

Cellular Decomposition for Non-repetitive Coverage Task Ensuring Least Discontinuities

Tong Yang¹, Jaime Valls Miro², Qianen Lai¹, Yue Wang^{1*} and Rong Xiong¹

Abstract—There are many non-repetitive coverage tasks which use a high-dimensional path to cover low-dimensional space, such as the polishing task using the manipulators. Due to the non-bijective mapping between the workspace and the joint-space, a continuous coverage path in the workspace is truncated in the joint-space, and there are multiple choices of configurations to cover a same waypoint of the coverage path. The discontinuous point implies a suspension of the coverage task, which is difficult to handle. For example, the suspension of the polishing task means the lift-off the end-effector from the surface of the object, where the transition between the force/position control is required, which is more complicated than the ordinary coverage process.

In this paper, motivated by the non-repetitive coverage task of non-redundant manipulators, we first prove that the least number of discontinuity is a parameter of the environment setting, independent to the choice of coverage paths, thus has a minimum. Then, a cellular decomposition method generating all optimal cellular decompositions is proposed. The algorithm is operational in any dimension, and we illustrate it through polishing tasks. Through simulated experiments, the least number of discontinuities is shown as a novel criterion of the quality of the placement of the manipulator (or the object). Besides, the method helps to solve the problem of multiple inverse kinematic (IK) solutions. In real-world experiment, the physical coverage path is generated to show the applicability of the proposed algorithm. [The video of the real-world experiments is given here](#):

I. INTRODUCTION

THE non-repetitive *coverage task* of a given object is an important application carried out by manipulators. This is for instance the case of inspecting a surface for defects at close range, painting, deburring or polishing. The task is effectively encapsulated as the generic coverage path planning (CPP) [?] [?] problem, which requires for the end-effector (EE) to traverse over all the points that define the surface of a given object exactly one time, whilst usually fulfilling additional task-specific constraints (e.g. sustain a desired orientation of the EE with respect to the surface, maintain contact or exerting a constant EE force/torque). Typically, joint-space dimension is higher than the workspace's, and the inverse kinematic (IK) mapping between task and joint space is thus non-bijective. As a result, planning in the higher dimension joint-space cannot ensure non-repetitive visiting, [with the task-to-joint mapping being non-injective](#), and coverage paths are

¹ Tong Yang, Qianen Lai, Yue Wang and Rong Xiong are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, P.R. China.

² Jaime Valls Miro is with the Centre for Autonomous Systems (CAS), Faculty of Engineering, University of Technology Sydney (UTS), NSW 2007 Sydney, Australia.

* Corresponding Author.

E-mail address: wangyue@iipc.zju.edu.cn

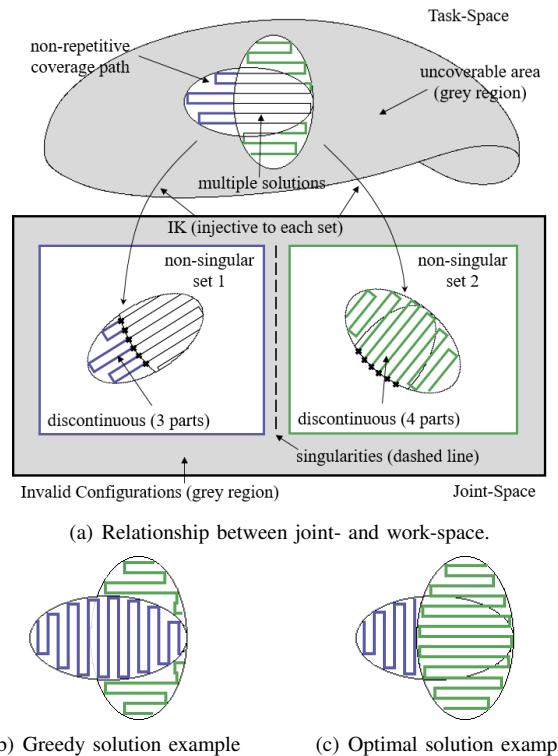


Fig. 1. (a) Illustration of the coverage task problem and the relationship between joint- and work-space. Different colors denote disjoint sets of non-singular joint configurations (arbitrarily, blue may for instance represent those with elbow-up, whilst green may represent elbow-down), and their corresponding path in the workspace. The colored segments of the arbitrary coverage path shown in black have unique IK solutions. However, multiple IK solutions exist for the intersecting area shown in black. The underlying continuous coverage path sought out thus becomes intermittent in the workspace after mapping onto the joint-space. In this example, whatever choice among the multiple IK solutions, six discontinuities are required (depicted by black crosses), since the path has three separate segments in set 1, and four in set 2. The case where the joint-space solutions are taken in full from set 2 (green) is depicted. (b) Starting from a configuration belonging to set 1, without explicitly calculating the reachable boundary of each set, the boundary of the set 2 within the reachable area of set 1 is unknown. So a greedy strategy will fully cover the set 1, dividing the uncovered region into two parts, leading to an extra lift-off. On the other hand, although at first sight it may appear the same as using the greedy strategy starting from set 2 (c) illustrates the concept of CPP optimality' in the joint space, whereby the continuity of the reachable area is explicitly considered, thus producing a coverage path with a single EE lift-off.

thus more suited to be designed directly in the workspace domain [?].

Yet what constitutes a continuous coverage path in the workspace may easily end up truncated into many seemingly *intermittent* sections after mapping them back onto the joint-space, with undesirable path discontinuities, as graphically

illustrated by Fig. 1. This is also the case if a simplistic greedy strategy is followed, as the example depicted in Fig. 1(b), whereby a complete path in task space that solves for all possible configurations leads to unnecessary lift-offs to accomplish full coverage. The problem is further compounded by additional task constraints, most notably obstacles, which produce usable configurations further divided into many disjoint sets, possibly requiring more undesirable “jumps” between sets for successful coverage. This work advocates for the minimisation of the cost incurred on these path discontinuities, which can significantly outweigh any improvements that may occur locally in the task space when it comes to coverage [?]. This is perhaps more apparent for the case of the uniform polishing task motivating this work, as that means lifting the EE off the object’s surface, adjusting the pose of the manipulator to the new configuration, and landing back into contact with the surface again. This may be not only sub-optimal for the speedy completion of the CPP task, but also introduces potentially avoidable complexity in transitioning between position and force/torque control [?] [?] [?] during the coverage task.

In this work, a mechanism is proposed to address this shortcoming and derive CPP solutions with a proven minimal number of discontinuities, with the aim to avoid unnecessary, costly EE lift-offs. The solution departs from locally optimising the shape of the coverage path in task space, or choosing appropriate but possibly disconnected configurations, but considering the reachability of each continuous motion globally. The solution departs from locally optimising the shape of the coverage path in task space, or choosing appropriate but possibly disconnected configurations, but explicitly seeking for least number of discontinuous motion through the analysis of the structure of the valid configurations in the joint-space.

The work is predicated on the fact that singularities have been proven to be the cause of bifurcation of the joint-space [?] [?], i.e. sitting at the intersection of different configurations (e.g. elbow-up and elbow-down). Notwithstanding singularities, for non-redundant manipulators, non-singular configurations thus form disjoint sets in the joint-space, as was illustrated by the earlier example in Fig. 1. Moreover, due to task constraints (obstacles, joint limits, etc), commonly the whole workspace cannot be mapped into a single set, which then implies that continuous joint-space paths between sets must visit some singularities along the way, inevitably incurring undesirable lift-offs. However, manipulators are locally omni-directional in the joint-space, and configurations corresponding to a segment of coverage path without lift-off have high dimensional continuity in the joint-space, independently of their sequencing order. Hence, instead of considering the design of a coverage path in the traditional sense, this work considers the global optimal cellular decomposition problem in joint-space to incur joint-space partitions with minimum sets. It is further noted that IK mapping from the reachable points in the workspace to a single set of configurations is injective, since there is no non-singular path connecting two configurations whose EEs are at a same point. By assigning a class (colours are used in this paper for easier visualization) to each configuration based on their joint-space continuity, different IK solutions for a same EE pose must possess a different colour.

As such, colouring a point in the surface to be covered means selecting a given IK solution for it, and the planning problem is transferred to designing a colour scheme for a topological configuration graph. In that way, the key concern is the joint-space continuity of the cells, and by efficiently discarding equivalent cellular decompositions, this work proves that total number of different cellular decompositions is finite, thus all optimal solutions are finitely solvable.

The two novel contributions of this paper can thus be summarised as:

- 1) Proving that the minimum number of path discontinuities, or “lift-offs”, for the non-repetitive coverage task with non-redundant manipulators is independent of the actual choice of coverage path. Instead, it is predicated on the surrounding environment - the relative pose between manipulator, object and the presence of any obstacles - and this motivates to formulate the problem as a global cellular decomposition process. On a side note, this also implies that the proposed scheme can be exploited as a criterion to evaluate the most advantageous placement of a manipulator, or object to be manipulated (e.g. polished, painted), both in a fixed configuration (automated production line), or in a mobile manipulation environment.
- 2) Proposing an effective finite cellular decomposition method to divide a workspace surface into the least number of cells whereby each is ensured to be traversable by any arbitrary inner path without incurring discontinuities.

The remainder of this paper is organised as follows. Section II reviews existing literature. Section III describes the proposed abstraction of the problem into a topological graph of surface cells corresponding to feasible, continuous configurations, hence administering the tools to prove that the number of path discontinuities for the CPP problem can be made independent to the eventual coverage path chosen. Section IV goes into further details about the process of finitely resolving the surface into cell elements, whilst Section V reports on the proposed iterative strategy to build on the cell elements to ensure CPP with a minimum number of discontinuities. Experimental results from simulations and on an actual non-redundant manipulator are collected in Section VI, with final concluding remarks gathered in Section VII.

II. RELATED WORK

Almost all state-of-the-art methods of the CPP problem first divide the given area and then solve the CPP problem in each cell, so called the cellular decomposition, which is further divided into two categories: the exact cellular decomposition methods [?] and the morse-based cellular decomposition methods [?] [?]. The exact cellular decomposition methods divide the free space into several simple, easy sub-regions, and use conventional coverage paths, such as the trapezoidal path [?] or the boustrophedon path [?] [?], to finish the coverage in each cell. The morse-based cellular decomposition methods apply the divisions of the free space based on the critical points of Morse functions to present more flexible shape for cells than the exact cellular decomposition.

The optimal CPP algorithms mainly focus on the path length and time to completion. Atkar et al. [?] optimized the coverage path through choosing optimal starting point. Huang [?] reduced the movement cost through using straight path as long as possible to minimize the number of returns. Jimenez [?] used a genetic algorithm to achieve optimal coverage. However, for the manipulator, avoiding an unnecessary lift-off overweighs any improvement of the performance during the normal coverage process, e.g., changing from boustrophedon path to spiral path [?].

Since the discontinuities within the CPP task occurs only for the manipulator, none of the algorithms designing for the mobile robots considers this problem. We notice that [?] considers optimizing the position of the mobile manipulator for coverage task, which strengthens the requirement of a valid criterion of the relative pose between the manipulator and the object.

III. PROBLEM FORMULATION

In this section, we first state the problem of optimal coverage path planning ensuring least number of discontinuities. Then, we show that the least number of discontinuities is independent to the choice of the physical coverage path, so the original problem is transformed to an optimal cellular decomposition problem. Finally, the problem is further transformed to a painting problem of a given a graph ensuring least number of different colors.

A. Problem Statement

Given the surface of an object, the structure of the manipulator, the shape of other obstacles and their relative poses, a valid coverage path consists of all valid joint-space poses of the manipulator which satisfy the following constraints:

(1) Kinematic Constraint: The manipulator is collision-free during the whole process. And when the EE contacts the surface, its z -axis is align with the normal vector of the surface at the contact point.

(2) Force Constraint: When the EE contacts the surface, the manipulator should be able to exert the required force on the EE in the z -axis.

(3) Manipulability Constraint: When the EE contacts the surface, the manipulator should keep well-conditioned (under some given manipulability measure) to deal with the perturbation.

Then, the optimal CPP problem is to find a valid joint-space path whose EE covers the workspace non-repetitively and ensures the least number of discontinuities. In this problem, the contact area between EE and the surface is seen as a particle.

B. Independence from the physical coverage path

Before introducing the construction of the proposed algorithm, an observation which simplifies the original problem is that the manipulator is omni-directional in the joint-space. A continuous joint-space path implicitly specify a continuous region in the joint-space in which we can find infinite many different coverage paths without lift-off. For example, in

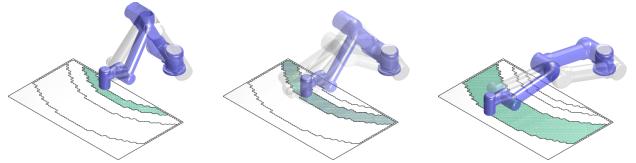


Fig. 2. Illustration of different IK solutions for covering a same points. The vivid configurations show one kind of configurations with shoulder-right, wrist-unflipped. There are also two kinds of configurations (with shoulder left/right and wrist flipped) that are usable. However, they can only cover the middle part of the reachable area.

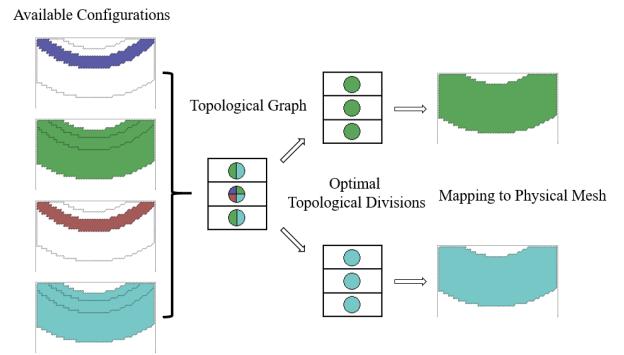


Fig. 3. Flowchart of our algorithm in solving the problem of Figure 2. All configurations are divided into four disjoint sets based on their joint-space continuities, represented by a same color. The small circle filled in with color(s) shows all possible colors to paint the corresponding area. The topological graph is created based on the distribution of the colors and being solved. Finally, in this example, we can get two optimal options which both require zero lift-off.

Figure 1, the ellipsoid in each set is coverable, whatever using a boustrophedon path in any direction, or using a spiral path starting from any boundary points. As a result, we are inspired to consider only the continuous region in the joint-space and its corresponding reachable area in the workspace, instead of the coverage path, which is equivalent to a cellular decomposition problem in the workspace considering the joint-space continuity. In the equivalent problem, each cell is guaranteed coverable with a joint-space continuous coverage path, thus no discontinuities. Once the cells are determined, the CPP problem within each cell is then trivial. In all, the original optimal CPP problem is transformed into a cellular decomposition problem considering the joint-space continuity.

C. Problem Modeling

First, we define some symbols. Let \mathcal{C} be the set of all valid configurations and M be the set of all reachable points on the surface. The pose of the EE is also denoted by M since there is an one-to-one correspondence between the pose of the EE and the point on the surface, so we do not distinguish them.

Second, the *color* in Figure 1 has significant meaning which deserves formal definition. In mathematical language, the joint-space continuity of two configurations is an equivalence relation (a reflexive, symmetric and transitive relation). Denote the equivalence relation by \sim , then each element in the quotient set \mathcal{C}/\sim corresponds to an unique color. Now we show the assignment of the color in usual language. Given a

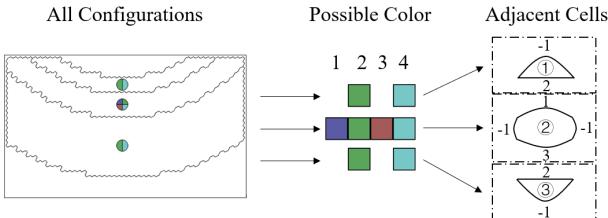


Fig. 4. TODO: add correspondence with physical mesh, show the connectivity of the mesh The elements of a cell. All colors are indexed. Each cell corresponds to a connected area on the surface of the object and has a list of possible colors. The cell stores the index of its adjacent cell in order. The index of the unreachable area is denoted by -1 . We use curves to show that the edge is topological but not physical.

configuration $p \in \mathcal{C}$ covering $m \in M$, following the joint-space continuity, there exist a neighborhood $(p \in) U_p \subset \mathcal{C}$ that can be reached continuously (without lift-off) from p , covering a piece of surface $(m \in) V_m \subset M$. See Figure 2, the poses of the manipulator in vivid color can be reached continuously. If any of them is chosen as p , then all of them are in U_p . Assume there are some other unassigned configurations, i.e., $\mathcal{C} \setminus U_p \neq \emptyset$, we choose another one $p' \in \mathcal{C} \setminus U_p$, then it specifies another set $V_{m'} \subset M$. Of course $V_{m'} \cap V_m = \emptyset$. After continuing this process, all configurations are assigned. Then \mathcal{C} is divided into (finite, where we omit the strict proof) number of disjoint sets, denoted by finite number of different colors.

Third, the *cell*, which is defined on the task-space, the same as that of the conventional cellular decomposition methods, but has a property of homogeneity. The basic observation is that the IK mapping shown in Figure 1 is injective in each branch. The multiple IK solutions appear when the inverse trigonometric function returns multiple solutions. Corresponding to the physical meaning of the 5DOF manipulator, the multiple solutions of the elbow joint causes “elbow-up/elbow-down”, etc. Without strict proof, there exist a joint-angle leading to a singularity in between (“elbow-straight” in this case). For example, in each subfigure of Figure 2, the greyed out configurations have the same pose of the EE with the vivid one, so they must be assigned with different colors. **The injectivity of each branch of the IK motivates us to map the property of joint-space continuity back to the surface, and thus the process of the algorithm can be visualized through drawing color on the surface.** For example, in Figure 3 we know that \mathcal{C} is divided into 4 disjoint sets. Since different IK solutions possess distinct colors, the available colors for points can be used to classify them. Let $\{c_i\}, i = 1, n$ be all colors, then for two points $m_1, m_2 \in M$ with their sets of available colors $c_{m_1} = \{c_{11}, \dots, c_{1i}\}, c_{m_2} = \{c_{21}, \dots, c_{2j}\}$, we say that m_1 and m_2 belong to a same cell if and only if

$$\begin{cases} m_1 \text{ and } m_2 \text{ are connected} \\ \{c_{11}, \dots, c_{1i}\} = \{c_{21}, \dots, c_{2j}\} \end{cases}$$

Typically, for the triangular mesh in our case, the connectivity is provided by the edges of the mesh. Figure 3 shows the creation of the cells.

Finally, a topological graph is created, whose element is all cells. Each cell possesses an index, records the possible

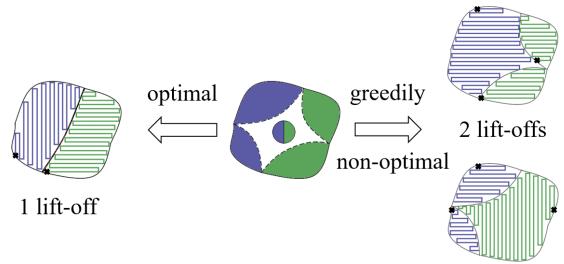


Fig. 5. In this graph, only the middle cell still needs solving. Whatever pure color filling into the middle cell, it requires an extra lift-off. Instead, if it is cutted into two parts, with two sub-cells filled in with different colors, only 1 lift-off is required which reach optimality.

colors and the indices of its adjacent cells in order, like Figure 4. Note that since the area of the surface is finite, and each cell must have least size to keep the decomposition meaningful, the number of cells must be finite. After creating the topological graph, the cellular decomposition process is transformed to painting all points in a graph with one of their available colors. The number of pieces in the graph means the number of coverage path segments, where discontinuities are required in between. So the optimal cellular decomposition problem is transformed to the painting problem ensuring the minimum pieces of colors. For a fully-filled graph, the distribution of the colors specifies a cellular decomposition ensuring no lift-off required in each cell, and the choice from all IK solutions at all points are uniquely specified by the corresponding color. So finally the problem is transformed to finding a color scheme for the graph ensuring least number of pieces.

IV. ENUMERATIVE SOLVER

The difficulty of solving the coloring problem is that, although the points are gathered into a same cell, they can be filled in with different colors, instead of only being seen as a whole and drawn with a single color, see an easy counter-example in Figure 5. We observe that the structure of the graph reflects in the connectivity of the topological edges, which is proved having only finite possible situations in IV-A.

In this section, we prove the finiteness of the number of divisions and introduce the enumerative solver through the example of solving a single cell with all its adjacent cells having been colored. We show that the simple cells with less than 4 topological edges can be enumeratively solved, and there are finite number of manners to divide a complicated cell into several simple cells, hence any cell can be solved.

A. Finiteness of Divisions

Since any path starting and ending at the boundary of a cell will divide the cell into two parts, there are infinite many physical solutions of dividing a cell into parts. However, there are only finite classes of them in the view of topological structure, because of the equivalence of physical divisions in the number of lift-offs.

See Figure 6(a), we show that the cutting paths which start or end at an ordinary point on some edges are unnecessary. Let a cutting path ends at an ordinary point of the edge

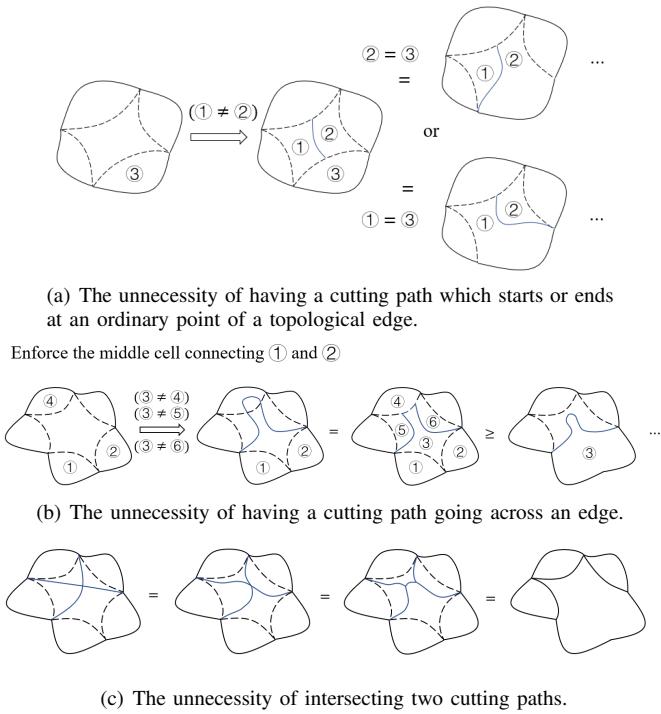


Fig. 6. Some physical divisions that are unnecessary or non-optimal.

connecting cell 3. From the definition of a cutting path, it implicitly enforces cell 1 and cell 2 having different colors. Then we may assume $1 = 3$ or $2 = 3$ (when $1 \neq 3$ and $2 \neq 3$ the division is trivial), which, however, is equivalent to two other cutting paths that start at the endpoint of the edge. Do the same discussion on the other endpoint of this cutting path, we know that it is complete to only consider all cutting paths which start and end at the endpoint of some edges.

See Figure 6(b), we show that the cutting paths which go across some edges are unnecessary. Let a cutting path go across the edge that connects cell 4. Then it can be continuously transformed onto the edge without changing the cost. This division enforces the constraint of color that $3 \neq 4, 3 \neq 5, 3 \neq 6$. However, it prevents cell 5 and cell 6 from being colored together, because they are separated physically by the cell 3, which may increase the cost and is not optimal. So we can directly discard the cutting paths which go across edges.

See Figure 6(c), we show that the cutting paths need not go across each other. When two cutting paths intersect, we can change the belonging of the path segment, and then the cutting paths can be continuously transformed onto the existed topological edges. So it is complete to discard the choices of cutting paths which have intersections.

In conclusion, we only need to consider all cutting paths which start and end at the endpoint of some topological edges and do not go across each other. Hence, the total number of topological divisions is finite and we just need to go through all possible divisions.

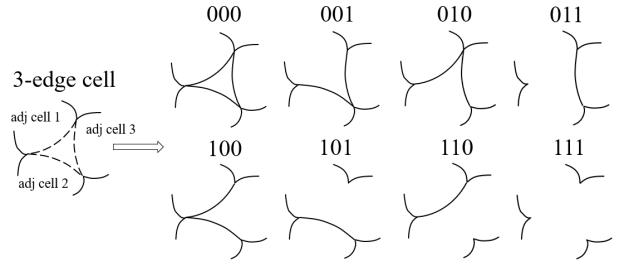


Fig. 7. All possible divisions of a 3-edge cell, which is the most complicated situation that need not further divisions. Although some of them are the same in the topological structure (e.g., 001, 010 and 100), or the division is impossible (e.g., we enforce 011 but the cell 1 and cell 2 do not have a same color), it has already been a finite problem, so we omit the description of further simplification.

B. Solution of Simple Cells

The following kinds of cells are so simple that can be solved directly without further divisions:

- (1) The cells containing less than four edges,
- (2) The cells with only one possible color,

because the cells of (1) cannot be divided further into several cells with less number of topological edges, and the cells of (2) have no other choice of color. We enumerate all possible topological divisions of a 3-edge cell (which is the most complicated case for direct enumeration) in Figure 7. We use a binary number to represent the connectivity of the edges, 1 for connection, 0 for disconnection and \times for the unspecified state. It is easy to see that there are at most 8 situations.

C. Solution of Complicated Cells

Following the idea of solving a simple cell, we use a binary number of length n to represent the connectivity of an n -edge cell, 1 for connection, 0 for disconnection and \times for the unspecified state. The continuous 1s imply that part of this cell must be painted with the same color as that of the adjacent cells which the 1s specify. The 0 means that the topological edge between the cell and the corresponding adjacent cell is kept so that the colors must be different. The unspecified states \times imply the generation of a sub-cell. An example of solving a 4-edge cell is shown in Figure 8. Through specifying the position of 1s, there are less than $2^n \times m$ branches for an n -edge cell with m possible colors, so the problem is finite. The distinction between 0 and \times will be given in subsection IV-D.

D. Discussion of Creating Sub-Cells

When there exist such number lists in the connectivities:

$$1 \times \times 1, 1 \times \times \times 1, \dots$$

the cell is divided into parts whose colors are enforced to be different, so called sub-cells. See the case of $\times \times 11, \times 11 \times, 1 \times \times 1, 11 \times \times$ in Figure 8. The original cell becomes a new one with fewer edges, because some edges are replaced by a single edge. We use the bracket (...) in the binary number of the original cell to represent the generation of sub-cells. Do the same division for the sub-cells, any n -edge

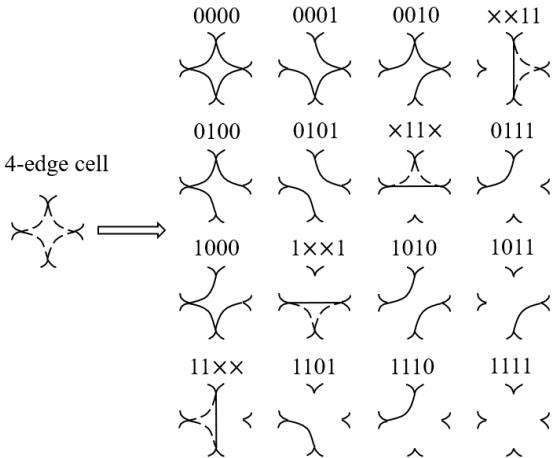


Fig. 8. The 2^4 possible divisions of an 4-edge cell. For the cell which has more than 3 edges, the sub-cell may be created. In this figure, the connectivities that correspond to generating a sub-cell is $\times \times 11$, $x11x$, $1xx1$, $11xx$.

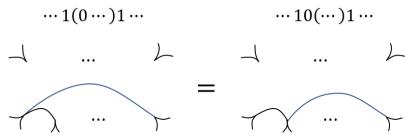


Fig. 9. The equivalence of moving the 0 outside the bracket. In order to reduce the number of edges of the sub-cell, we always enforce the sub-cell looking like the right side.

cell can be continuously divided into a set of 3-edge cells and then be solved enumeratively.

Since the sub-cell is generated from an original one, there are extra constraints on its connectivity specified by the previous division. However, these constraints cannot change this problem into one with polynomial time solution, so we just give some examples among them:

(1) Single \times cannot form a sub-cell, because

$$\dots 1 \times 1 \dots = \dots 101 \dots \text{ or } \dots 111 \dots$$

but both conditions of the right side are considered in other branches. This is why there is no \times in Figure 7.

(2) The 0s can be freely moved outside the bracket, because of the equivalence

$$\dots \times 1(0 \times \dots \times 1)1 \times \dots = \dots \times 10(\times \dots \times 1)1 \times \dots$$

and the same is true for the right bracket based on the symmetry of the number list, because the cutting graph makes sense only when the inner boundary two numbers are 1s. See Figure 9. This is why no brackets in Figure 8.

(3) The new topological edge created by the cutting path must be kepted (always 0, impossible to be 1 or \times) because it is manually created. Hence, for the entire problem, no extra possiblities appear after the divisions

V. ITERATIVE SOLVER

A. Iteration Process

Regarding the enumerative solver as a basic step, we iteratively solve the graph. Starting from a fully-unpainted

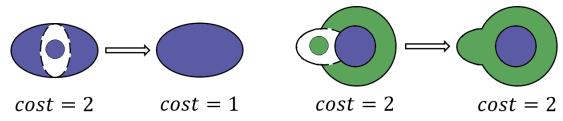


Fig. 10. Left: the middle cell connects two distinct cells, so the cost variation is $1 - 2 = -1$. Right: two edges connect with the same adjacent cell, then the cost variation is $1 - 1 = 0$ but not $1 - 2 = -1$.

graph, we choose an unsolved cell and enumeratively solve it. Assume that the cell has n -edges with m possible colors, there are at most $2^n \times m$ possible divisions. We create a branch for each possible solution of the closed cell, and in each branch the closed cell is filled in with the specified color (so it will not change any more). In the next iteration, we choose an unsolved branch, choose an unsolved cell in it and do the same steps as before. Note that the constraints given by the solved adjacent cells hugely restrict the possible solutions, because the state of an edge resticks the connectivity of the cells on both side. After iterative execution, all branches reach a contradiction or a valid coloring scheme. A valid coloring scheme for the graph uniquely specify the configuration to polish each point among its valid IK solutions. The algorithm runs like the deepest-first-searching (DFS) algorithm so that the memory requirement is not high. Since the execution is an exhaustive searching, all optimal physical cellular decompositions must be homeomorphic to one of our result schemes, with the position of the physical boundaries of the cells slightly different to the physical ones.

B. Calculation of Cost

The physical meaning of the cost for a (partly filled) graph is the number of pieces of color in the current graph. Describing the formula incrementally, after we solve a cell,

- (1) If its connectivity is all zero, then the cost will increase 1 after coloring this cell, because this cell forms a new piece.
- (2) If its connectivity has only one 1 connecting a solved cell, then the cost will not change, because this cell can be filled together with the connected adjacent cell.

(3) For its connectivity with i 1s, note that there may exist multiple edges which connect the same adjacent cell (See Figure 10). In order to be consistent with the physical meaning of the cost, if these edges connect j distinct solved cells, then the variation of cost is

$$\Delta\text{cost} = 1 - j$$

VI. EXPERIMENTAL RESULTS

Our algorithm works on any non-repetitive coverage task using non-redundant manipulators in any dimension. In this paper, experiments are taken by using a 5DOF manipulator to polish the surface of an object. And the manipulator should cover all reachable points even if it cannot fully cover the surface.

First, we discuss the proposed algorithm using behind other cellular decompositions, from which the proposed algorithm can be seen as a tool for the evaluation of other cellular decomposition algorithms. Then, in the first simulated experiment, a hemispherical object is polished at different poses,

one casually placed and the other being designed precisely. Through calculating the least required number of lift-offs, the quality of the placement of the object can be evaluated. In the second simulated experiment, we show that the proposed algorithm directly contributes to the choice of the configurations. We show some non-optimal configurations, which will never be chosen through applying the proposed algorithm. Finally, in the real-world experiment, the manipulator polishes a wok following a manually generated physical coverage path under the presence of the obstacles, which proves the applicability of the proposed algorithm. [The video of the real-world experiment is here](#):

Unless otherwise stated, the environment contains only the manipulator, the object and the ground plane. And note that all figures in this section are just an example of the optimal solutions. As we have discussed before, there are infinite many physical cutting paths of decomposition that result in same number of lift-offs of the EE, and the choices of the physical coverage paths within each cell are also arbitrary.

A. Calculating least number of discontinuities for other methods

Since all cellular decompositions divide the whole reachable region into parts, implicitly imply discontinuities between cells. Then, the original problem can be solved through finding coverage path in each cell, and the result path is the concatenation of the path segments in each cell using paths where the EE is lifted off. Let a cellular decomposition methods divides the surface into n parts, then $n - 1$ lift-offs are required. However, notice that tracking the coverage path in each cell still requires lift-off, denoted by p_i . Then the least number of lift-offs a given cellular decomposition is

$$n - 1 + \sum_{i=1}^n p_i$$

where p_i can be calculated through using the proposed algorithm in the i -th cell.

To show the optimality of the proposed algorithm, we apply it to the cellular decomposition created by itself. Since each cell is guaranteed polishable with no lift-off, $p_i = 0, \forall i$. Moreover, the number n is optimal since we use haustive searching strategy. Hence, the proposed algorithm certainly outperforms other methods in the sense of the number of lift-offs.

B. Covering a hemisphere

In this experiment, through the polishing task of a hemisphere shape object, we show that the number of lift-offs provided by our algorithm can evaluate the quality of the placement of the object (or the manipulator).

See Figure 11, a common scene is that the object is casually placed (e.g., on an automated production line) and a coverage path is designed, since there is no apparent criterion on the quality of the placement. However, the proposed algorithm shows that such a placement of the object requires at least 3 lift-offs, but still fails in full coverage (the farthest area is unreachable), which is equivalent to at least 4 lift-offs. Instead

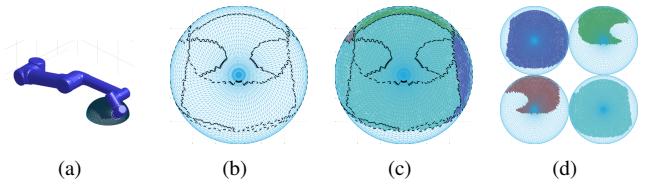


Fig. 11. (a) The object is casually placed. (b) The initial graph. (c) One optimal solution which requires 3 lift-offs, but the manipulator still cannot fully cover the farthest part of the mesh (the top area in the figure). (d) The reachable area of four kinds of valid configuration chosed by the optimal solution in (c).

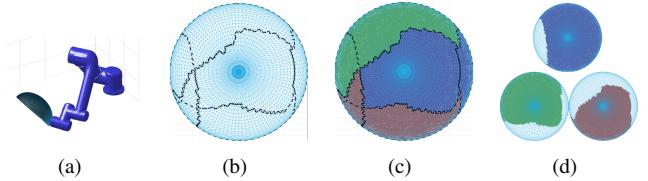


Fig. 12. (a) The object is placed obliquely. (b) The initial graph. (c) One optimal solution which only requires 2 lift-offs but realizes full coverage. (d) The reachable area of three kinds of valid configurations which are chosed by the optimal solution in (c).

of the usual setting, see Figure 12, if we precisely design the pose of the object (obliquely at $(0.7m, 0.1m, 0.08m)$), then not only the least number of lift-offs decreases to 2, but the manipulator can fully cover the surface, which performs much better than the scene in Figure 11. Hence, the required number of lift-offs guides the user to try different poses of the object (or the manipulator) to have better performance on coverage.

C. Covering a pipe

In this subsection, through the polishing task of a half pipe, we show that the proposed algorithm can clarify the unnecessary configurations, avoid the “trapped” configurations which cause extra lift-offs.

The pipe is placed obliquely. Although the object is common, the normal of its surface varies for π rad, which causes difficulty for the manipulator. We show the initial graph and directly give its optimal solution in Figure 13. The optimal solution of this coverage task requires only 1 lift-off.

There are many valid configurations which directly leads to non-optimality, because they just cover some points which have many IK solutions but break the connectivity of the uncovered region. See Figure 14 for an example of the distinguishability of the trap configurations. The first kind of configurations and the third one are finally chosen by one of the optimal solutions shown in Figure 13. The second kind of configurations can cover large area without lift-offs, which is very likely to be chosen if the IK solutions are chosen randomly or greedily. However, it cannot cover the corners of the mesh (which are eventually covered by the other two ones). Once any configurations belonging to the second kind is chosen, after the coverage of the middle part of the pipe, sooner or later it has to waste two lift-offs to finish the full coverage, which leads to non-optimality. The proposed algorithm provides all optimal solutions where none of them uses the second color, thus prevent from choosing these non-optimal configurations.

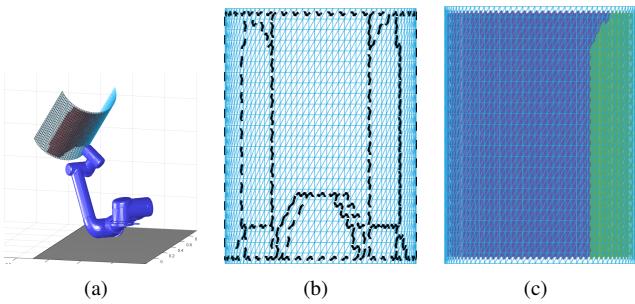


Fig. 13. (a) The manipulator is polishing the top-right corner where can only be polished through kinds of configurations. (b) The initial topological graph of this problem. (c) One optimal solution which requires 1 lift-off.

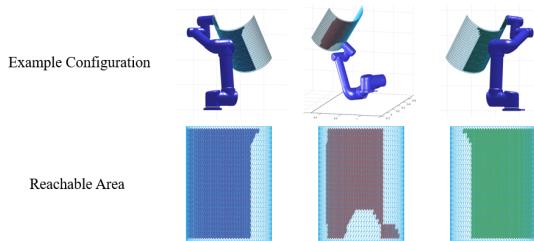


Fig. 14. Example of three different kinds of configurations and the distribution of their reachable area. Once any configuration belonging to the second kind is mistakenly chosen to cover any area of the surface, sooner or later we have to change to the first one and the third one to finish the full coverage of the surface, which wastes an unnecessary lift-off.

D. Real World Experiments with Presence of Obstacles

In this subsection, we use a manipulator polishing the outer surface of a wok to show a physical coverage path generated based on the proposed cellular decomposition method. The physical coverage path uses simple back and force motions, whose generation is not part of our concern. The concatenation between paths in different cells are created by demonstration. The manipulator is UR5, with its last joint abandoned, equivalent to 5DOF, which is non-redundant. Since the hybrid position/force control is beyond our contribution, we do not involve the real contact.

See Figure 15, the nearest and farthest part of the wok are unreachable. Considering the two points shown in Figure 15(d)(e), the manipulator has to keep its wrist “above” its fore-arm in order to avoid collision between them, which leads to the requirement of both shoulder-left configurations and shoulder-right configurations. The total number of lift-offs is 1. Since each color covers some points which only have one possible color, the solution is definitely optimal. Note that any division keeping the connectivity is optimal, so we may just divide the surface through the middle.

See Figure 16, the manipulator is obstructed by a cylindrical obstacle. Since the obstacle may collide with the upper-arm, fore-arm or the EE, and the wrist may collide with the fore-arm, to avoid all collisions, the least number of lift-offs is 2. Similarly, each color covers some points which only have one possible color, so the solution is also optimal.

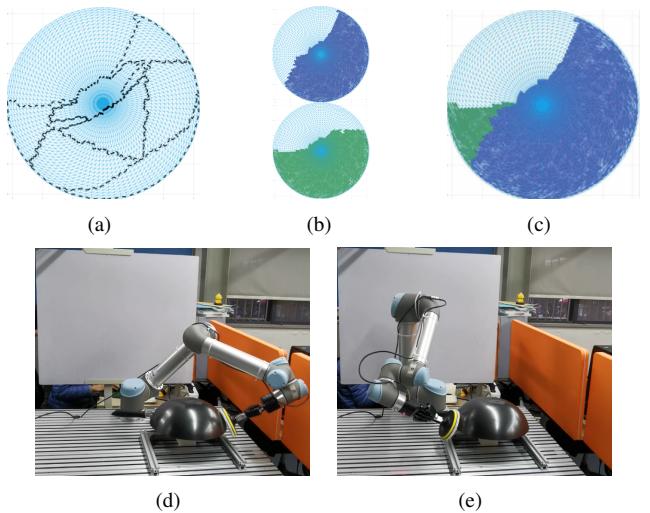


Fig. 15. (a) Initial graph. (b) The reachable area of two kinds of configurations which are closed by the solution in (c). (c) The cutting graph is arbitrary, so we may divide the graph through the middle. (d)(e) Example of the extreme poses of the two kinds of configurations.

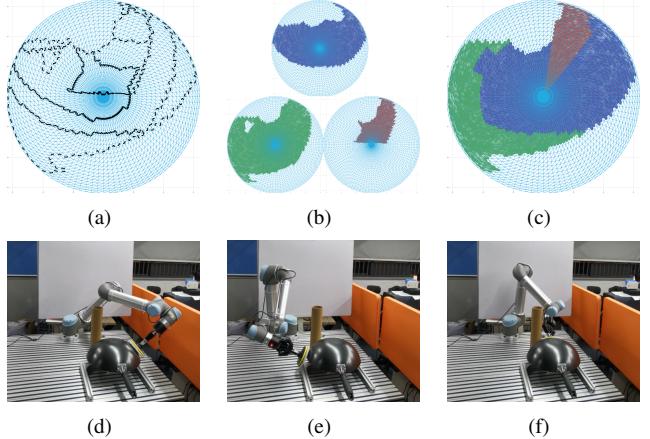


Fig. 16. (a) The initial graph. (b) The reachable area of three kinds of configurations which are closed by the solution in (c). (c) One optimal solution. (d)(e)(f) Example of the three kinds of configurations, where (d) The shoulder-left configurations avoid collision between the upper-arm and the obstacle. (e) The wrist is above the fore-arm so that the wrist will not hit the fore-arm. (f) The only valid configuration to cover this point is putting the wrist below the fore-arm.

VII. CONCLUSION

In this paper, we first prove that the least number of the discontinuities is independent to the choice of the coverage path, thus becomes a criterion evaluating the quality of the placement of the manipulator (or the object), which may contribute to the mobile manipulator or the designing of the assembly line. Then, the proposed algorithm provides a novel cellular decomposition strategy, which, after applying the conventional CPP algorithm in each cell, generates the result coverage path containing the least number of discontinuities, which is verified through simulated and real-world experiments. Also, as a direct corollary, applied to the result of other cellular decomposition methods, the proposed algorithm can tell the user the least number of discontinuities obeying the given cellular decomposition.

Because of the transition strategy which is more complicated than usual movement for coverage required by the discontinuities, and due to the unreachability of the optimal coverage path, an NP problem, the optimality criterion of the coverage path that ensuring least number of discontinuities becomes more significant, and reducing the number of discontinuities is practical to reduce the cost of the coverage task, which is solvable through the proposed algorithm.

In the future, a more complicated situation will be considered, where the initial topological graph contains ring-like cells, which can be broken during the iterative solving process. And the real contact process will be evolved to quantize the energy saving.

ACKNOWLEDGMENT

The authors would like to thank...



Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.