



2D Accessibility Analysis for Waterjet Cleaning

Wei Li¹, Saurabh Garg² and Sara McMains³

¹University of California, Berkeley, liw@berkeley.edu

²University of California, Berkeley, sgarg@berkeley.edu

³University of California, Berkeley, mcmains@me.berkeley.edu

ABSTRACT

Effective cleaning with high pressure waterjets requires direct impact of jets and sufficiently high impact pressure. The objective of this research is to find all such cleanable regions, given a CAD model of a workpiece, by means of geometric accessibility analysis. We use a configuration space (C-space) approach for addressing the problems of both optimum surface proximity for effective cleaning and collision avoidance between the cleaning lance and the workpiece. Minkowski sums are used to compute the C-spaces and cleanable regions are then found by visibility analysis. Implementations and results for 2D examples are shown to validate the approach.

Keywords: waterjet cleaning, accessibility, configuration space, Minkowski sums.

DOI: 10.3722/cadaps.2009.231-241

1. INTRODUCTION

Cleaning engine components to remove hard particle contaminants introduced during the manufacturing process is becoming a significant issue for industry [2][3]. Examples of mechanical components that are particularly prone to solid particle contamination are engine blocks and cylinder heads fabricated by sand casting wherein the sand particles from the mold get embedded onto the surface of the cast product. A commonly used cleaning method is High Pressure Jets (HPJs), where the cleaning action is obtained from the impact of a high speed jet of water. Experience in the automotive industry has shown that efficacy of HPJ cleaning is strongly dependent on the geometry of the workpiece [1].

Using HPJ cleaning for the removal of embedded particles such as casting sand, it has been found that direct impact of the jets is often necessary for their removal. Hence, the HPJ lance - a rigid tube with a cleaning nozzle on the end - must be able to access the features such that the entire surface area receives a direct impact from the jets. Another necessary condition is sufficiently high impact pressure on the workpiece surface when it is hit by the waterjets from the nozzle. If the distance between the nozzle and the surface becomes large enough, the impact pressure drops below the minimum requirement for effective cleaning. In this paper we take into consideration both the availability of direct impact and the magnitude of impact pressure. We say a region of the surface is *cleanable* if waterjets can directly access it and the impact pressure is high enough to perform effective cleaning. Otherwise we say it is *non-cleanable*. Fig. 1 illustrates two types of non-cleanable regions.

In this paper we present an approach for finding all possible cleanable regions of a workpiece by means of geometric accessibility analysis. We consider a simplified 2D problem where the workpiece is represented as a polygon. We also assume that the workpiece position and orientation is fixed and

that the lance can only translate (not rotate) relative to the workpiece during the cleaning process. We only consider the collision between the lance and workpiece and do not consider the lance holder and other obstacles in our implementation, although our algorithm could be trivially extended to incorporate these inputs as well. An example result of our algorithm is shown in Fig. 2. Although the algorithm is demonstrated in 2D, it can be extended to 3D as well.

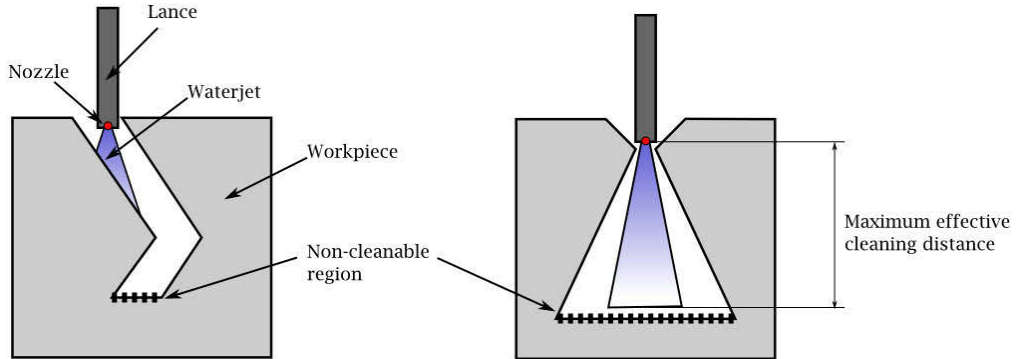


Fig. 1: Two types of non-cleanable regions (dotted). On the left the surface is not directly accessible by the waterjet. On the right the distance between the nozzle and the surface is too far to provide enough impact pressure for effective cleaning.

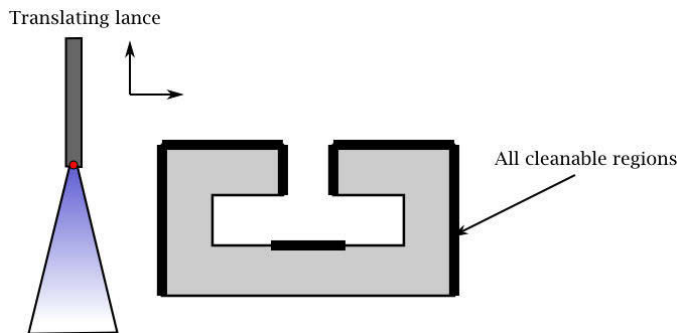


Fig. 2: An example of all possible cleanable regions (thickened). The workpiece is fixed and the lance can only translate.

Our algorithm consists of two major stages. In the first stage, the entire C-space is divided into three disjoint C-spaces - cleaning C-space, non-cleaning C-space, and interference C-space - by using Minkowski sums (section 4). Then in the second stage, we find all the regions of the workpiece that are cleanable when the lance is positioned in the cleaning C-space (section 5), based on visibility and effective cleaning distance.

2. RELATED WORK

Configuration spaces (C-spaces) have been heavily used for collision avoidance in automatic path planning. Lozano-Pérez first introduced the C-space approach for motion planning of a rigid object among physical obstacles [10]. Every point in the C-space corresponds to a set of independent parameters that characterize the position and orientation of the rigid object. Choi et al. applied the C-space approach to 3-axis NC tool path generation for sculptured surface machining [5]. Tool paths generated from the C-spaces were gouge-free and collision-free.

C-spaces are usually computed by using Minkowski sums. Given two sets P and Q , their *Minkowski sum*, denoted by $P \oplus Q$, is the set $\{p + q | p \in P, q \in Q\}$. For 2D polygons, various algorithms for

computing their Minkowski sums are implemented in CGAL, the Computational Geometry Algorithms Library [4]. For 3D convex polyhedra, their exact Minkowski sums can be constructed relatively efficiently using either a convex hull or a Gaussian map approach [6]. For two concave polyhedra with m and n vertices respectively, their Minkowski sum can have complexity as high as $O(m^3n^3)$. One commonly used approach decomposes the two concave polyhedra into convex pieces, computes their pairwise Minkowski sums, and finally the union of the pairwise Minkowski sums [12].

Accessibility analysis has been recognized as an important tool in various applications such as dimensional inspection, machining, assembly, and mold design. In machining, for example, accessibility analysis helps in determining machinability by finding the set of directions from which the part may be approached by the cutting tool. Spitz et al. presented a discrete approximation algorithm for computing accessibility information by exploiting computer graphics hardware [11]. Woo introduced the concept of spherical visibility maps and applied them to various manufacturing problems like optimal workpiece orientation [13].

3. WATERJET MODEL

Before we analyze the accessibility of waterjets, we need to compute the *effective water zone* – the water zone where the impact pressure is high enough for effectively cleaning. The impact pressure becomes low where the distance to the nozzle is large and where waterjets mix with the surrounding air. To compute the impact pressure inside the waterjets, we use a mathematical model introduced in [9]. Let the nozzle center be the origin O . The impact pressure P at some point inside the waterjets is a function of the standoff distance x and radial distance r , as depicted in Fig. 3(a).

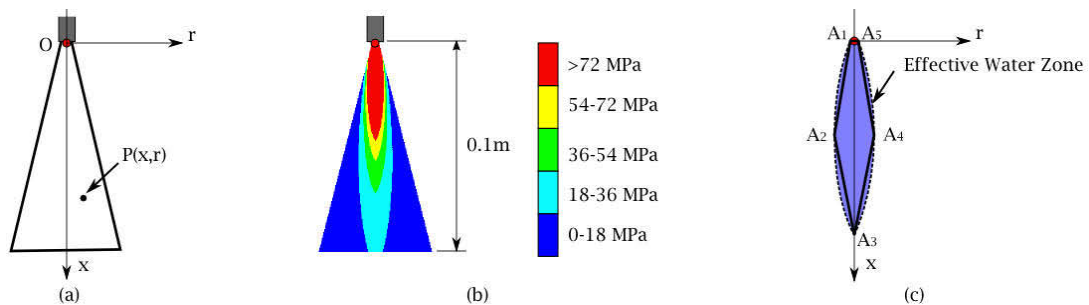


Fig. 3: (a) Impact pressure is a function of standoff distance x and radial distance r . (b) An example of impact pressure distribution inside the waterjets. (c) The effective water zone is approximated as a polygon $A_1A_2A_3A_4A_5$.

According to [9], the impact pressure can be expressed as:

$$P(x, r) = \left(\frac{D}{x}\right)^2 \left[1 - \left(\frac{r}{Cx}\right)^{1.5}\right]^3 \tag{3.1}$$

where C is the spreading coefficient of water, and D is a parameter determined by the water pressure from the pump, the nozzle radius, and some physical constants of water. Both C and D can be assumed as constants during the cleaning process. Based on this equation, we can compute the impact pressure distribution inside the waterjets. Fig. 3(b) illustrates such an example where the water pressure from the pump is 200MPa and the nozzle radius is 0.3mm. Suppose the minimum effective impact pressure is P_{min} , then the effective water zone is the set of all locations satisfying $P \geq P_{min}$. We can see from Fig. 3(b) that iso-pressure curves have an ellipse-like shape. For simplicity we approximate the effective water zone as a polygon $A_1A_2A_3A_4A_5$, as shown in Fig. 3(c). The short edge A_1A_5 represents the width of the nozzle. Since usually the nozzle radius r_0 is very small and the impact

pressure near the nozzle is very high, we assume the effective water zone emanates from the full width of the nozzle and set the coordinates of A_1 and A_5 as:

$$\begin{aligned} (r_1, x_1) &= (-r_0, 0), \\ (r_5, x_5) &= (r_0, 0). \end{aligned} \tag{3.2}$$

The coordinates of A_2 and A_4 are determined by equations:

$$\begin{cases} \partial P / \partial x = 0, \\ P = P_{\min}; \end{cases} \tag{3.3}$$

and the coordinates of A_3 are determined by equations:

$$\begin{cases} r = 0, \\ P = P_{\min}. \end{cases} \tag{3.4}$$

Solving these equations we obtain the coordinates of A_2, A_3 and A_4 as:

$$\begin{aligned} (r_2, x_2) &= \left(-0.26 \frac{CD}{\sqrt{P_{\min}}}, 0.58 \frac{D}{\sqrt{P_{\min}}} \right), \\ (r_3, x_3) &= \left(0, \frac{D}{\sqrt{P_{\min}}} \right), \\ (r_4, x_4) &= \left(0.26 \frac{CD}{\sqrt{P_{\min}}}, 0.58 \frac{D}{\sqrt{P_{\min}}} \right). \end{aligned} \tag{3.5}$$

In our algorithm implementation, we use the polygon $A_1A_2A_3A_4A_5$ as the effective water zone. Slightly more accurate simulations could be performed using an n -gon with more sides, at the expense of longer running times.

4. COMPUTING C-SPACES

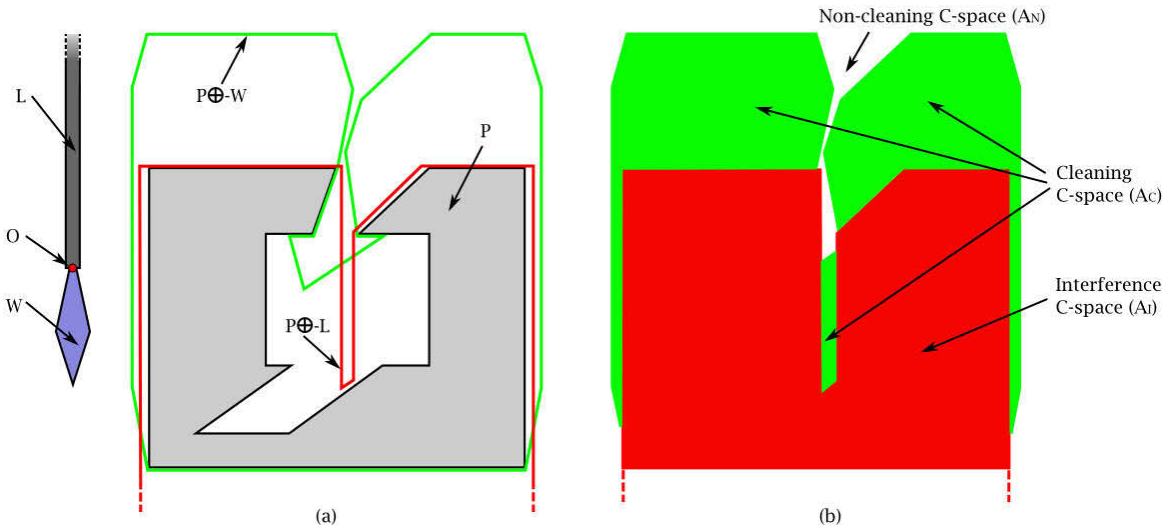


Fig. 4: Computation of the cleaning, non-cleaning, and interference C-spaces. O is the reference point.

As the first stage of our algorithm, we divide the entire configuration space into three subsets based on the geometry of the lance, the workpiece, and the effective water zone. A configuration of an object is defined by independent parameter values that characterize the position the object [10]. In our simplified cleaning problem, the lance system is a rigid polygon which can only translate in the plane, so each configuration has two degrees of freedom. We pick a point in the polygon as the *reference point*, and use its x and y coordinates to specify the corresponding configuration. The set of all configurations is the configuration space (C-space). Some configurations are forbidden during the cleaning process because the lance will collide with the workpiece under these configurations. The others are safe; however some of them are not valid for cleaning if the effective water zone doesn't "touch" some region of the surface of the workpiece. (Note that not all surfaces touched by the efficient water zone will necessarily be cleaned in the corresponding configuration due to visibility constraints, as described in section 5.) Thus we can divide the whole C-space of waterjet cleaning into three disjoint C-spaces:

- *Interference C-space* (A_I): the C-space where the lance collides with the workpiece;
- *Non-cleaning C-space* (A_N): the C-space where the lance doesn't collide with the workpiece but the waterjets fail to effectively clean the workpiece;
- *Cleaning C-space* (A_C): the C-space where the waterjets can effectively clean some regions of the workpiece.

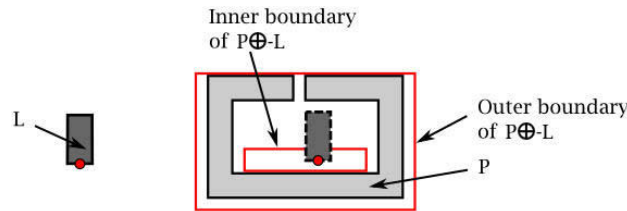


Fig. 5: The lance cannot access holes in $P \oplus -L$, such as in the dashed configuration pictured, although these configurations would not be in the interference C-space.

We represent the lance, the effective water zone, and the workpiece as polygons L , W , and P respectively, and place the reference point O at the center of the water nozzle, as shown in Fig. 4(a). We also assume that the workpiece P doesn't have any self-intersections or holes (holes, if any, can be marked as non-cleanable and removed since the lance can never go into these holes). Since we don't consider the lance holder geometry in our problem, we model the lance as a rectangle with infinite height to simulate the existence of the lance holder. In practice a height larger than the diagonal of the workpiece's bounding box is sufficient. (Alternatively, a polygon with more sides could be used to accurately represent the combined lance and lance holder geometry.) Computation of the three C-spaces can be described using Minkowski sums as follows:

$$\begin{aligned}
 A_I &= P \oplus -L \\
 A_C &= (P \oplus -W) \setminus A_I \\
 A_N &= (A_I \cup A_C)^c
 \end{aligned}
 \tag{4.1}$$

In the above equation, the interference C-space A_I is the Minkowski sum of the workpiece P and the inverse lance $-L$. Since L has an infinite height, there will be no holes in $P \oplus -L$ (because a ray emanating from any point in $P \oplus -L$ and going down vertically also lies in $P \oplus -L$). Thus we avoid the case where the lance can be "placed" in cavities without collision but cannot access them from the outside (see Fig. 5). The cleaning C-space A_C is the difference between $P \oplus -W$ and A_I . The non-cleaning C-space A_N is the complement of the union of the cleaning and the interference C-space. The computation result of these three C-spaces is illustrated in Fig. 4(b), where the (light) green area is the cleaning C-space and the (dark) red area is the interference C-space.

5. FINDING CLEANABLE REGIONS

As the second stage of our algorithm, we find all the cleanable regions, building on the computation of the three C-spaces, which give us information of where we can place the lance for effective cleaning and where the lance collides with the workpiece. Under any configuration in the cleaning C-space, the waterjets clean some regions of the workpiece. In this section, we describe an algorithm for finding all such cleanable regions of a workpiece, given a translating lance. We solve this problem in two steps. First we find the cleanable regions under a specific configuration (lance position), taking visibility into account (section 5.1), and then we use a simulation approach to find all such cleanable regions (section 5.2).

5.1 Finding Cleanable Regions under a Specific Configuration

For a configuration $c \in A_c$, we want to find the cleanable regions when the lance is positioned in this configuration. These cleanable regions are a subset of the surface regions that lie inside the effective water zone, since they must also be visible from the nozzle. The actual nozzle has a radius that is usually very small compared to the size of the lance or the workpiece. In our algorithm, we test the visibility from point O' , the intersection of lines A_2A_1 and A_4A_5 bounding the effective water zone (see Fig. 6a).

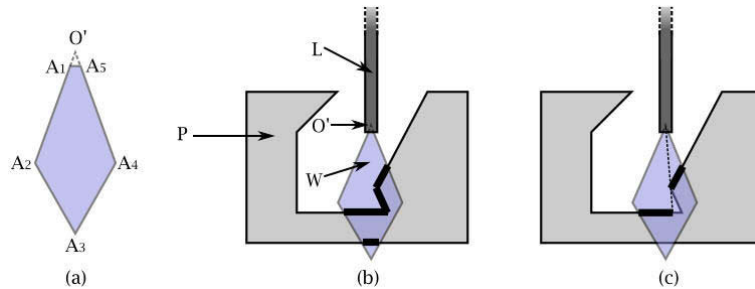


Fig. 6: Computation of the cleanable regions under a specific configuration.

The algorithm for finding cleanable regions under a specific configuration consists of three steps:

1. Compute the regions of the workpiece that lie inside the effective water zone, which is a set of line segments, denoted as R_E (highlighted in Fig. 6b);
2. For each line segment in the set R_E , compute the portion of it that is visible from the nozzle (highlighted in Fig. 6c);
3. The cleanable regions under this configuration are the union of all such visible sub-segments.

For the first step, we can use either a naïve approach to compute the intersections between each edge of the workpiece and the effective water zone, or a more efficient one with hierarchical data structures such as quadtrees. For the second step, we first give a clear definition of visibility to handle ambiguous cases. We say a point Q is *visible* from point O' if the line segment $O'Q$ doesn't intersect the interior of the workpiece. For simplicity, in the remaining of the paper, when we say a point or a region is visible, we mean that it is visible from the point O' . According to this definition, points A and D in Fig. 7(a) are not visible, but points B and C are visible. We give the algorithm for computing the visibility of a point below.

Algorithm 1: compute the visibility of point Q

1. Find all intersection points between the line segment $O'Q$ and the boundary of the workpiece. (If an edge of the workpiece overlaps $O'Q$, only its endpoints are deemed potential intersection points.)
2. Store these intersection points in a list L and sort them lexicographically by coordinates.
3. $visibility \leftarrow \text{TRUE}$

4. **for** each pair of adjacent points $L[i], L[i + 1]$ in L **do**
5. $M \leftarrow (L[i] + L[i + 1]) / 2$
6. **if** M is in the interior of the workpiece **then**
7. $visibility \leftarrow FALSE$
8. **end if**
9. **end for**
10. **return** $visibility$

The visible regions on a line segment are the set of all visible points on it. We can prove by contradiction that for a line segment on the workpiece, its visible regions are connected.

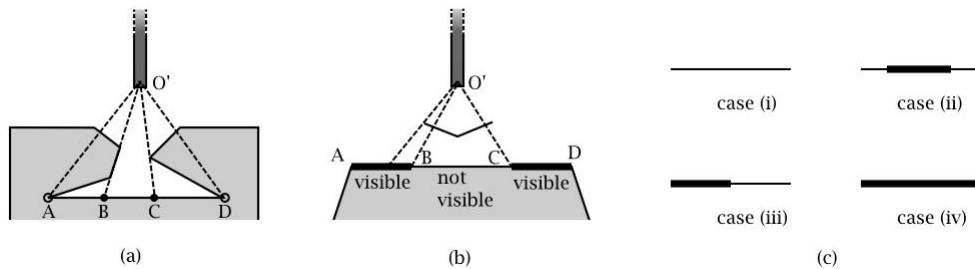


Fig. 7: (a) Points A and D are not visible; point B and C are visible. (b) Disjoint visible regions on a line segment. (c) Four different visibility cases for a line segment.

Theorem 1: The visible regions of a line segment on the workpiece are connected.

Proof: Suppose that on a line segment AD , there exist disjoint visible regions AB and CD (Fig. 7b). Since BC is not visible, there must exist a region of the workpiece inside $\Delta O'BC$. Since the workpiece is a polygon without holes, the boundary of this region should be connected with line segment AD , and it cannot cross line segment BC (otherwise the workpiece is self-intersecting). It cannot go through point O' either, otherwise the lance would be colliding with the workpiece. So this region of the workpiece inside $\Delta O'BC$ has to cross the interior of line segment $O'B$ or $O'C$, which causes the neighborhoods of B or C to be invisible. This contradicts the assumption that the whole of line segments AB and CD are visible. \square

Based on this theorem, we can conclude that any line segment on the workpiece belongs to exactly one of the four different visibility cases shown in Fig. 7(c): the whole line segment is invisible (case i), a region in the interior is visible (case ii), a region on one end is visible (case iii), or the whole line segment is visible (case iv). Thus, to compute the visible regions of a line segment, we first find the visible vertices of the workpiece that occlude the line segment from the nozzle, if any. We then compute the distinct intersection(s) between the ray(s) emanating from the nozzle that pass(es) through these vertices, and the line segment. If the number of distinct visible intersection points is one or two, the line segment belongs to case (iii) or case (ii), respectively. If zero, it is either case (i) or case (iv), which can in turn be differentiated by the visibility of the middle point of the line segment. The detailed algorithm is described below.

Algorithm 2: compute the visible regions of a line segment Q_1Q_2 on the workpiece

1. Find all visible vertices of the workpiece, and store them in an array A_{vert}
2. Initialize an empty point array A_{pnt}
3. **for all** $V \in A_{vert}$ **do**
4. **if** ray OV intersects the line segment Q_1Q_2 at point T and T is visible **then**
5. add T to A_{pnt}

```

6.   end if
7.   end for
8.   Remove duplicate points in  $A_{pnt}$ 
9.   if  $size(A_{pnt}) = 1$  then {this is case (iii)}
10.  if  $Q_1$  is visible then
11.      return line segment  $(Q_1, A_{pnt}[0])$ 
12.  else
13.      return line segment  $(A_{pnt}[0], Q_2)$ 
14.  end if
15.  else if  $size(A_{pnt}) = 2$  then {this is case (ii)}
16.      return line segment  $(A_{pnt}[0], A_{pnt}[1])$ 
17.  else if  $size(A_{pnt}) = 0$  then
18.       $M \leftarrow (Q_1 + Q_2) / 2$ 
19.      if  $M$  is visible then {this is case (iv)}
20.          return line segment  $(Q_1, Q_2)$ 
21.      else {this is case (i)}
22.          return NULL
23.      end if
24.  else
25.      assert(FALSE) {this should not happen}
26.  end if

```

We apply this algorithm to each line segment in the set R_E to find its visible regions, and then take the union of all these visible regions. The union is the cleanable regions under the current configuration.

5.2 Finding All Cleanable Regions

All possible cleanable regions are the union of the cleanable regions when the lance is positioned under each configuration of the cleaning C-space. A naïve approach might sample the whole cleaning C-space, calculate the cleanable regions under each sample configuration, and then take their union. However, we can prove that it is sufficient to sample only the boundary between the cleaning and the interference C-spaces to cover all the possible cleanable regions.

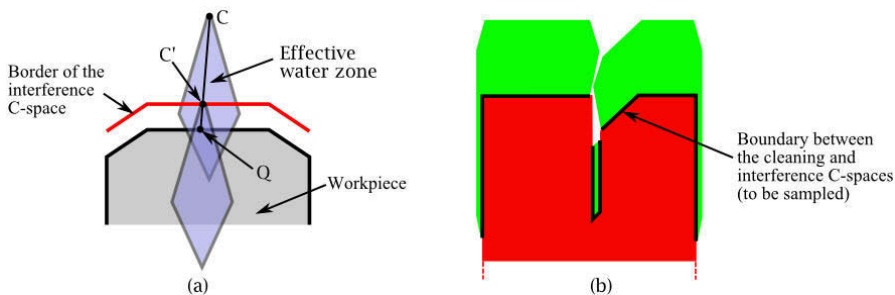


Fig. 8: Only the boundary between the cleaning and interference C-spaces needs to be sampled.

Theorem 2: If a point on the workpiece is cleanable, then it is cleanable under some configuration on the boundary between the cleaning and interference C-spaces.

Proof: Suppose that a point Q on the workpiece is cleanable. There must exist a configuration C in the cleaning C-space such that Q can be cleaned when the lance is positioned at C (see Fig. 8a). Since Q lies inside the interference C-space and C lies outside, the line segment QC must intersect the border of

the interference C-space. Call the intersection point C' (if more than one intersection point exists, we pick the one closest to C). Because of the convexity of the effective water zone, Q still lies in the effective water zone if the lance is moved from C to C' . Since Q is visible from C , it is also visible from C' . So Q can be cleaned under configuration C' . Since C' lies on the border of the interference C-space and the lance positioned at it cleans part of the workpiece, C' must belong to the boundary between the interference and cleaning C-spaces. \square

For the example in Fig. 4, the boundary between its cleaning and interference C-spaces are shown in Fig. 8(b). To compute the boundary, we first compute the border of $P \oplus -L$ (see Fig. 4a), then find the portions that lie inside $P \oplus -W$. In practice, we offset the boundary into the cleaning C-space for a very small distance to avoid the situation where a portion of the boundary may overlap the edges of the workpiece. We then regularly sample points along the boundary, compute the visible regions for each sample point using the algorithm described in section 5.1, and take the union of all these visible regions.

6. IMPLEMENTATION AND DISCUSSION

We implemented all the above algorithms based on CGAL [4], which offers robust implementations of some basic algorithms like Minkowski sums, point membership classification (PMC), intersection point computation, etc. We used the predefined Cartesian kernel with exact predicates and constructions to guarantee robustness when handling degenerate cases like the visibility computation in Fig. 7(a) [7].

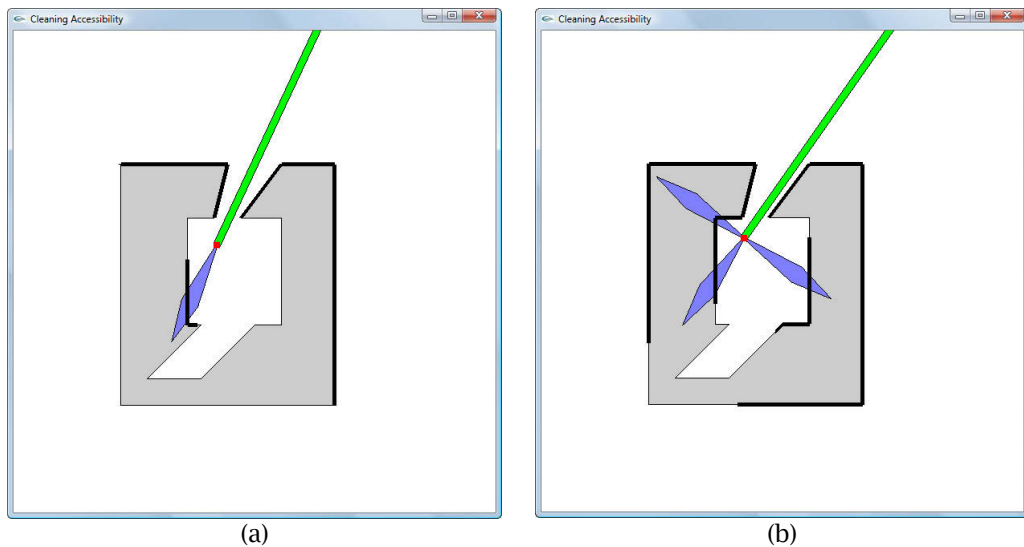


Fig. 9: Example results of our program. All the possible cleanable regions for a slanted (a) single-orifice and (b) multi-orifice lance are highlighted.

In our interactive program, the user can translate the lance or change its orientation using the keyboard. The color of the lance indicates the current C-space it lies in: red means the interference C-space, green means the cleaning C-space, and gray means the non-cleaning C-space. The cleanable regions under the current configuration are highlighted. The user can also invoke the simulation to find all possible cleanable regions. Our program supports multi-orifice nozzles as well. Two examples of our results are shown in Fig. 9. The accuracy of the results is determined by the sampling frequency on the boundary between interference and cleaning C-spaces. In our implementation, a fixed sampling step equal to the nozzle radius r_0 (see equation 3.2) was used to prevent unintended gaps between the cleanable regions due to the discrete sampling.

An exact algorithm for finding all the cleanable regions remains as future work. A promising approach would find the exact locations of “critical points” on the boundary of the workpiece (the endpoints of the highlighted line segments in Fig. 9). To find these critical points directly would be difficult; instead we find a set of possible candidate points, which includes all the actual critical points. Such candidates subdivide the boundary of the workpiece into many line segments. Inside each line segment, all points have the same cleanability, i.e., they are all either cleanable or non-cleanable. To find the possible candidates for critical points, we consider each pair consisting of an edge (e_w) of the workpiece and an edge (e_c) of the boundary between the interference and cleaning C-space. Except the original vertices of the workpiece, possible candidates include, but are not limited to:

- Intersection points between e_w and the boundary of $e_c \oplus W$, such as C_1 and C_2 in Fig. 10(a);
- Extreme points caused by the occlusion of another vertex on the workpiece, such as C_3 in Fig. 10(b).

Conditions for finding the complete set of candidates remain future work.

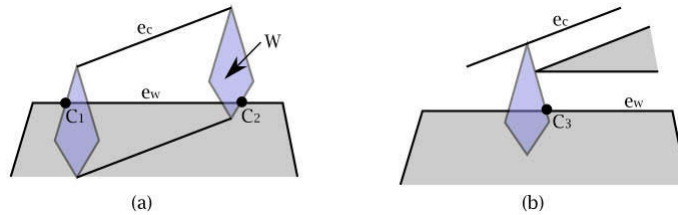


Fig. 10: Possible candidates for critical points.

The approach described in this paper can be extended to 3D as well. Algorithm 1 for computing the visibility of a point still holds in 3D, but Theorem 1 is no longer true - the visible regions on a face of a 3D workpiece may be disjoint. To find the visible regions on a face, we can employ a spherical visibility map [13] or graphics-hardware based visibility algorithms [8]. Theorem 2 still holds true in 3D, therefore we only need to sample the boundary faces between the interference and cleaning C-spaces to find all the possible cleanable regions.

7. ACKNOWLEDGEMENTS

The authors were supported in part by NSF under grant number DMI-0547675 and 0621198.

8. CONCLUSIONS

In this paper we presented an approach for finding all the cleanable regions of a polygon under a 2D waterjet cleaning process by means of geometric accessibility analysis. Cleanability is determined by three factors - visibility from the nozzle, sufficient impact pressure, and a collision-free position of the lance. Our approach first constructs the cleaning C-space where the lance doesn't collide with the workpiece and at the same time the waterjet produces sufficient impact pressure for effective cleaning. Visibility testing is then performed to find the cleanable regions under a specific configuration. The boundary between the cleaning and interference C-spaces is sampled to find all the possible cleanable regions. Results demonstrate that our approach can be applied to different lance orientations and multi-orifice nozzles.

8. REFERENCES

- [1] Arbelaez, D.; Avila, M.; Krishnamurthy, A.; Li, W.; Yasui, Y.; Dornfeld, D.; McMains, S.: Cleanability of mechanical components, Proceedings of 2008 NSF Engineering Research and Innovation Conference, Knoxville, Tennessee, Jan 2008.
- [2] Avila, M.; Reich-Weiser, C.; Dornfeld, D.; McMains, S.: Design and manufacturing for cleanability in high performance cutting, Proceeding of 2nd International High Performance Cutting Conference, CIRP, Vancouver, BC, 2006, Pager Number 63.
- [3] Berger, K.: Burrs, chips and cleanness of parts - activities and aims in the German automotive industry, Presentation at CIRP Working Group on Burr Formation, Paris, January, 2006.
- [4] CGAL, Computational Geometry Algorithms Library, <http://www.cgal.org>.

- [5] Choi, B.; Kim, D.; Jerard R.: C-space approach to tool-path generation for die and mould machining, *Computer-Aided Design*, 29(9), 1997, 657-669.
- [6] Fogel, E.; Halperin, D.: Exact and efficient construction of Minkowski sums of convex polyhedra with applications, *Computer-Aided Design*, 39(11), 2007, 929-940.
- [7] Halperin, D.: Robust geometric computing in motion, *the International Journal of Robotics Research*, 21(3), 2002, 219-232.
- [8] Khardekar, R.; Burton, G.; McMains, S.: Finding feasible mold parting directions using graphics hardware, *Computer-Aided Design*, 38(4), 2006, 327-341.
- [9] Leu, M. C.; Meng, P.; Geskin, E. S.; Tismeneskiy, L.: Mathematical modeling and experimental verification of stationary waterjet cleaning process, *Journal of Manufacturing Science and Engineering*, 120(3), 1998, 571-579.
- [10] Lozano-Pérez, T.: Spatial planning: a configuration space approach, *IEEE Transactions on Computers*, 32(2), 1983, 108-220.
- [11] Spitz, S. N.; Spyridi, A. J.; Requicha, A. G.: Accessibility analysis for planning of dimensional inspection with coordinate measuring machines, *IEEE Transactions on Robotics and Automation*, 15(4), 1999, 714-727.
- [12] Varadhan, G.; Manocha, D.: Accurate Minkowski sum approximation of polyhedral models, *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference*, 2004, 392-401.
- [13] Woo, T. C.: Visibility maps and spherical algorithms, *Computer-Aided Design*, 26(1), 1994, 6-16.