Searching Method for Holding Points during Robotic Food Cutting

Chenyu Dong, Masaru Takizawa, Kohei Kimura, Takashi Suehiro, Shunsuke Kudoh
The Graduate School of Informatics and Engineering
The University of Electro-Communications, Tokyo, Japan
tou@robo.lab.uec.ac.jp, takizawa@robo.lab.uec.ac.jp, kimura@uec.ac.jp, suehiro@is.uec.ac.jp, s-kudoh@uec.ac.jp

***Abstract*—Food cutting is an essential skill of the housekeeping robot. The cutting motion also requires a stable holding. In this paper, we proposed a search method for finding a stable holding point set during the cutting motion. We analyzed the fracture and friction forces caused by the knife at a pose. Additionally, we discuss the calculation of the holding force corresponding to the holding point set to resist the knife wrench. Then, we use an evaluation function to score the holding point set by the magnitude and direction of the holding force. Finally, we search for the holding point set with the highest score as the final output. We evaluated the method on our dual-arm robot system with a two-finger hand. The results indicate that our holding point set can perform stable holding in the cutting situation.**

***Index Terms*—Dual Arm Manipulation, Domestic Robotics**

I. Introduction

Housekeeping robots have advanced significantly in recent years. In various housekeeping tasks, cooking is necessary for everyone in their daily lives. Cutting usually occurs when people process food before cooking.

The holding of food is a crucial motion for successful cutting. The first reason is that holding motions can prevent an unintentional movement caused by a knife and help the cutting motion more precisely. When the knife begins to contact the food, it starts from the point of contact. However, the cutting force applied to that contact point is often not through the center of mass of the food, producing an unintentional torque. Also the friction force between foods and the cutting board may not be sufficient to resist unintentional movements. Holding food can also help cut it more efficiently. For humans, we often cut hard foods by sliding the knife forward and backward instead of simply pushing the knife down with a large force. This slide motion can help to produce cutting cracks more easily. According to D. Zhou's [1], [2] research, the sliding motion can make the cutting motion more efficient than a simple push-down one. Without holding, the slide motion can move food by the knife wrench.

There are many difficulties in performing holding tasks in a food-cutting situation. First, foods have various shapes, and it is difficult to search for a stable holding method (combination of holding positions and finger forces) using a simple shape (such as a cylinder or sphere). Second, evaluating for a holding method is also a critical problem. As many factors need to be considered for a successful holding, a metric is needed to properly evaluate a holding method in order to select an appropriate holding method from many candidates.

There are some factors that should be considered while holding a food:

1) Finger position constraints: If the knife and fingers are too close, the knife's trajectory may be hindered during the cutting process. In the other hand, if too far, greater force is necessary to counteract the torque exerted by the knife on the food.
2) Dynamic force variations: As shown in [1], [2], the sliding knife motion has been shown to reduce the required cutting forces compared to simple vertical pressing. However, it also introduce time-varying forces on foods.
3) Force redundancy: When decomposing the force exerted by the knife into contact forces at multiple finger points, there exist infinitely many sets of force solutions that satisfy the static equilibrium conditions. An efficient search is required to select the most appropriate solution from these possibilities.

In this paper, we proposed an offline method that searches for a holding point set (a collection of contact positions for all fingers, with one position per finger) and can hold food stably through a given knife's trajectory. More specifically, we propose an algorithm that searches for a holding point set that can most effectively resist time-varying forces from a knife, using the shape of the food (3D point cloud model), the friction coefficient, the center of mass, and the knife trajectory as inputs. The main contributions of this study are summarized as follows:

1) We developed an offline method to realize a holding task that considers the entire trajectory of the knife.
2) We designed a new evaluation function for holding point sets, which integrates force and position to enable efficient searching of appropriate sets.
3) We designed a new method to decompose the knife's force into finger forces, addressing the issue of force redundancy. This method can generate appropriate finger forces (such as ensuring all forces point toward the interior of the food) with high probability, thereby accelerating the search for effective solutions more efficiently.

The rest of this paper is organized as follows: The next section introduces the related researches. Section III will discuss the entire method flow, problem definition and some assumptions in this paper. The section IV and V introduce

the essential parts of our proposed method: How to reduce the searching range and the evaluation functions. The section VI describes the details of these experiments and shows the results for each experiment.

II. Related works

In previous studies, "cutting motion" and "grasping motion" have been treated as separate research topics. Research on cutting tasks primarily focuses on the principles of cutting mechanisms or motion generating, with little consideration given to the fixation of the object being cut. On the other hand, grasping research often addresses stability in general scenarios such as lifting or moving objects, and there are few studies that consider special conditions like external forces applied during cutting tasks.

For research on cutting: Zeng [3] and Jung [4] proposed blade control methods (trajectory tracking) during material processing. Here, the material was fundamentally fixed by clamps during processing. Zhou [1], [2] and Atkins [5] analyzed the stress distribution during cutting tasks and demonstrated that "slicing while moving the blade is easier for cutting than pushing it straight down." In recent years,many methods based on machine learning have emerged. Mitsioni [6] used a long short-term memory (LSTM) network to predict the system's state and performed velocity control of a robot arm via model predictive control. Cutting of soft materials has also been studied: Long [7] and Hang [8] developed a control algorithms of adapting to material deformations based on camera tracking. Mu [9], [10] analyzed the fracture force and friction force of a knife during cutting tasks, utilized information from force sensors attached to the robot arm's wrist to predict the mechanical parameters of food materials, and performed cutting tasks along optimized cutting trajectories.

For research on grasping: Considerable work has been done on the mechanical analysis of grasping. Nguyen [11] discussed the classification of finger-object contacts and grasping representationand proposed a grasping method based on force closure in the planar space. Yoshikawa [12] proposed "manipulative force" and "grasping force", using them to analyze object manipulation with a three-fingered hand. Miller [13] developed an algorithm to search for hand joint configurations for grasping based on object shapes. Mirtich [14] proposed a method to simplify the calculation of force-closure grasps. Donner [15] introduced a new physically plausible method for decomposing fingertip forces by formulating the problem as a convex optimization problem with several constraints. Additionally, machine learning based grasping methods have become an active research area in recent years. Pelossof [16] performed grasping tasks using a support vector machine trained on simulation-generated training data. Murali [17] and Lou [18] proposed networks that directly generate hand joint configurations for grasping from images. The authors implemented instance segmentation using Mask-RCNN and Siamese-NN, and used two networks to predict reachability and grasping stability considering collisions. Ultimately, they achieved stable grasps.

In contrast to these prior studies, our research integrates the traditionally separate research areas of "cutting" and "grasping," establishing a method to prevent the movement of food during cutting tasks, which is a key contribution of this study.

III. Methodology

A. Problem formulation

Let the surface shape of the food be denoted by Ω. The desired cutting procedure for this food is specified by the knife trajectory. The state of the knife at a given time is represented by the pair consisting of its pose T and velocity $\boldsymbol{v}$, and the trajectory of the knife is expressed as a sequence of states $\mathbb{T} = \{(T_1, \boldsymbol{v}_1), (T_2, \boldsymbol{v}_2), \ldots\}$. Here, each pose T_i is defined in the knife's local coordinate frame, whose origin is placed at the heel of the blade and whose x-axis is aligned with the blade direction while the y-axis is aligned with the normal of the blade's central plane, as shown on the left of Fig. 1. The goal of this study is to determine an appropriate holding point set $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ for n fingers by given the food surface Ω and the knife trajectory $\mathbb{T}$. Each $\boldsymbol{p}_i$ is a 3-D vector that represents the holding position of the i-th finger.

In addition, the following assumptions are made for the food, the knife, and the cutting board:

1) The motion of the food is limited to translation in the xy-plane and rotation about the z-axis, 3 degrees of freedom.
2) The food and the cutting board are in stable planar contact (e.g., a half cut potato on a board).
3) The contact between the fingers and the food is point contact, means only pushing force and friction force are generated, and no torque arises from friction.
4) The center of mass of the food C_{o} is known.
5) The trajectory of knife is within the cutting plane. In other words, the task involves planar cutting rather than sculpting the food.
6) When the knife creates new cut surfaces Ω_{c_1} and Ω_{c_2} (right of Fig. 1), the pressure applied to these surfaces is uniformly distributed.

B. Method Flow

The overall procedure of our proposed method is shown in Algorithm 1: First, we randomly generate many holding point set $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ on the food surface Ω and collect them into the set $\mathbb{P}_{\mathrm{all}}$. Next, we create a dictionary $\mathbb{S}$ whose keys are the elements of $\mathbb{P}_{\mathrm{all}}$ and values are their evaluation scores. We then construct a dictionary $\mathbb{S}$ whose keys are the holding point set in $\mathbb{P}_{\mathrm{all}}$ and whose associated values storage their respective evaluation scores. All scores are initialized to 0.

The main loop in Algorithm 1 executes these steps:

First, for each element $(T, \boldsymbol{v})$ of the knife trajectory $\mathbb{T}$, we divide the food surface Ω into three small parts: two contact surfaces $\Omega_{\mathrm{c}1}$ $\Omega_{\mathrm{c}2}$ and the holding part Ω_{g}. The contact surfaces $\Omega_{\mathrm{c}1}$ and $\Omega_{\mathrm{c}2}$ are calculated by the shapes of the knife and the blade, as well as the food surface. As for determining the holding part Ω_{g}, in common food cutting tasks, the knife

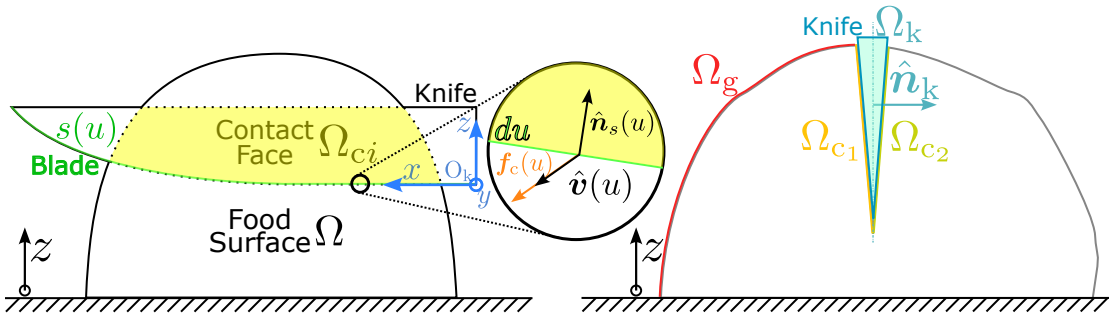


Fig. 1. Left: The zoomed-in part shows the calculation of the fracture force caused by an infinitesimal element du (in green text and line). Right: The crack generated by knife creates two contact faces (Ω_{c_1}, and Ω_{c_2}). Also, the searching only use one side (the red curve) of the food model.

Algorithm 1 Full progress

1: N: the number of elements in $\mathbb{P}$
2: Ω: A surface of food.
3: $\mathbb{T}$: Trajectory of knife, a sequence of pose T and velocity $\boldsymbol{v}$
4: $\mathbb{S}$: Map for final evaluation score of a holding point set P. All scores initialize with 0.
5: $\mathbb{P}_{\text{all}}$: All possible holding point set P from Ω_{g}.
6:
7: **for all** $(T, \boldsymbol{v}) \in \mathbb{T}$ **do**
8: $\quad \Omega_{\text{c1}}, \Omega_{\text{c2}}, \Omega_{\text{g}}, \mathbb{P}_{\text{init}} \leftarrow$PREPAREDATA($\Omega$, T)
9: $\quad \mathbb{P} \leftarrow$FILTERBYGEOSCORE($\mathbb{P}_{\text{init}}$, T)
10: $\quad$ **for** $P \in (\mathbb{P}_{\text{all}} \setminus \mathbb{P})$ **do**
11: $\quad\quad \mathbb{S}[P] \leftarrow -\infty$
12: $\quad$ **end for**
13: $\quad \{S_{\text{pos,i}}\} \leftarrow$ CALPOSITIONALSCORE($\mathbb{P}$, T)
14: $\quad \boldsymbol{t}_{\text{k}} \leftarrow$CALKNIFEFORCE($\Omega_{\text{c1}}$, Ω_{c2}, T, $\boldsymbol{v}$)
15: $\quad \{S_{\text{force,i}}\} \leftarrow$ CALDYNAMICSCORE($\boldsymbol{t}_{\text{k}}, \mathbb{P}$, Ω_{g})
16: $\quad$ **for** $i \leftarrow 1, N$ **do**
17: $\quad\quad \mathbb{S}[P_i] \leftarrow \mathbb{S}[P_i] + S_{\text{pos},i} + S_{\text{force},i}$
18: $\quad$ **end for**
19: **end for**
20: $P_{\text{fin}} \leftarrow P$ such that the value $\mathbb{S}[P]$ is the maximum.

divides the food into two parts: one for holding and the other to be cut off. We follow this approach and calculate Ω_{g} from the food surface Ω, by using the contact surfaces Ω_{c1}, Ω_{c2}, and the knife plane Ω_{k} to divide the food surface, and usually, the part with a larger area is selected as the holding part. (Adding details of dividing the holding part Ω_{g})

The holding point sets in $\mathbb{P}_{\text{all}}$ that lie on Ω_{g} constitute the search space $\mathbb{P}$. Then, the function FILTERBYGEOSCORE in Algorithm 2 removes inappropriate elements from $\mathbb{P}$. Details of this reduction are given in Section IV. In short, positions eliminated here violate low-computational cost geometric constraints, such as excessive proximity between fingers. For every holding point set in $\mathbb{P}_{\text{all}} \setminus \mathbb{P}$, the corresponding score in $\mathbb{S}$ is set to $-\infty$. A holding point set whose score is set to $-\infty$ will never be selected as the final holding point set, even if it reappears in $\mathbb{P}$ in later iterations (lines 10–12 of Algorithm 1).

Next, the positional scores $\{S_{\text{pos,i}}\}$ for every holding point set in $\mathbb{P}$ are calculated by the function CALPOSITIONALSCORE. Subsequently, the force and torque $\boldsymbol{t}_{\text{k}}$ exerted by the knife on the food are computed. $\boldsymbol{t}_{\text{k}}$ is a 6-D vector that concatenates the force and torque components. For each holding point set P_i in $\mathbb{P}$, the function CALDYNAMICSCORE uses $\boldsymbol{t}_{\text{k}}$ to compute the dynamic scores $\{S_{\text{force,i}}\}$. Finally, for each holding point set P, the positional score $\{S_{\text{pos,i}}\}$ and dynamic score $\{S_{\text{force,i}}\}$ are summed and accumulated in $\mathbb{S}[P]$ (loop in lines 16–18 of Algorithm 1).

Finally, after the loop terminates, the holding point set with the highest score in $\mathbb{S}$ is selected and returned as the final holding point set P_{fin} (line 20 of Algorithm 1).

The symbols used in the above descriptions are summarized in Table III-B. finger position

TABLE I
LIST OF SYMBOLS

Symbol	Name	Description
n	Number of fingers	Number of contact points
$\boldsymbol{p}_i$	Finger position	xyz coordinates of the i-th contact point
P	Holding point set	Set of n contact points $\boldsymbol{p}$
$\mathbb{P}$	Search space	Set of holding point sets P
$\boldsymbol{t}_{\text{k}}$	knife wrench	6-D force and torque applied to the food
$\boldsymbol{f}_{\text{f}i}$	Finger tip force	3-D Force vector of the i-th finger tip
$\boldsymbol{f}$	Holding force	Vector of tip forces of n fingers arranged as a column, a $3n$-D vector
Ω		Surface shape of the food
$\Omega_{\text{c1,2}}$		Contact surfaces (right part in Fig.1)
Ω_{g}		Surface part of the food used for holding

IV. SEARCH SPACE REDUCTION VIA GEOMETRIC CONSTRAINTS

This section describes a method to reduce inappropriate elements from a given search space based on geometric constraints. This corresponds to the function CALGEOSCORE in Algorithm 1. In this paper, the evaluation score of each holding point set is calculated by an evaluation function described in Section.V, but the calculation involves force computations with relatively high computational cost. Therefore, the purpose of reducing the search space by using only geometric constraints with low computational cost before calculating the evaluation score.

The specific flow of the function CALGEOSCORE is shown in Algorithm 2. For each element P in inputed search space $\mathbb{P}$,

Algorithm 2 Recudce the search space

function FILTERBYGEOSCORE($\mathbb{P}$, T)
  N: the number of elements in $\mathbb{P}$
  $P_1, \ldots, P_N$: each element of $\mathbb{P}$
  $w_{\mathrm{fin}}, w_{\mathrm{knf}}, w_{\mathrm{tbl}}$: weights
  L: an empty list
  r: output ratio ($r \in (0, 1]$)
  for $i \leftarrow 1, N$ **do**
    $E_{\mathrm{fin,i}} \leftarrow E_{\mathrm{fin}}$ for P_i
    $E_{\mathrm{knf,i}} \leftarrow E_{\mathrm{knf}}$ for P_i
    $E_{\mathrm{tbl,i}} \leftarrow E_{\mathrm{tbl}}$ for P_i
  end for
  $\{E_{\mathrm{fin},i}\} \leftarrow$ NORMALIZE($\{E_{\mathrm{fin},i}\}$)
  $\{E_{\mathrm{knf},i}\} \leftarrow$ NORMALIZE($\{E_{\mathrm{knf},i}\}$)
  $\{E_{\mathrm{tbl},i}\} \leftarrow$ NORMALIZE($\{E_{\mathrm{tbl},i}\}$)
  for $i \leftarrow 1, N$ **do**
    $S_{\mathrm{geo}} \leftarrow w_{\mathrm{fin}} E_{\mathrm{fin},i} + w_{\mathrm{knf}} E_{\mathrm{knf},i} + w_{\mathrm{tbl}} E_{\mathrm{tbl},i}$
    Append (P_i, S_{geo}) to L
  end for
  Sort L by S_{geo}
  $L \leftarrow$ the first $\lfloor rN \rfloor$ elements of L
  return the set of P_i in L
end function

function NORMALIZE($\{E_i\}$)
  N: the number of elements in $\{E_i\}$
  $M \leftarrow \max(E_1, \ldots, E_N)$
  $m \leftarrow \min(E_1, \ldots, E_N)$
  for all $E \in \{E_i\}$ **do**
    $E \leftarrow (E - m)/(M - m)$
  end for
  return $\{E_i\}$
end function

Algorithm 3 Calculate the positional score

function CALPOSITIONALSCORE($\mathbb{P}$, T)
  N: the number of elements in $\mathbb{P}$
  $P_1, \ldots, P_N$: each element of $\mathbb{P}$
  $w_{\mathrm{pdir}}, w_{\mathrm{pdis}}$: weights
  for $i \leftarrow 1, N$ **do**
    $E_{\mathrm{pdir},i} \leftarrow E_{\mathrm{pdir}}$ for P_i
    $E_{\mathrm{pdis},i} \leftarrow E_{\mathrm{pdis}}$ for P_i
  end for
  $\{E_{\mathrm{pdir},i}\} \leftarrow$ NORMALIZE($\{E_{\mathrm{pdir},i}\}$)
  $\{E_{\mathrm{pdis},i}\} \leftarrow$ NORMALIZE($\{E_{\mathrm{pdis},i}\}$)
  for $i \leftarrow 1, N$ **do**
    $S_{\mathrm{pos},i} \leftarrow w_{\mathrm{pdir}} E_{\mathrm{pdir},i} + w_{\mathrm{pdis}} E_{\mathrm{pdis},i}$
  end for
  return $\{S_{\mathrm{pos},i}\}$
end function

the geometric score S_{geo} is calculated, and the pair (P, S_{geo}) is storaged in the list L. After calculating geometric scores for all elements of $\mathbb{P}$, the elements of L are sorted in descending order of S_{geo}, and the top $\lfloor rN \rfloor$ elements (but only holding point set) are returned. Here, N is the number of elements in $\mathbb{P}$, and $r \in (0, 1]$ is a parameter controlling the proportion of elements to be returned.

The geometric score S_{geo} consists of three parts: "distance between fingers", "distance between fingers and the knife," and "height of fingers". The "distance between fingers" part is designed to prevent proximity of fingers. Specifically, if two fingers are too close, they effectively act as a single finger, which negates the purpose of using a multi-finger hand. The equation is defined as follows:

$$E_{\mathrm{fin}} = \min_{\substack{i,j=1,\ldots,n \\ i \neq j}} (D(\|\boldsymbol{p}_i - \boldsymbol{p}_j\|))$$

$$D(x) = \begin{cases} \dfrac{e^{ax/b} - 1}{e^a - 1} & (x \leq b) \\ 1 & (\text{otherwise}) \end{cases}$$

Since a certain distance between fingers is sufficient to prevent their proximity, an upper limit b is set for $D(x)$. When the distance between fingers exceeds b, the value of $D(x)$ is always 1. The parameter a adjusts the curvature of the function within the range $0 \leq x \leq b$, with larger a resulting in greater curvature.

The "distance between fingers and the knife" part evaluates the shortest distance between fingers and the knife to prevent collision, defined as:

$$E_{\mathrm{knf}} = \min_{i=1,2,\cdots,n} (\mathrm{Dist}(\boldsymbol{p}_i, \hat{\boldsymbol{n}}_{\mathrm{k}}, \boldsymbol{p}_{\mathrm{k}}))$$

Here, $\boldsymbol{n}_{\mathrm{k}}$ is the normal vector of the knife plane, and $\boldsymbol{p}_{\mathrm{k}}$ is the position of the knife in the world coordinate system.

The "height of holding positions" part evaluates the lowest finger position (smallest z-coordinate) to prevent collision with the table, defined as:

$$E_{\mathrm{tbl}} = \min(p_{1z}, \ldots, p_{nz})$$

where p_{iz} is the z-coordinate of the i-th finger position $\boldsymbol{p}_i$.

Finally, these three scores are normalized to the range $[0, 1]$ using the function NORMALIZE defined in lines 24–32 of Algorithm 2, and combined with weights $w_{\mathrm{fin}}, w_{\mathrm{knf}}, w_{\mathrm{tbl}}$ to calculate S_{geo}:

$$S_{\mathrm{geo}} = w_{\mathrm{fin}} E_{\mathrm{fin}} + w_{\mathrm{knf}} E_{\mathrm{knf}} + w_{\mathrm{tbl}} E_{\mathrm{tbl}} \tag{1}$$

V. EVALUATION SCORE CALCULATION

This section describes the method for calculating the evaluation score for holding point sets. In Algorithm 1, this calculation mainly involves the functions CALPOSITIONALSCORE and CALDYNAMICSCORE. The function CALPOSITIONALSCORE takes the search space $\mathbb{P}$ and the knife pose as inputs, assigning an evaluation score S_{pos} to each holding point set P in $\mathbb{P}$. The function CALDYNAMICSCORE takes the knife wrench $\boldsymbol{t}_{\mathrm{k}}$, food surface Ω, and a single holding point set P as arguments, calculating the force evaluation score S_{dynamic} for that holding point set. Even when the holding positions are determined, there is redundancy in the holding force applied by the fingers to the food, which means multiple possibles holding force can satisfy equilibrium with the knife wrench. The proposed method considers this redundancy, calculates scores for all possible holding force, and finally uses

the maximum score as the force evaluation score S_{dynamic} for a holding point set.

A. Calculation of positional score S_{pos}

S_{pos} consists of two parts: "direction" and "distance."

"Direction" part evaluates the direction of the line approximating the holding point set and the direction of the knife. This part is designed to ensure that the fingers can resist knife wrench from various directions as much as possible. Specifically, the more the principal component direction of each finger position is parallel to the knife plane, the higher the direction score. Therefore, the direction score E_{pdir} is defined as:

$$E_{\mathrm{pdir}} = 1 - |\mathrm{PCA}(P) \cdot \hat{\boldsymbol{n}}_{\mathrm{k}}|$$

Here, $\mathrm{PCA}(P)$ is a function that calculates the principal component direction vector of P.

"Distance" part evaluates the distance from each finger position to the knife, designed to assess how easily the fingers can resist the knife wrench. As fingers closer to the knife plane are better able to resist the knife wrench, the distance score E_{pdis} is defined as:

$$E_{\mathrm{pdis}} = -\min(\mathrm{Dist}(P, \hat{\boldsymbol{n}}_{\mathrm{k}}, \boldsymbol{p}_{\mathrm{k}}))$$

Here, $\mathrm{Dist}(\cdot)$ is a function that calculates the distance from a point to a plane, taking the point's position, and data to define a plane: the plane's normal vector, and a point on the plane as inputs. In this equation, we use knife's normal vector $\hat{\boldsymbol{n}}_{\mathrm{k}}$ as the plane's normal vector, and the origin point of knife's coordinate as the point $\boldsymbol{p}_{\mathrm{k}}$ on the plane. This equation calculates the distance from the holding point set P to the knife plane and outputs the minimum of these distances. A negative sign is applied to ensure that smaller distances result in higher scores.

Finally, these two scores are normalized to the range $[0, 1]$ using the function NORMALIZE, and combined with weights w_{pdir} and w_{pdis} (lines 9–13 of Algorithm 3) to calculate S_{pos}:

$$S_{\mathrm{pos},i} = w_{\mathrm{pdir}} E_{\mathrm{pdir},i} + w_{\mathrm{pdis}} E_{\mathrm{pdis},i} \tag{2}$$

B. Calculation of knife wrench

This section describes the calculation of the knife wrench used in the Section.V-C. In this study, we use a method based on Mu [9] to compute the knife wrench. The knife wrench is decomposed into two components: fracture force (the force required to create cracks in the material) and friction force (the frictional force between the food and the knife), which are considered separately. Each component is calculated as follows:

1) Fracture force: The blade curve is parameterized as $s(u)$, where $\hat{\boldsymbol{n}}_s(u)$ denotes the blade normal vector at blade position u, and the velocity and its unit direction vector are written as $\boldsymbol{v}(u)$ and $\hat{\boldsymbol{v}}(u)$, respectively (left in Fig. 1). The fracture toughness of the food is denoted κ, the fracture force $\boldsymbol{f}_{\mathrm{c}}(u)$ at position u is defined as:

$$\boldsymbol{f}_{\mathrm{c}}(u) = -\kappa \left(\hat{\boldsymbol{v}}(u) \cdot \hat{\boldsymbol{n}}_s(u)\right) \hat{\boldsymbol{v}}(u)$$

By integrating $\boldsymbol{f}_{\mathrm{c}}(u)$ over the entire blade curve, the total fracture force $\boldsymbol{f}_{\mathrm{c}}$ applied to the food is calculated as:

$$\boldsymbol{f}_{\mathrm{c}} = \int \boldsymbol{f}_{\mathrm{c}}(u)\, \mathrm{d}u$$

Additionally, using the food's center of mass $\boldsymbol{g}$, the torque $\boldsymbol{\tau}_{\mathrm{c}}$ exerted by the fracture force is computed as:

$$\boldsymbol{\tau}_{\mathrm{c}} = \int (s(u) - \boldsymbol{g}) \times \boldsymbol{f}_{\mathrm{c}}(u)\, \mathrm{d}u$$

2) Friction force: Under the assumption (6) stated in Section III-A, this paper assumes that the pressure applied to the contact surfaces $\Omega_{\mathrm{c}j}$ $(j = 1, 2)$ is uniform, denoted as P. For the knife's velocity $\boldsymbol{v}$, let $\boldsymbol{v}_{\mathrm{c}j}$ be the velocity component parallel to the contact surface $\Omega_{\mathrm{c}j}$. Given the normal vector $\hat{\boldsymbol{n}}_{\mathrm{c}j}$ of contact surface $\Omega_{\mathrm{c}j}$, this velocity component is calculated as $\boldsymbol{v}_{\mathrm{c}j} = \boldsymbol{v} - (\boldsymbol{v} \cdot \hat{\boldsymbol{n}}_{\mathrm{c}j})\, \hat{\boldsymbol{n}}_{\mathrm{c}j}$. With μ as the coefficient of friction between the knife and the food, the frictional force is calculated using Coulomb's law as:

$$\boldsymbol{f}_{\mathrm{fr}} = \mu P \sum_{j=1}^{2} \iint_{\Omega_{\mathrm{c}j}} \hat{\boldsymbol{v}}_{\mathrm{c}j}\, \mathrm{d}S,$$

$$\boldsymbol{\tau}_{\mathrm{fr}} = \mu P \sum_{j=1}^{2} \iint_{\boldsymbol{x} \in \Omega_{\mathrm{c}j}} (\boldsymbol{x} - \boldsymbol{g}) \times \hat{\boldsymbol{v}}_{\mathrm{c}j}\, \mathrm{d}S.$$

Here, S represents the area of an infinitesimal element of the contact surface $\Omega_{\mathrm{c}j}$, and $\boldsymbol{x}$ denotes the central position of the infinitesimal element on $\Omega_{\mathrm{c}j}$.

3) knife wrench: Combining the above, the knife wrench $\boldsymbol{t}_{\mathrm{k}}$ is defined as a 6D vector of translational force and rotational moment:

$$\boldsymbol{t}_{\mathrm{k}} = \left[\boldsymbol{f}_{\mathrm{k}}^{\mathrm{T}}, \boldsymbol{\tau}_{\mathrm{k}}^{\mathrm{T}}\right]^{\mathrm{T}} = \left[\left(\boldsymbol{f}_{\mathrm{fr}} + \boldsymbol{f}_{\mathrm{c}}\right)^{\mathrm{T}}, \left(\boldsymbol{\tau}_{\mathrm{fr}} + \boldsymbol{\tau}_{\mathrm{c}}\right)^{\mathrm{T}}\right]^{\mathrm{T}}.$$

In Algorithm 1, the function CALKNIFEFORE executes this calculation.

4) Holding force & Finger force: This section, referred to in Section V-C, presents a method for generating multiple holding forces. Use the previous section's calculation of the knife wrench $\boldsymbol{t}_{\mathrm{k}}$, we establish an equation that relates the holding force and the knife wrench. Using this equation, we propose a new method to generate multiple holding forces.

First, we introduce the concepts of "holding force" and its resultant wrench exerts on the food. Under Assumption (3), we ignore the torques at the fingertips and only consider the forces $\boldsymbol{f}_{\mathrm{f}i}$ $(i = 1, \ldots, n)$ applied by each fingertip. Arranging these forces into a $3n$-dimensional vector $\boldsymbol{f} = [\boldsymbol{f}_{\mathrm{f}1}^{\mathrm{T}}, \ldots, \boldsymbol{f}_{\mathrm{f}n}^{\mathrm{T}}]^{\mathrm{T}}$, we define $\boldsymbol{f}$ as the "holding force". The resultant wrench of the holding force $\boldsymbol{f}$ on the food is denoted as $\boldsymbol{t}_{\mathrm{h}}$ (called as holding resultant wrench below), which consists of the force component $\boldsymbol{f}_{\mathrm{h}}$ and the torque component $\boldsymbol{\tau}_{\mathrm{h}}$, and can be written as $\boldsymbol{t}_{\mathrm{h}} = [\boldsymbol{f}_{\mathrm{h}}^{\mathrm{T}}, \boldsymbol{\tau}_{\mathrm{h}}^{\mathrm{T}}]^{\mathrm{T}}$. Also, the holding resultant wrench $\boldsymbol{t}_{\mathrm{h}}$ should conteract the knife wrench, which satisfied $\boldsymbol{t}_{\mathrm{h}} = -\boldsymbol{t}_{\mathrm{k}}$.

To simplify the analysis under the given assumptions, In this study, under Assumptions (1) and (2), the forces along the z-axis and the torques about the x- and y-axes are assumed to

be canceled out by the cutting board. So, when considering the required holding resultant wrench $\boldsymbol{t}_{\mathrm{h}}$, we only need to take into account the x- and y-components of the force $\boldsymbol{f}_{\mathrm{h}}$ and the z-component of the torque $\boldsymbol{\tau}_{\mathrm{h}}$, which are extracted as $[f_{\mathrm{h}x}, f_{\mathrm{h}y}, \tau_{\mathrm{h}z}]$.

Moreover, the relationship between the finger forces $\boldsymbol{f}$ and the holding force $\boldsymbol{t}_{\mathrm{h}}$ can be written as follows:

$$\boldsymbol{t}_{\mathrm{h}} = G\boldsymbol{f}$$

Here, G is a $6 \times 3n$ matrix called the grasping matrix, which defined as follows:

$$G = \begin{bmatrix} I_3 & \cdots & I_3 \\ R_1 & \cdots & R_n \end{bmatrix} \tag{3}$$

where I_3 is the 3×3 identity matrix, and R_i is a 3×3 skew-symmetric matrix defined by $R_i \boldsymbol{f}_{\mathrm{f}i} = (\boldsymbol{p}_i - \boldsymbol{g}) \times \boldsymbol{f}_{\mathrm{f}i}$. Correspondingly, to align with the reduced holding resultant wrench $[f_{\mathrm{h}x}, f_{\mathrm{h}y}, \tau_{\mathrm{h}z}]$, the grasping matrix G should be considered as the one obtained by removing the 3rd, 4th, and 5th rows. Solving this equation for $\boldsymbol{f}$ gives the following result:

$$\boldsymbol{f} = G^{+}\boldsymbol{t}_{\mathrm{h}} + [I_{3n} - G^{+}G]\,\boldsymbol{k} \tag{4}$$

where G^{+} is the pseudo-inverse of G, and $\boldsymbol{k}$ is an arbitrary $3n$-D vector.

Next, we discuss the method of generating multiple holding forces. Since there exists an arbitrary vector $\boldsymbol{k}$ in Equation (4), even if the holding point set and the knife wrench are determined, there can be infinitely many holding forces that can balance the knife wrench. In contrast, in this study, we decided to adopt a method where a large number of holding forces are generated first, and then the best one is selected from them to calculate the force evaluation score. The simplest way to generate candidate combinations of fingertip forces is to assign random values to $\boldsymbol{k}$ in Equation (4). However, this method has a high probability of generating physically implausible candidates, resulting in very low generation efficiency. Therefore, to generate valid candidates more efficiently, the following method is proposed.

1) Randomly generate an initial fingertip force $\boldsymbol{f}_{\mathrm{init}}$, but limit its direction to the interior of the friction cone.
2) To correctly balance the randomly generated $\boldsymbol{f}_{\mathrm{init}}$ with the knife wrench, make the following adjustments:

$$\boldsymbol{t} = (-\boldsymbol{t}_{\mathrm{k}}) - G\boldsymbol{f}_{\mathrm{init}}$$
$$\boldsymbol{f} = \boldsymbol{f}_{\mathrm{init}} - G^{+}\boldsymbol{t}$$

The function GENERATEFINGERFORCE employs proposed method to generate physically plausible holding forces more efficiently.

C. Calculation of dynamic score S_{dynamic}

Section V-B4 described how multiple valid holding force $\boldsymbol{f}$ are generated a holding point set P. This section utilizes these generated holding force by evaluating them based on specific criteria (magnitude, direction, and variance) to determine the resulting dynamic score S_{dynamic} for a holding point set P.

Algorithm 4 Algo of the holding score calculation

```
1: function CalDynamicScore(t_k, ℙ, Ω_g)
2:     for i ← 1, N do
3:         S_force,i ← CalDynamicScoreEach(t_k, P_i, Ω_g)
4:     end for
5:     return {S_force,i}
6: end function
7:
8: function CalDynamicScoreEach(t_k, P, Ω_g)
9:     L: an empty list
10:    M: the number of finger force to generate
11:    [f_1, f_2, ... f_M] ←GenerateFingerForce(t_k, P)
12:    for i ← 1, M do
13:        E_mag,i ← E_mag for P
14:        E_dir,i ← E_dir for P
15:        E_var,i ← E_var for P
16:    end for
17:    {E_mag,i} ← Normalize({E_mag,i})
18:    {E_dir,i} ← Normalize({E_dir,i})
19:    {E_var,i} ← Normalize({E_var,i})
20:    for i ← 1, M do
21:        S ← w_mag E_mag,i + w_dir E_dir,i + w_var E_var,i
22:        Append S to L
23:    end for
24:    return the maximum element in L
25: end function
```

Specifically, the scores for magnitude E_{mag}, direction E_{dir}, and variance E_{var}, referenced in Algorithm 4 (lines 13-15), are calculated as follows:

The magnitude score E_{mag} is defined such that configurations requiring less finger force receive higher scores, according to the following equation:

$$E_{\mathrm{mag}} = -\sum_{i=1}^{n} \|\boldsymbol{f}_{\mathrm{f}i}\|.$$

The direction score E_{dir} is designed to minimize finger slippage. It assigns higher scores when the finger force direction aligns better with the food surface normal vector at the contact point $\boldsymbol{p}_i$, defined as:

$$E_{\mathrm{dir}} = \sum_{i=1}^{n} \hat{\boldsymbol{n}}_{\Omega}(\boldsymbol{p}_i) \cdot \hat{\boldsymbol{f}}_{\mathrm{f}i}.$$

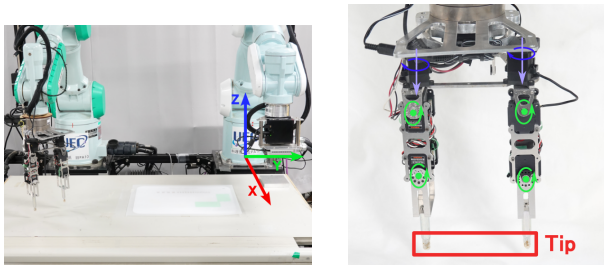
Here, $\hat{\boldsymbol{n}}_{\Omega}(\boldsymbol{p}_i)$ represents the normal vector at position $\boldsymbol{p}_i$ on the food surface Ω.

The magnitude variance score E_{var} encourages uniform distribution of force magnitudes among fingers, defined as:

$$E_{\mathrm{var}} = -\mathrm{Var}(\|\boldsymbol{f}_{\mathrm{f}1}\|, \|\boldsymbol{f}_{\mathrm{f}2}\|, \ldots, \|\boldsymbol{f}_{\mathrm{f}n}\|).$$

Similar to the previous section, these three evaluation metrics are normalized to the range $[0, 1]$ using the NORMALIZE function. They are then combined using weights to calculate the score S for a holding force:

$$S = w_{\mathrm{mag}} E_{\mathrm{mag},i} + w_{\mathrm{dir}} E_{\mathrm{dir},i} + w_{\mathrm{var}} E_{\mathrm{var},i} \tag{5}$$



(a) The workspace and coordinate of robots

(b) The joint configuration of the finger hand

Fig. 2. The left figure shows the entire robot workspace for implementing our method. The robot arms are both attached with a RGB-D camera and a 6-axis force sensor. The left robot arm attaches a fingered type hand, and the right attaches a parallel gripper. The right figure shows the joint configuration of the hand.

Finally, the dynamic score S_{dynamic} for the holding point set P is determined by selecting the maximum value from all scores S calculated for it (Algorithm 4, line 20–24).

VI. Experiments

A. Robot environments

1) Hardwares: We tested our methods with two PA10-7C, a 7-degree-of-freedom arm made by Mitsubishi. They were set at a height of 700 mm from the ground and separated by a distance of 804 mm. Each robot arm has a gripper attached to it. The left robot arm in Figure 2a is a fingered hand-type gripper that is used to hold food during the experiment. The right robot arm in Figure 2a is a parallel gripper with large closing power to rigidly attach the knife to the robot. A Realsense D435 (RGB-D camera), manufactured by Intel, was installed at the front of the parallel gripper to grab the 3D surface information of the food.

For the hand-type gripper, we used a 3-axis two-fingered servo hand. The servo is a FUTABA RS405CB, which has a 48.0 $[\mathrm{kgf \cdot cm}]$ max torque and 0.1 [deg] angular accuracy. The configuration of each finger was a z-y-y-tip, as shown in Figure 2b. To control the fingertip's position, we used a Jacobian-based method to calculate the finger-joint only inverse kinematics (IK).

2) Surface Acquisition: First, we acquire the food surface Ω. The food surface shape Ω is acquired as a point cloud using an RGB-D camera. The robot arm adjust the camera to surrounding the food and transform each point cloud into the world coordinate system (shown in Figure 2a) to merge them. The merged point cloud was denoized using the moving least squares (MLS) method [19] implemented in the point cloud library (PCL) [20]. We downsampled the smoothed point cloud model to approximately 100 points to search the holding points more quickly.

The acquired point cloud Ω using the method described above represents only the food's exterior surface and lacks internal information. Consequently, the data for the contact surfaces Ω_{c_1} and Ω_{c_2}, which are necessary for calculating the knife force is missing. Therefore, these contact surfaces

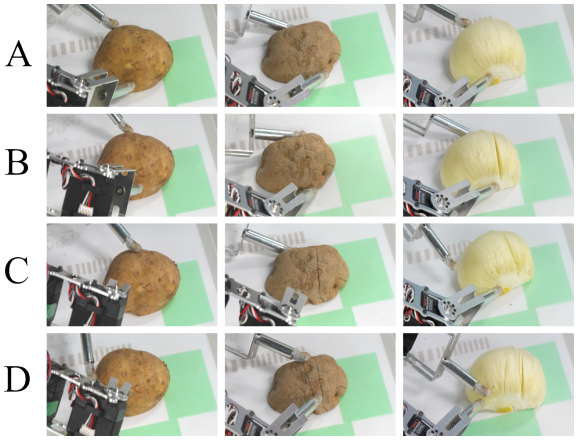


Fig. 3. The images of cutting food stuffs on each type of method (in the row).

TABLE II
The Results of Generating Holding Force

Point Cloud No.	Proposed Method		Random k		Improve %
	Passed	Ratio/%	Passed	Ratio/%	
1	56661961	31.47	6918682	3.84	818.97
2	32693962	18.16	4447361	2.47	735.13
3	71345970	39.63	9037480	5.02	789.45
4	77728717	43.18	10174652	5.65	763.94
5	33053551	18.36	4081123	2.27	809.91
6	77206038	42.89	10616018	5.90	727.26
Avg.	58115033	32.28	7545886	4.19	774.11

are generated computationally as follows: First, based on the known geometry (position and normal vector) of the theoretical contact plane created by the knife cut, points from the acquired surface cloud Ω that lie in proximity to this plane are projected onto it. These projected points collectively form the contour of the contact surface. Next, to fill the area inside this contour with points, a polygon is generated from the contour points using the Euclidean Minimum Spanning Tree (EMST), and then point filling is performed within this polygon. Finally, a mesh representing the contact surface is created by applying the Delaunay triangulation method to the combined set of contour points and the filled interior points.

B. Experiment of generating holding force

This section describes an experiment conducted to verify the effectiveness of the proposed holding force generation method (detailed in Section V-B4). We compare the proposed method against a random generation approach by measuring the proportion of generated forces that lie within the friction cone, hereafter referred to as the 'success rate'.

First, we prepared point cloud data for 6 different potatoes and a single knife trajectory capable of cutting all of them. Calculations were performed according to Algorithm 1 using this data. During the execution of the CalDynamicScore function (line 15 of Algorithm 1), specifically at the step where holding forces are generated (line 11 of Algorithm 4),

we compared two approaches: using the proposed GENERATEFINGERFORCE function versus using a random generation method (assigning random values to $\boldsymbol{k}$ in Eq. (4) to generate holding forces). For each approach, 1.2×10^5 holding force were generated per trial. We counted the number of these vectors that fell within the friction cone and calculated the success rate for each method. The friction cone was defined with its axis aligned with the inward normal vector at the finger position and an apex angle of 40 degrees. Furthermore, in this experiment, the size of the search space returned by the FILTERBYGEOSCORE function was fixed to the top 1500 candidates, rather than the $\lfloor rN \rfloor$ fraction specified in Algorithm 2.

The results of this experiment are presented in Table II. Each row in the table corresponds to a different potato point cloud. 'Passed' indicates the number of generated holding force vectors that lie within the friction cone, and 'Ratio' represents this count as a percentage of the total number generated for that trial (1.2×10^5). 'Improve' quantifies the increase in success rate achieved by the proposed method compared to the random method, expressed as a percentage. These results show that, on average, the proposed method achieved a success rate 7.7 times higher than the random generation method.

C. Experiment of real robot holding

In this experiment, we validated the effectiveness of the proposed method by performing cutting tasks on potatoes and onions while holding them using the proposed approach and three other comparison methods. We used 6 half-cut potatoes and 4 half-cut onions (ensuring a flat bottom surface for stability). Each food item was cut 4 times using the proposed method and each of the three comparison methods. The physical parameters for potato and onion were determined based on the method presented by Mu et al. [9]. The measured values used were: fracture toughness $409.66\,[\mathrm{J/m^2}]$ and $646.46\,[\mathrm{J/m^2}]$, and pressure distribution $3352.07\,[\mathrm{N/m^2}]$ and $3971.2\,[\mathrm{N/m^2}]$, for potato and onion respectively.

The weights used in the evaluation score calculations were set as follows:

- For the geometric evaluation score (Eq. (1)): $w_{\mathrm{fin}} = 1$, $w_{\mathrm{knf}} = 4.4$, and $w_{\mathrm{tbl}} = 6$.
- The output ratio r for Algorithm 2 was set to 0.6.
- For the positional evaluation score (Eq. (2)): $w_{\mathrm{pdir}} = 5$ and $w_{\mathrm{pdis}} = 4$.
- For the force candidate evaluation score (Eq. (5)): $w_{\mathrm{dir}} = 2$, $w_{\mathrm{mag}} = 2$, and $w_{\mathrm{var}} = 1$.

Three comparison methods (referred to as Methods **B**–**D**) were used alongside the proposed method (Method **A**):

- **Method A**: The proposed method, selecting the holding point set with the highest overall evaluation score.
- **Method B** and **Method C**: Selected holding point sets with medium and the lowest overall evaluation scores, respectively. Note that these selections were made only from the top 60% of candidates remaining after the geometric filtering step (Algorithm 2 with $r = 0.6$).
- **Method D**: Selected a holding point set randomly from the search space that before geometric filter (means from $\mathbb{P}_{\mathrm{all}}$).

A reciprocal (back-and-forth) slicing trajectory was used for the knife. The trajectory consisted of two phases: first, a constant velocity linear movement forward and downward at a 25-degree angle relative to the cutting board; second, a constant velocity linear movement backward and downward, also at 25 degrees. The transition between these phases occurred when the knife reached half the initial contact height on the food item.

The cutting plane for each trial was specified manually by a human operator according to two criteria: positioned towards the right side of the food (to ensure a sufficiently large area for holding) and ensuring a cutting height of at least 3 cm (to accommodate a long knife trajectory). For comparison purposes, each food needed to be cut four times (once per method A–D). However, repeatedly cutting the exact same location drastically reduces the required cutting force after the first cut. To mitigate this, after each cut, the planned cutting plane for the subsequent trial on the same food was shifted 2 mm along its normal vector.

The stability of the hold was evaluated by comparing images of the food taken before and after each cutting trial using a fixed camera (referred to as evaluation photos). Specifically, the translation (in pixels) and rotation (in degrees) of the food were measured by mapping the 'after' image to the 'before' image.

The overall experimental procedure was as follows:

1) Place the food (half-cut potato or onion) on the cutting board.
2) Capture the point cloud of the food to acquire its shape Ω.
3) Select a holding point set using one of the methods (**A**–**D**).
4) For the selected holding point set, perform the following steps:
    a) Capture the 'before' evaluation photo.
    b) Grasp the food at the selected holding point set.
    c) Perform the cutting motion.
    d) Capture the 'after' evaluation photo.

The experimental results are shown in Table III. Methods **A** and **B** yielded better results than Methods **C** and **D**, despite some outliers. This difference is attributed to frequent grasp failures with Methods **C** and **D**, often resulting in significant displacement of the food during the cutting process. Figure 3 illustrates the holding point sets in the experiment. Visually, there appears to be little significant difference in the holding point set outcome between Method **A** (highest score) and Method **B** (medium score). We attribute this to the effectiveness of the search space reduction (Section.IV); since many clearly unsuitable holding point set candidates were eliminated early, the remaining candidates with high-to-medium evaluation scores likely represented generally stable grasp configurations. Consequently, selecting among these well-

TABLE III
TRANSLATION (IN PIXEL) & ROTATION (IN DEGREE) AFTER CUTTING. THE POTATO CUT 6 TIMES, ONION CUT 4 TIMES.

Holding Type	Food	Data Type	Avg.	Std.	Oringin Data					
					1	2	3	4	5	6
A	Potato	T	113.85	93.13	259.40	47.00	197.70	61.50	28.20	89.30
		R	3.48	4.30	0.00	0.00	3.53	0.00	7.65	9.70
	Onion	T	84.45	117.06	14.40	14.80	50.40	258.20	—	—
		R	0.99	1.99	0.00	0.00	0.00	3.97	—	—
B	Potato	T	230.03	290.15	646.00	24.00	12.30	36.70	106.50	554.70
		R	5.28	6.36	13.12	0.00	0.00	0.00	5.73	12.84
	Onion	T	41.03	16.49	48.50	35.30	59.20	21.10	—	—
		R	0.00	0.00	0.00	0.00	0.00	0.00	—	—
C	Potato	T	165.07	191.30	526.20	76.40	114.50	21.70	222.00	29.60
		R	3.85	5.54	14.50	0.00	2.14	5.06	1.40	0.00
	Onion	T	142.90	109.59	289.80	145.90	108.00	27.90	—	—
		R	4.01	5.16	11.45	3.43	1.15	0.00	—	—
D	Potato	T	407.50	198.05	759.50	304.70	263.70	520.30	345.50	251.30
		R	7.69	7.31	16.92	1.62	3.34	8.39	15.87	0.00
	Onion	T	429.70	202.56	160.40	403.30	633.00	522.10	—	—
		R	6.22	3.38	1.28	7.25	7.39	8.96	—	—

rated candidates (whether highest-scoring like Method A or medium-scoring like Method B) appears to yield similarly stable holding results. Conversely, Method **C** (using low-scoring holding point sets) and Method **D** (using randomly selected holding point sets) resulted in significantly reduced holding stability. This result indicates that the proposed methodology, which prioritizes higher scores , effectively identifies more stable holding point sets.

However, outliers (instances of poor stability) were observed even with the generally successful Methods **A** and **B**. Further investigation suggested that these failures were less likely due to issues with the core selection logic, but rather stemmed from noise and errors in perception and control, as detailed below:

1) Errors in the food point cloud: Residual noise in the point cloud acquired by the RGB-D camera could not be completely eliminated, leading to discrepancies between the geometric model used in calculations and the actual shape of the food item.
2) Fingertip position control errors: The PID position control for the servos used in the hand incorporated a drive current deadband to prevent vibrations (set to the manufacturer's default of 13% of maximum current in this experiment). This deadband introduced position control inaccuracies at the servo level, which, given the kinematics of the robot hand, could amplify to fingertip position errors of up to approximately 1 cm.

Another limitation relates to the evaluation method itself. As evaluation photos were captured only before and after the cutting task, the stability during the cutting process could not be directly assessed. It was observed that even if a grasp was unstable and the food moved significantly during cutting, the reciprocal motion of the knife could potentially return the food close to its original position by the end of the task. In such cases, our evaluation based solely on pre- and post-cutting images would erroneously indicate small displacement. Some of the seemingly better results observed for Methods **C** and **D** might be attributable to this phenomenon. Evaluating holding stability throughout the entire cutting process remains a topic for future work.

VII. CONCLUSION

In this research, we proposed a method for searching appropriate holding point sets for robotic food cutting tasks. First, we described the calculation methods for the knife wrench and the corresponding finger forces needed to counteract it. Subsequently, we proposed a method to generate candidate holding point sets based on these calculations, along with evaluation functions to assess them. The proposed methodology was implemented on a dual-arm robot system equipped with a two-fingered hand, and experiments were conducted. The results from cutting potatoes and onions using the proposed method and three comparison approaches showed that stable holds were achieved using the proposed method (which selects the highest-scoring holding point set) and methods employing candidates with relatively high scores from the evaluation functions, thus demonstrating the effectiveness of our proposed approach.

Future work will focus on extending the method beyond the current assumption of a flat bottom surface for the food item, which primarily limits movement to 2D on the cutting board; achieving stable holds for non-flat or near-spherical items capable of full 3D motion presents a significant challenge. Furthermore, as mentioned in the discussion, improving the evaluation methodology to enable the assessment of holding stability throughout the entire cutting process, not just pre- and post-cut states, is another important task for the future.

REFERENCES

[1] D. Zhou, M. Claffee, K.-M. Lee, and G. McMurray, "Cutting, "by pressing and slicing", applied to robotic cutting bio-materials. i. modeling of stress distribution," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2896–2901.

[2] ——, "Cutting, 'by pressing and slicing', applied to the robotic cut of bio-materials. ii. force during slicing and pressing cuts," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2256–2261.

[3] G. Zeng and A. Hemami, "An adaptive control strategy for robotic cutting," in *Proceedings of International Conference on Robotics and Automation*, vol. 1, 1997, pp. 22–27 vol.1.

[4] S. Jung and T. Hsia, "Adaptive force tracking impedance control of robot for cutting nonhomogeneous workpiece," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 3, 1999, pp. 1800–1805 vol.3.

[5] A. Atkins, X. Xu, and G. Jeronimidis, "Cutting, by 'pressing and slicing,' of thin floppy slices of materials illustrated by experiments on cheddar cheese and salami," *Journal of Materials Science*, vol. 39, pp. 2761–2766, 01 2004.

[6] I. Mitsioni, Y. Karayiannidis, and D. Kragic, "Modelling and learning dynamics for robotic food-cutting," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 1194–1200.

[7] P. Long, W. Khalil, and P. Martinet, "Force/vision control for robotic cutting of soft materials," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4716–4721.

[8] L. Han, H. Wang, Z. Liu, W. Chen, and X. Zhang, "Vision-based cutting control of deformable objects with surface tracking," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2016–2026, 2021.

[9] X. Mu, Y. Xue, and Y.-B. Jia, "Robotic cutting: Mechanics and control of knife motion," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3066–3072.

[10] X. Mu and Y.-B. Jia, "Physical property estimation and knife trajectory optimization during robotic cutting," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2700–2706.

[11] V.-D. Nguyen, "Constructing force-closure grasps," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, 1986, pp. 1368–1373.

[12] T. Yoshikawa and K. Nagai, "Manipulating and grasping forces in manipulation by multifingered robot hands," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 1, pp. 67–77, 1991.

[13] A. Miller, S. Knoop, H. Christensen, and P. Allen, "Automatic grasp planning using shape primitives," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, 2003, pp. 1824–1829 vol.2.

[14] B. Mirtich and J. Canny, "Easily computable optimum grasps in 2-d and 3-d," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 739–747 vol.1.

[15] P. Donner, S. Endo, and M. Buss, "Physically plausible wrench decomposition for multieffector object manipulation," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1053–1067, 2018.

[16] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 4, 2004, pp. 3512–3518 Vol.4.

[17] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *ICRA 2020 (virtual)*, June 2020.

[18] X. Lou, Y. Yang, and C. Choi, "Collision-aware target-driven object grasping in constrained environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6364–6370.

[19] P. Lancaster and K. Salkauskas, "Surfaces generated by moving least squares methods," *J. Math. Comput.*, vol. 37, pp. 141–155, 01 1981.

[20] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI - Künstliche Intelligenz*, vol. 24, pp. 345–348, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:3345029