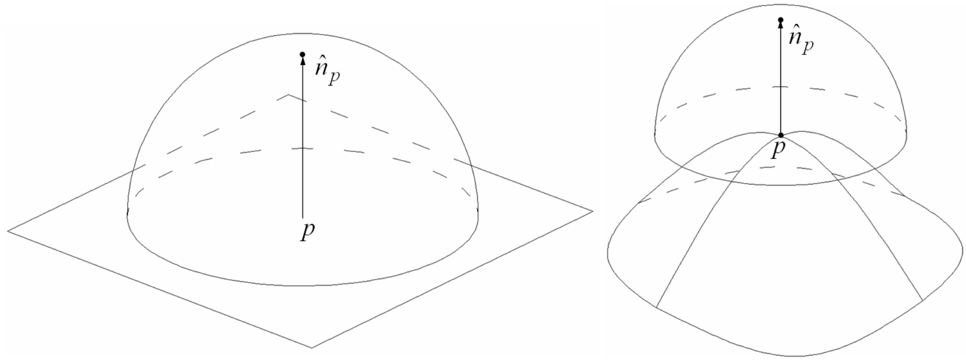

Fig. 2: LAC of point on planar and convex surface as hemisphere.

For a concave surface, certain light rays at p intersect with s. Hence, the local point accessibility cone (LpAC) for the concave surface is never the hemisphere. $LpAC(p)$ can be determined by finding the rays from p to every point on the boundary of s and then constructing the LAC. That is, the cone is constructed from the boundary, called a *bounding cone*. Fig.3 shows how to find the LpAC for the concave surface.

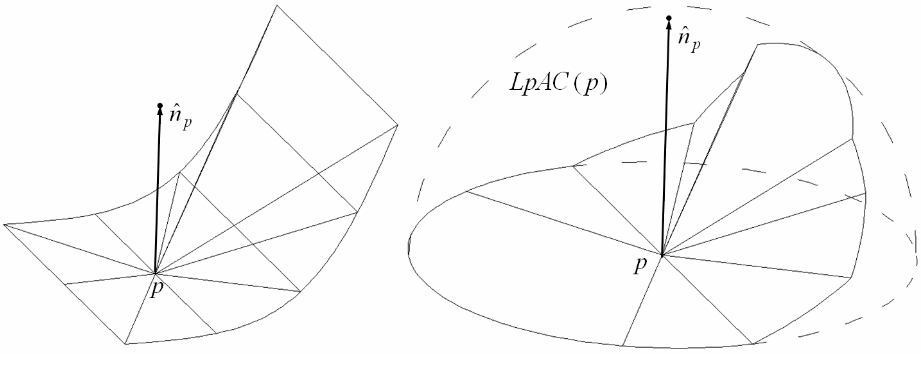

Fig. 3: LAC of point on concave surfaces as bounding cone.

Likewise, the LpAC for the saddle surface is never the hemisphere. Unlike the concave surface, there are some light rays traverse below *s*. Determining the LpAC as only the bounding cone is not possible. However, it can be simply done by: $LpAC(p) = H \cap (bounding\ cone)$, where H is the hemisphere of which the pole is the endpoint of the unit normal at p.

For local patch accessibility. A surface patch is said "*local patch accessibility*" when all the points on the surface patch are entirely accessible with the directions in which the light rays do not intersect with the interior of the surface patch. Likewise, this is done excluding with other neighbor surface patches. The result of local patch accessibility is called local patch accessibility cone (denoted as LPAC). Notice that we have more chance to see (i.e. visibility) a point on the surface patch than the entire surface patch; that is, $LPAC \subseteq LpAC$. Given a surface patch s, its LPAC can be written as: $LPAC(s) = \bigcap LpAC(p_i), \quad for\ \forall p_i \in s$. To reduce the difficulty of intersection, we propose the method for computing the LPAC, as follows:

(1) Find the tangent vectors (denoted as $\vec{T}$) at the endpoints of each iso-parametric curve in u and v.
(2) For each iso-parametric curve in u and v, determine the local accessibility cone (denoted as $LPAC^u$ and $LPAC^v$) by the tangent vectors.
(3) Construct an LPAC of S by $LPAC(s) = LPAC^u \otimes LPAC^v$, where $\otimes$ is a 3D LAC operator.

To understand how to compute the LPAC, the more details of each step are given below.

Firstly, finding the tangent vectors at the endpoints of each iso-curve is done. Due to the valid surface, there are only three types of the iso-parametric curve; i.e. linear, convex, and concave. Because the linear curve is simple, the tangent vectors are calculated for only the convex curve and the concave curve. For each iso-parametric curve in u or in v, the tangent vector at the starting point (i.e. $u=0$ or $v=0$) is denoted as $\vec{T}_s$ and the tangent vector at the endpoints (i.e. $u=1$ or $v=1$) as $\vec{T}_e$.

Secondly, the $LPAC^u$ and $LPAC^v$ are determined. This is done in 2D. Each of the $LPAC^u$ and $LPAC^v$ is represented as the LAC of the curve and it is (see Fig.4):

- *For a linear curve*, $LAC = 180°$.
- *For a convex curve*, $LAC = \angle(-\vec{T}_s, \vec{T}_e)$.
- *For a concave curve*, $LAC = \angle(\vec{T}_s, -\vec{T}_e)$.

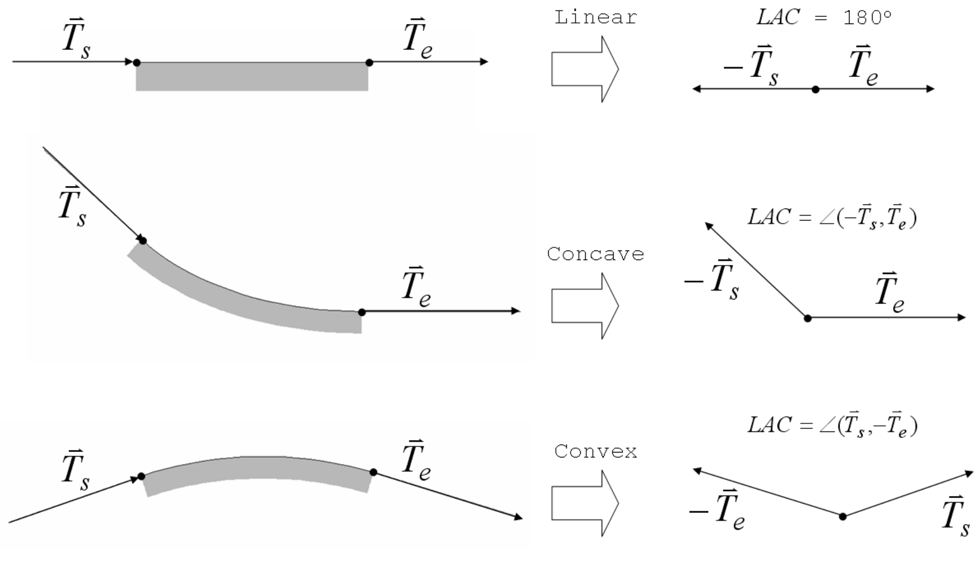

Fig. 4: Representation of $LPAC^u$ and $LPAC^v$.

Finally, the local patch accessibility cone (LPAC) is readily constructed from $LPAC^u \otimes LPAC^v$. The operator $\otimes$ means that the $LPAC^u$ and $LPAC^v$ with the minimum angle are selected and the LPAC is then constructed directly from them. Fig.5 shows how to construct the LPAC of a concave ruled-surface.

3.3 Global Inaccessibility Determination

To find the inaccessibility cone (IAC) for either a point or a surface patch is simply done. The IAC determination for a polygonal facet in Ref.[33] is applied without finding the convex region on the hemisphere because in this case are the point inaccessibility considered.

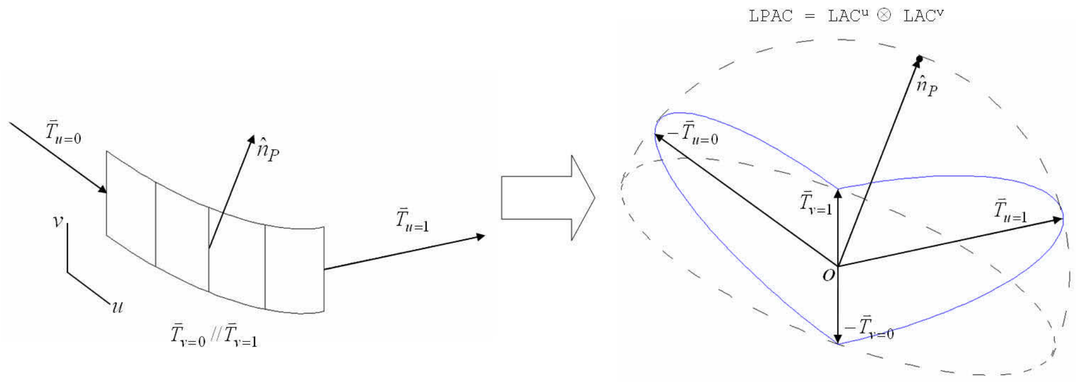

Fig. 5: LPAC of concave ruled-surface.

For point inaccessibility. The inaccessibility cone of the point on a surface is simply determined. We consider the point on the surface with the polygons on the neighbors of the surface. Given a point p on a surface s and a polygon G on other surface s', the point inaccessibility cone (denoted as $pIAC(p)$) can be determined, as follows (see Fig.6):

(1) Finding the inaccessibility rays from p to the vertices of G,

(2) Constructing the pIAC directly from all the inaccessibility rays.

(3) Finally, the total point inaccessibility cone is determined by: $pIAC(p) = \bigcup_{i,j} pIAC(p, G_{i,j})$, where $p \text{ on } s \text{ and } \forall G_{i,j} \text{ on } s'_j$

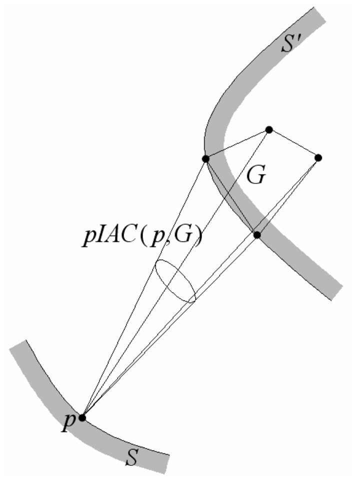

Fig. 6: Point inaccessibility cone (pIAC).

For patch inaccessibility. Likewise, the inaccessibility of a patch can be simply determined. Although each surface patch composes of a number of polygons, determining the inaccessibility from a pair of polygons is unnecessary. In the case of the patch inaccessibility, the determination of point inaccessibility is applied only at the points on the boundary of the surface patch (p *on* $boundary(s)$). Given a surface patch s, and a polygon G on other surface s', The determination of patch inaccessibility cone (PIAC) is as follows:

(1) Finding $pIAC(p_1,G)$, where p_1 is on $boundary(s)$ and G is on s'. See Fig.7(a).
(2) Traversing $pIAC(p_1,G)$ to $pIAC(p_2,G)$ and then constructing $pIAC(\overline{p_1p_2},G)$, where p_2 is next to p_1 and p_2 is also on $boundary(s)$. See Fig.7(b).
(3) Repeatedly traversing $pIAC(p,G)$ along $boundary(s)$ and getting $pIAC(boundary(s),G)$. See Fig.7(c).
(4) Finally, the patch inaccessibility cone is determined by (Fig.7(d)) : $PIAC(s,G) = unit\ sphere \cap pIAC(boundary(s),G)$

That is, the total patch inaccessibility cone can be written as: $PIAC(s) = \bigcup_{i,j} PIAC(s,G_{i,j})\quad for\ \forall G_{i,j}\ on\ s'_j$.

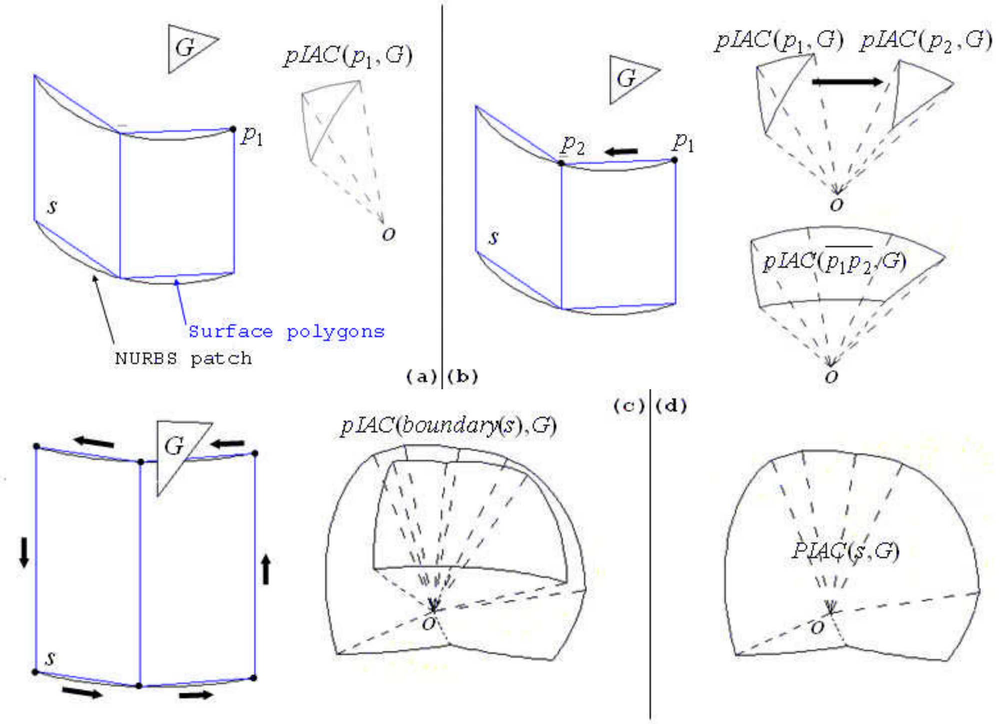

Fig. 7: Patch inaccessibility cone (PIAC) due to polygon.

3.4 Global Accessibility Determination

Thus far, the global point accessibility cone (GpAC) and global patch accessibility cone (GPAC) are readily determined. In Ref.[33], they can be determined by the compliment of the union of inaccessibility cones; in order to reduce the computational time, however, the near-exact computation of the polyhedral-based accessibility computation can be applied directly to compute the near-exact GpAC and GPAC as well.

4 RESULTS, COMPUTATIONAL COMPLEXITY AND EVALUATION

For illustrating the overall of the approach, an example model in Fig.8(a) is given to compute its global point accessibility at a point p and its global patch accessibility of a patch S, the neighboring patches of which are $s_1', s_2', s_3', \dots$. Fig.8(b) and Fig.8(c) shows the geometric results of global accessibility cones of p and s, respectively.

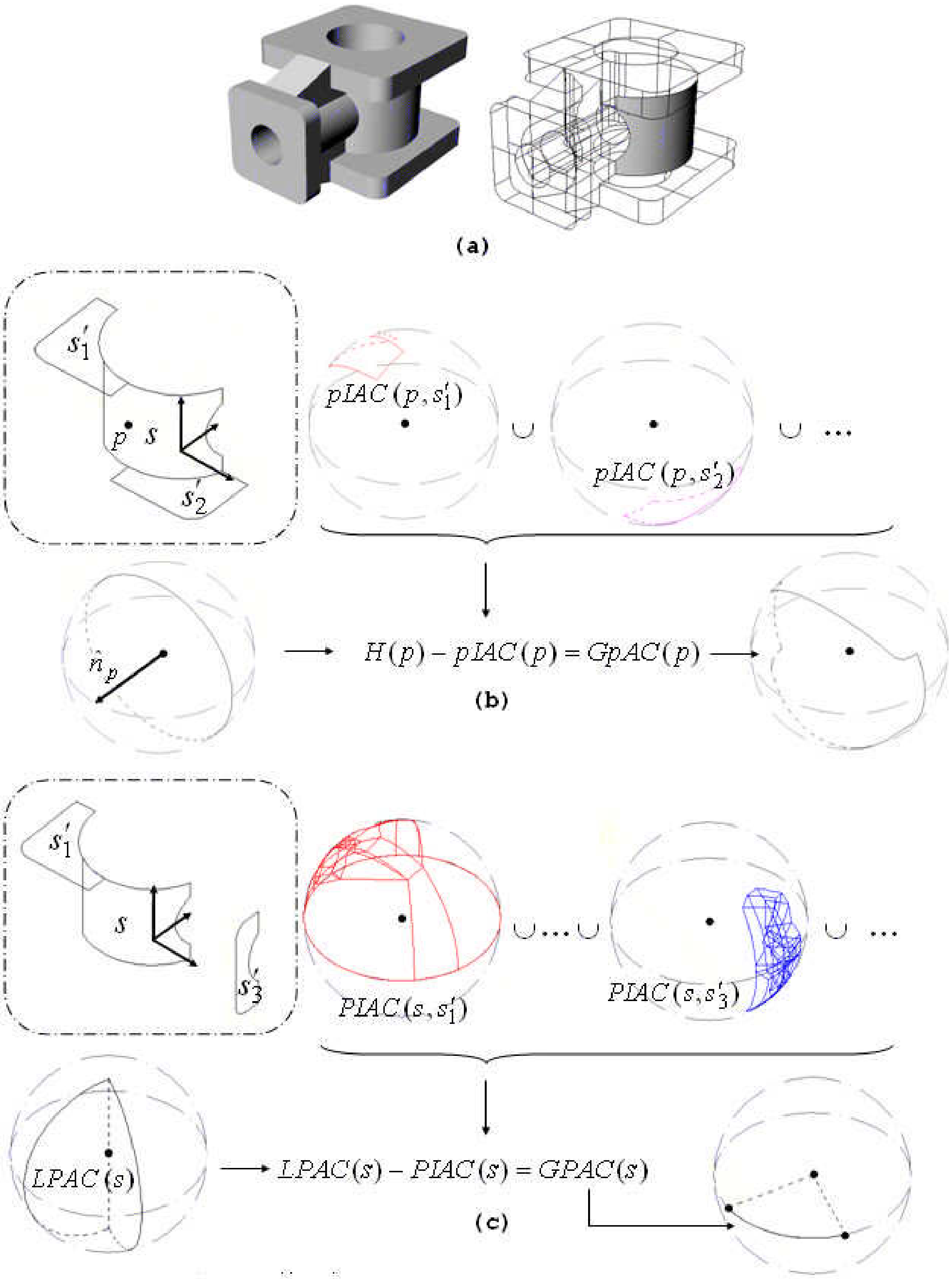

Fig. 8: Example model with geometric results: (a) Model, (b) Global point accessibility, and (c) Global patch accessibility.

4.1 Computational Complexity

The computational complexity of accessibility computation for a NURBS model is summarized in Table 2. Given a NURBS model having N_s surface patches and N corresponding surface polygons and each surface patch s having the resolution of u and v, the computational complexity of accessibility computation is as follows:

- *Local accessibility cone (LAC).* For a NURBS model, each s requires $O(u+v)$ because we the LAC^u and LAC^v in a linear fashion, and the LAC for the entire model thus requires $O((u+v)N_s)$.
- *Inaccessibility cone (IAC).* To determine point inaccessibility, we compare each point with N surface polygons, and thus requires $O(N)$. If we sample points by the resolution of u and v, there are totally uvN_s points for the entire model and it therefore requires $O(uvN_sN)$. For determining patch inaccessibility, likewise, we compare each point with N surface polygons, but only the points on the boundary of the surface patch. There are totally $(u+v)N_s$ points and therefore $O((u+v)N_sN)$ is required.
- *Global accessibility cone (GAC).* To determine global accessibility, we need operate the union of IACs and it requires $O(N^2)$. For the entire model, hence, $O(uvN_sN^2)$ and $O((u+v)N_sN^2)$ are required for global point accessibility determination and global patch accessibility determination, respectively.

Procedure	***Computational Complexity***		
	NURBS		*Polyhedral*
	Point	*Point*	*Facet*
LAC determination	$O((u+v)N_S)$	$O((u+v)N_S)$	-
IAC determination	$O(uvN_SN)$	$O((u+v)N_SN)$	$O(N_f^2)$
GAC determination	$O(uvN_SN^2)$	$O((u+v)N_SN^2)$	$O(N_f^3)$

u, v : Resolution of NURBS (integer), N : Number of surface polygons,
N_S : Number of surface patches, N_f : Number of polygonal facets,

Tab. 2: Computational complexity of accessibility computation.

4.2 Computation Evaluation

In Table 2, it compares with the computational complexity of the polyhedral-base approach as well. By comparing with the computational complexity of the polyhedral-based accessibility, the accessibility computation for a NURBS model has the following advantages.

- Considering a polyhedral model, we do not need to compute its LAC because the LAC is a hemisphere automatically for every planar facet; for the region on the polyhedral model, however, computing the LAC needs the intersection of the hemispheres occupied by each point in the region. The Boolean intersection consumes much computational time. In this approach, we can compute LACs without the Boolean intersection.
- The computational complexity for point inaccessibility is approximately the same for facet inaccessibility. Assume that a NURBS model is tessellated with the very high resolution. $uvN_s \approx N$ and $N \approx N_f$, and therefore $O(uvN_sN) \approx O(N_f^2)$. Using the NURBS model, however, we can do:
    - o Adaptively tessellate the model and
    - o Compute $pIAC(p,G)$, where G is *not* the surface polygon in the surface patch occupying p
- For region inaccessibility, the computational complexity for patch inaccessibility is less than for facet inaccessibility, because $O((u+v)N_sN) < O(N_f^2)$ at the high resolution of NURBS.

5 CONCLUSION AND FUTURE RESEARCH

An approach for computing the global accessibility from a NURBS model has been proposed. It has been succeeding by developing the polyhedral-based approach able to compute the accessibility from a NURBS model (i.e. in an adaptive resolution). In this approach, global point accessibility for usually used in the applications of CMM measuring and machining as well as global patch accessibility for mold design can be alternatively determined. At the high resolution of NURBS, both accessibility computations are faster than the computation of the facet accessibility.

In this paper, however, the computational complexity of an exact global accessibility is approximately $O(N^3)$. This computational time is expensive because of the union operation. Every kind of accessibility computation (i.e. point, patch and facet) needs the union operation. In the future research, we will thus develop the algorithm of operating the union with fast computation.

REFERENCES

[1] Chen, L. L.; Chou, S. Y.; Woo, T. C.: Parting directions for mould and die design, Computer-Aided Design, 25 (12), 1993, 762-768.

[2] Yin, Z. P.; Ding, H.; Xiong, Y. L.: Virtual prototyping of mould design: geometric moldability analysis for near-net-shape manufactured parts by feature recognition and geometric reasoning, Computer-Aided Design, 33, 2001, 137-154.

[3] Elber, G.; Chen, X.; Cohen, E.: Mold accessibility via Gauss map analysis, In: Proceedings of the Shape Modeling International, IEEE Computer Society, 2004, 263-272.

[4] Yin, Z. P.; Ding, H.; Li, H. X.; Xiong, Y. L.: Geometric moldability analysis by geometric reasoning and fuzzy decision making, Computer-Aided Design, 36, 2004, 37-50.

[5] Fuh, J. Y. H.; Zhang, Y. F.; Nee, A. Y. C.; Fu, M. W.: Computer-aided injection mold design and manufacture, Marcel Dekker, 2004.

[6] Khardekar, R.; Burton, G.; McMains, S.: Finding feasible mold parting directions using graphics hardware, Computer-Aided Design, 38, 2006, 327-341.

[7] Chen, Y.; Rosen, D. W.: A region based method to automated design of multipiece molds with application to rapid tooling, ASME Journal of Computing and Information Science in Engineering, 2(2), 2002, 86-97.

[8] Huang, J.; Gupta, S. K.; Stopel, K.: Generating sacrificial multi-piece molds using accessibility driven spatial partitioning, Computer-Aided Design, 35, 2003, 1147-1160.

[9] Priyadarshi, A. K.; Gupta, S. K.: Geometric algorithms for automated design of multi-piece permanent molds, Computer-Aided Design, 36, 2004, 241-260.

[10] Tseng, Y. J.; Joshi, S.: Determining feasible tool-approach directions for machining Bézier curves and surfaces, Computer-Aided Design, 23 (5), 1991, 367-379.

[11] Chen, L. L.; Chou, S. Y.; Woo, T. C.: Separating and intersecting spherical polygons: computing machinability on three-, four-, and five-axis numerically controlled machines, ACM Transactions on Graphics, 12(4), 1993, 305-326.

[12] Gupta, P.; Janardan, R.; Majhi, J.; Woo, T. C.: Efficient geometric algorithms for workpiece orientation in 4- and 5-axis NC machining, Computer-Aided Design, 28(8), 1996, 577-587.

[13] Quinsat, Y.; Sabourin, L.: Optimal selection of machining direction for the three-axis milling of sculptured parts, International Journal of Advanced Manufacturing Technology, 27, 2006, 1132-1139.

[14] Kang, J. K.; Suh, S. H.: Machinability and set-up orientation for five-axis numerically controlled machining of free surfaces, International Journal of Advanced Manufacturing Technology, 13, 1997, 311-325.

[15] Wang, J.; Robert, C. A.; Danielson, S.: Local and global accessibility evaluation with tool geometry, Computer-Aided Design and Applications, 4, 2007, 19-29.

[16] Jensen, C. G.; Red, W. E.; Pi, J.: Tool selection for five-axis curvature-matched machining, Computer-Aided Design, 34, 2002, 251-266.

[17] Bi, Q.; Ding, H.; Wang, Y.: Safe and short tool length generation for 3+2 axis NC machining of a ball-end cutter using graphics hardware. In: Proceedings of the First International Conference on Intelligent Robotics and Applications: Part II, Springer-Verlag, 2008, 348-355.

[18] Balasubramaniam, M.; Laxmiprasad, P.; Sarma, S.; Shaikh, Z.: Generating 5-axis NC roughing paths directly from a tessellated representation, Computer-Aided Design, 32, 2000, 261-277.
[19] Balasubramaniam, M.; Sarma, S. E.; Marciniak, K.: Collision-free finishing toolpaths from visibility data, Computer-Aided Design, 35, 2003, 359-374.
[20] Limaiem A.; ElMaraghy H.E.: A general method for analyzing the accessibility of features using concentric spherical shells. International Journal of Advanced Manufacturing Technology, 13, 1997, 101-108.
[21] Ziemian, C.W.; Medeiros, D.J.: Automated feature accessibility algorithm for inspection on a coordinate measuring machine. International Journal of Production Research, 35 (10), 1997, 2839-2856.
[22] Jackman, J.; Park, D. K.: Probe orientation for coordinate measuring machine systems using design models – theory, methods, and systems, Robotics and Computer-Integrated Manufacturing, 14(3), 1998, 229-236.
[23] Kweon, S.; Medeiros, D. J.: Part orientations for CMM inspection using dimensioned visibility maps, Computer-Aided Design, 30(9), 1998, 741-749.
[24] Spitz, S. N.; Spyridi, A. J.; Requicha, A. A. G.: Accessibility analysis for planning of dimensional inspection with coordinate measuring machines, IEEE Transactions on Robotics and Automation, 15(4), 1999, 714-727.
[25] Bernhard, A.; Veron, M.: Visibility theory applied to automatic control of 3D complex parts using plane laser sensors, Annals of the CIRP, 49(1), 2000, 113-118.
[26] Wu, Y.; Liu, S.; Zhang, G.: Improvement of coordinate measuring machine probing accessibility, Precision Engineering, 28, 2004, 89-94.
[27] Elber, G.; Zussman, E.: Cone visibility decomposition of freeform surfaces, Computer-Aided Design, 30(4), 1998, 315-320.
[28] Fernández, P.; Rico, J. C.; Álvarez, B. J.; Valiño, G.; Mateos, S.: Laser scan planning based on visibility analysis and space partitioning techniques, International Journal of Advanced Manufacturing Technology, 39, 2008, 699-715.
[29] Jang, H. Y.; Moradi, H.; Minh, P. L.; Lee, S.; Han, J. H.: Visibility-based spatial reasoning for object manipulation in clustered environments, Computer-Aided Design, 40, 2008, 422-438.
[30] Dhaliwal, S.; Gupta, S. K.; Huang, J.; Priyadarshi, A.: Algorithms for computing global accessibility cones, ASME Journal of Computing and Information Science in Engineering, 3(3), 2003, 20-209.
[31] Li, Y.; Frank, M. C.: Computing non-visibility of convex polygonal facets on the surface of a polyhedral CAD model, Computer-Aided Design, 39, 2007, 732-744.
[32] Liu, M.; Ramani, K.: Computing an exact spherical visibility map for meshed polyhedra. In: Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling, 2007, 367-372.
[33] Suthunyatanakit, K.; Bohez, E. L. J.; Annanon, K.: A new global accessibility algorithm for a polyhedral model with convex polygonal facets, Computer-Aided Design, 41, 2009, 1020-1033.
[34] Liu, M.; Liu, Y. S.; Ramani, K.: Computing global visibility maps for regions on the boundaries of polyhedra using Minkowski sums, Computer-Aided Design, 41, 2009, 668-680.
[35] Spyridi, A. J.; Requicha, A. A. G.: Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines, In: Proceedings of the IEEE international conference on Robotics and Automation, Cincinnati, OH, 1990, 1284-1289.
[36] Chen, L. L.; Woo, T. C.: Computational geometry on the sphere with application to automated machining, ASME Journal of Mechanical Design, 114, 1992, 288-295.
[37] Kim, D. S.; Papalambros, P. Y.; Woo, T. C.: Tangent, normal, and visibility cones on Bezier surfaces, Computer-Aided Geometric Design, 12, 1995, 305-320.
[38] Elber, G.; Cohen, E.: Arbitrarily precise computation of Gauss maps and visibility sets for freeform surface. In: The third ACM/IEEE Symposium on Solid Modeling and Applications, Salt Lake City, Utah, 1995, 271-279.
[39] Elber, G.; Cohen, E.: A unified approach to accessibility in 5-axis freeform milling environments, In: Proceedings of the IFIP TC5 WG5.3 International Conference on Sculptured Surface Machining, Kluwer, B.V., 1998, 376-386.
[40] Yang, W.; Ding, W.; Xiong, Y.: Manufacturability analysis for a sculptured surface using visibility cone computation, International Journal of Advanced Manufacturing Technology, 15, 1999, 317-321.

[41] Robert, C. A.; Rawat R.: An approach for tool accessibility evaluation for multi-axis machining models, Computer-Aided Design and Applications, 4, 2007, 385-394.

[42] Lee, Y. S.; Chang, T. C.: 2-phase approach to global tool interference avoidance in 5-axis machining, Computer-Aided Design, 27 (10), 1997, 715-729.