# Blades: A New Class of Geometric Primitives for Feeding 3D Parts on Vibratory Tracks

Onno C. Goemans*, Ken Goldberg‡ and A. Frank van der Stappen*
*Institute of Information and Computing Sciences, Utrecht University, PO Box 80089, 3508 TB Utrecht, The Netherlands
‡Department of Industrial Engineering and Operations Research, University of California at Berkeley, Berkeley, CA 94720, USA

***Abstract*— The vibratory bowl feeder remains the most common approach to the automated feeding (orienting) of industrial parts. We study the algorithmic design of devices on the bowl feeder track that filter out all but one orientation of a given polyhedral part. In this context, we propose a simple new primitive, consisting of one horizontally mounted convex polygonal metal "blade", that can feed a broad class of three-dimensional parts by reorienting and rejecting all but those in a desired orientation. This powerful new 3D geometric feeding primitive combines the reorientation functionality of fences with the rejection functionality of traps.**

**Due to its simplicity, the proposed primitive allows for the development of methods to automate its design process. We present a complete procedure that takes as input any polyhedral part along with its center of mass. Given this input, the procedure identifies all single blade solutions that feed the part. The output is either the set of all valid blade designs or a notification that the part cannot be fed using a single blade.**

## I. Introduction

A *part feeder* takes in a stream of identical parts in arbitrary orientations and outputs them in a uniform orientation. We consider the problem of *sensorless orientation* of parts, in which the initial orientation of the part is assumed to be unknown. In sensorless manipulation, parts are positioned and/or oriented using passive mechanical compliance. The input is a description of the part shape and the output is a sequence of open-loop actions that moves a part from an unknown initial orientation into a unique final orientation. Among the sensorless part feeders considered in literature are the parallel-jaw gripper [1], [2], the single pushing jaw [3], [4], [5], the conveyor belt with a sequence of (stationary) fences [2], [6], [7] or pins [8] placed along its sides, the conveyor belt with a single rotational fence [9], the tilting tray [10], [11], vibratory plates and programmable vector fields [12].

The oldest and still most common approach to automated feeding is the vibratory bowl feeder. It consists of a bowl filled with parts surrounded by a helical metal track [13]. The bowl and track undergo an asymmetric helical vibration that causes parts to move up the track, where they encounter a sequence of mechanical devices such as wiper blades, grooves and traps. Most of these devices are filters that serve to reject (force back to the bottom of the bowl) parts in all orientations except for the desired one. Thus, a stream of oriented parts emerges at the top after successfully running the gauntlet. Such a device is said to *feed* a part, if it allows the part to pass in only one specific orientation.

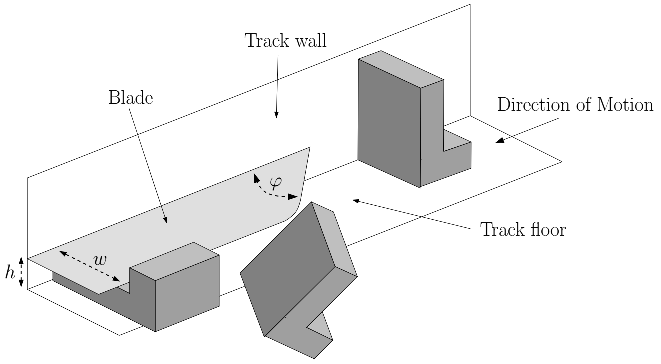

Fig. 1. Vibratory track with blade $B(\varphi, h, w)$. The depicted blade feeds the leftmost part orientation.

Feeder manufacturing still relies on skill, experience and ad-hoc guidelines. Currently, the largest cost of bowl feeders is related to the design of the custom mechanisms used to feed an individual part. The costs and time associated with design of part feeders remains a barrier to flexible automation. In a typical scenario, the feeder bowl takes up to 200 hours to manufacture; 95 % of this time is spent designing the bowl [14]. It is evident that automation of the design process would greatly reduce the production costs and time.

In this light, researchers have used simulation [15], [16], [17], [18], heuristics [2], [19], and genetic algorithms [20] to aid in the design of vibratory bowl layouts. Geometric analysis tools have been developed that help designers visualize the configuration space of a given combination of part and bowl layout [21]. Research to single reorientation and rejection mechanisms has been focused at the design of traps, which are devices constructed by removing sections of the track [22], [23], [24]. The drawback of the latter research is that the proposed devices only apply to two-dimensional parts.

Considering a broader context, the majority of the achievements in the field of sensorless feeding apply to flat, two-dimensional parts, or to parts where the face on which the part rests is known beforehand [25]. Closest in spirit to our work is the work by Berretty *et al.* on reorienting polyhedral parts with a sequence of fences and tilted plates [26].

In this paper, we narrow the gap between industrial bowl feeders and the scientific work on sensorless feeding by introducing a powerful new 3D geometric feeding primitive: the *blade*. Blades combine the reorientation functionality of fences [2], [6], [7] with the rejection functionality of traps [22], [23], [24] into a device that can feed a broad class of 3D

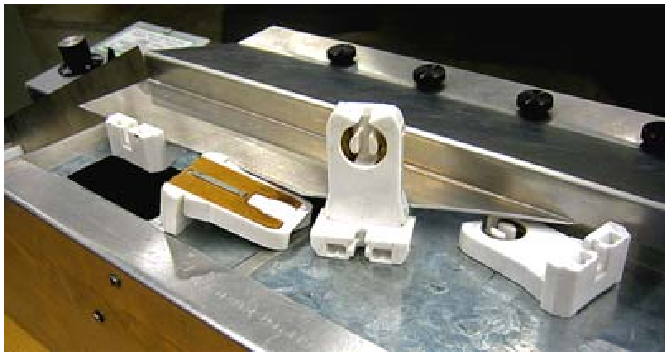

Fig. 2. A side view of the blade feeding the fluorescent fixture mount. Normally, parts fall off the side of the track as illustrated in Fig. 1, but for practical reasons we mimicked this mechanism with the (black) cut-out in the track. The leftmost orientation is fed by the blade, the rightmost orientation will after reorientation end up as the leftmost orientation and thus will also survive the blade, while the middle two orientations will be rejected.

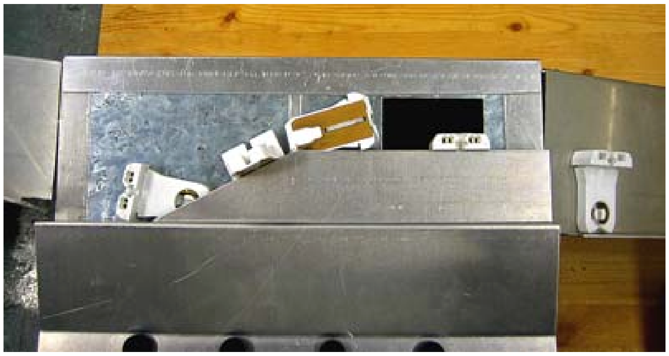

Fig. 3. A top view of the same setup as in Fig. 2.

parts by reorienting and rejecting all but one orientation. The design of blades is inspired by similar devices used in existing bowl feeder systems [13]. Due to its simplicity, the proposed primitive allows for automated design.

A blade is a horizontally mounted convex polygonal metal plate attached to the feeder wall. This plate is parallel to the track and consists of a triangular shaped segment and a rectangular shaped segment (see Fig. 1). A blade is characterized by three parameters: The blade angle, $\varphi$, expressing the slope of the triangular segment. The blade height, $h$, specifying the distance between track floor and blade. The blade width, $w$, which is the width of rectangular shaped segment. Other modeling aspects have no relevant effect on the blade-part interaction, hence a blade can be described as $B(\varphi, h, w)$.

Let us first introduce several assumptions and modeling aspects, before providing an intuitive idea of how blades operate. We consider rigid three-dimensional polyhedral parts of a single type, all of which are assumed to be identical. As in the earlier works on mechanisms for vibratory feeders, we assume these parts move along a flat track in a quasi-static manner without interfering with each other. We also assume the position of the center of mass $C$ to be known, as it dictates the stability of the part. Throughout the paper, we assume zero friction between the part and the blade. While moving up the track, a part rests on the track in a stable pose; that is, the part rests on the track floor and against the track wall, where both contacts support $C$. The stable floor contact is the result of gravity, while a slight tilt of the feeder track assures a stable wall contact. Finally, the part is rejected, i.e. falls of the track, when its center of mass is no longer supported by the track floor. The radius of the helical track is assumed to be large compared to the dimensions of the part, thus allowing us to approximate the section of the track as linear.

Intuitively, the blade operates as follows. A part moves up the track toward the blade in an arbitrary stable pose. As soon as it reaches the blade, the part starts moving along the edge of the triangular segment and, consequently, at some point breaks contact with the track wall. Simultaneously, the part changes its orientation as it moves from its stable pose against the track wall to a stable pose against the blade edge. The specific reorientation of the part depends on both the blade height and blade angle. From the triangular segment, the part moves onto the rectangular segment whereat the rejection of parts takes place. The rejection depends on the blade width, as the blade width dictates whether $C$ is supported by the track floor. Now, to realize the desired feeding property of the blade, we select a width such that all but one part orientation is rejected. The role of the reorientation phase is to manipulate the stable track poses such that such a blade width exists.

Let us formalize our problem definition. Our goal is to develop a procedure for the automated design of blades; the procedure is complete in the sense that it identifies all existing valid blade designs. It receives as input the polyhedral part $P$ along with its center of mass center $C$. The procedure either outputs the set $S$ of all valid blade designs that feeds $P$ or reports that no such blade exist. As a forward note on future research, we remark that $S$ may serve as the starting point for post-processing procedures to identify the blade design that best satisfies a desired set of criteria such as feed-rate optimization or insensitivity to tolerances.

We have conducted a physical experiment where we successfully constructed a blade to feed a concave fluorescent fixture mount (see Fig. 2 and 3). These preliminary experimental results show the practical applicability of the proposed feeding device. In addition, our experimental efforts have provided us several insights into practical issues of our approach. Based on these insights, we make several practical notes throughout the paper, which aim to provide an intuition of how theory translates to practice.

This paper is organized as follows. In section 2, we discuss the part-blade interaction in more detail and provide a number of definitions. In section 3, we first provide an intuitive explanation of our procedure and continue with a discussion at a more detailed level. We conclude in section 4 with a brief summary, several remarks and points of future work.

## II. Modeling the Blade

We will first address the *track poses* which are the possible poses of $P$ whilst it moves up the track. The second subsection provides a similar discussion on *blade poses* which are stable poses of $P$ against the edge of the blade. We connect these two concepts in subsection three by explaining how the blade

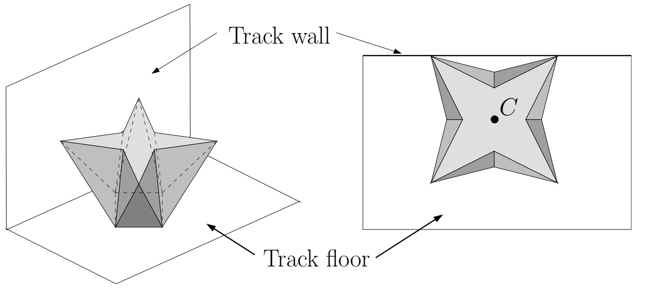

Fig. 4. Side and top view ($P^f_\perp$) of a stable track pose. There exist four possible wall contacts for the depicted floor contact, which correspond to the four edges of the rectangular convex hull of $P^f_\perp$.

reorients $P$ from a track to a blade pose. We conclude with a subsection on the rejection mechanism of the blade.

We can split the blade into two segments. Firstly, the sloped segment of the blade, the *reorientation segment*, which is responsible for reorienting the part as well as transporting the part away from the track wall. Secondly, the blade segment parallel to the wall, the *rejection segment*, which is responsible for rejecting undesired part orientations.

### A. Track Poses

A polyhedral part in three-dimensional space has three rotational degrees of freedom. While moving up the track, $P$ settles in a stable pose on the track, thus discretizing its set of possible orientations. A stable track placement consists of two stable contacts, namely the contact with the (track) floor and with the (track) wall .

A stable *floor contact* is a placement in which $P$ rests on the floor with a face of its convex hull that supports $C$. Observe that such a contact discretizes two of the three degrees of rotational freedom. In the remainder of this paper, we refer to a stable floor contact simply as a floor contact. We denote $P$ in a given floor contact by $P^f$, where $f \in \{1, \ldots, \Sigma_f\}$ specifies the stable convex hull face on which $P$ rests. Note that $O(\Sigma_f) = O(n)$, where $n$ is the number of vertices of $P$.

We assume that blade interaction does not change the floor contact of $P$. Our physical experiments show that for eccentric parts this assumption can be unrealistic, because these parts may topple for certain combinations of floor contacts and relatively large blade heights. This behavior is a subject for future research.

Let us now turn to the second aspect of a stable pose, the stable *wall contact*, which is ensured by the earlier mentioned track tilt. This contact discretizes the remaining degree of freedom, the *roll orientation*, which is the rotation about the axis perpendicular to the floor contact of $P^f$. Let $P^f_\perp$ be the orthogonal projection of $P^f$ on the floor. A stable wall contact for $P^f$ corresponds to a roll orientation in which $P^f$ rests with a stable edge of $P^f_\perp$ against the track wall (see Fig. 4).

Putting the above together, we denote a (stable) track pose as $P^f_e$, where $e \in \{1, \ldots, \Sigma_e\}$ specifies the stable edge of $P^f_\perp$ with which $P^f_e$ rests against the track wall. For each floor contact applies $O(\Sigma_e) = O(n)$, hence in total $O(\Sigma_f \Sigma_e) = O(n^2)$ track poses exist.

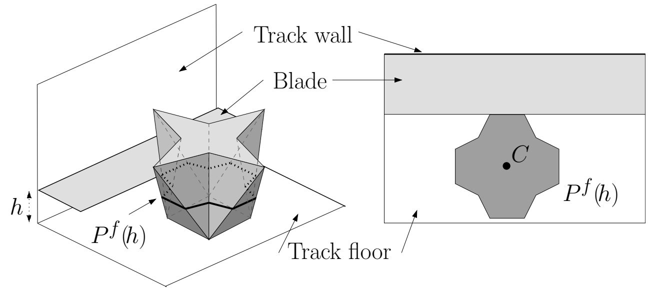

Fig. 5. Side view and cross-section $P^f(h)$ of a stable blade pose. There exist eight possible blade contacts for the depicted floor contact and blade height, which correspond to the eight stable edges of the convex hull of $P^f(h)$.

As a notational remark, we note that throughout the paper the combination of superscript $f$ and subscript $e$, e.g. $P^f_e$, indicates that a constant, variable or function is tied to one specific track pose, i.e., to a unique combination of floor and wall contact. Furthermore, the superscript $f$ notation, e.g. $P^f$, refers to a part for which we only consider the floor contact.

### B. Blade Poses

Let us start with two notes on the modeling of the blade itself. We model $B$ as a planar surface which is parallel to the track and defined by the earlier introduced parameters. Let us refer to the plane in which $B$ lies as the *blade plane*.

Sharing the same concept of floor contact, the blade pose differs from the track pose in that the stable wall contact is replaced by a stable blade contact. Similar to the wall contact, the blade contact discretizes the roll orientation of $P^f$, however, the discretization now depends on the blade height.

More precisely, the blade height $h$ selects a cross-section of $P^f$ with which $P^f$ rests against the blade. This cross-section, denoted by $P^f(h)$, is the polygonal intersection of $P^f$ and the blade plane. The location of the center of mass of $P^f(h)$ is the orthogonal projection of $C$ onto the blade plane (see Fig. 5).

Summarizing, given a blade at height $h$, a stable blade pose is a combination of a stable floor and blade contact, where the latter is a stable placement of $P^f(h)$ against the edge of the blade. We denote a blade pose as $P^f_\theta$, where $\theta$ specifies the roll orientation with respect to a fixed world frame. The combination of superscript $f$ and subscript $\theta$, e.g. $P^f_\theta$, indicates that a constant, variable or function is tied to one specific blade pose.

### C. Part Reorientation

We start off with a general notion. A core concept in our approach is that we model reorientation and the subsequent rejection separately for each track pose $P^f_e$; i.e., for each $P^f_e$, we model the effect of the blade parameters $\varphi$, $h$ and $w$. The resulting set of $P^f_e$ models enables us to quickly determine the effect of any given $B = B(\varphi, h, w)$ on all track poses.

With this concept in mind, we discuss how the reorientation segment of $B$ rotates a given $P^f_e$ to a $P^f_\theta$. We note that this reorientation only depends on the choice of $\varphi$ and $h$. Furthermore, recall that the reorientation process only affects the roll orientation of $P^f_e$, which we will denote by $\theta^f_e$. In the following, we respectively address the role of $h$ and $\varphi$.

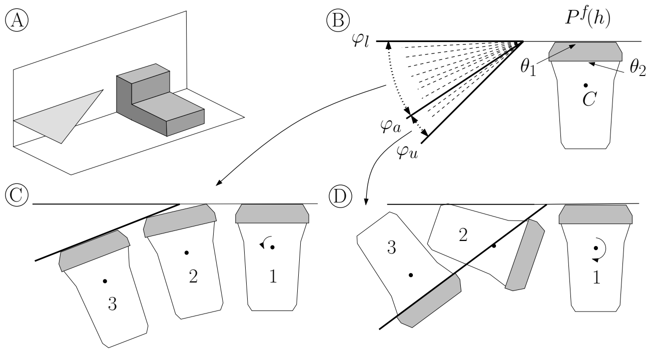

Fig. 6. Side view (fig. A) en top view (fig. B) of the situation just before the reorientation of the depicted fluorescent fixture mount. The gray polygons depict $P^f_e(h)$, which features two stable roll orientations $\theta_1$ and $\theta_2$. Blade angle intervals $\langle \varphi_l, \varphi_a]$ and $[\varphi_a, \varphi_u\rangle$ respectively map $P^f_e$ to $P^f_{\theta_1}$ (fig. C) and to $P^f_{\theta_2}$ (fig. D).

As explained in section II-B, height $h$ selects a polygonal cross-section $P^f_e(h)$ of $P^f_e$ with which the blade interacts. Hence, the reorientation of $P^f_e$ for a fixed $h = H$ effectively is the planar reorientation of $P^f_e(H)$, and visa versa.

This brings us to the role of the blade angle $\varphi$. The choice of $\varphi$ determines how $B$ reorients polygon $P^f_e(H)$ from $\theta^f_e$ to a new roll orientation $\theta'$, and thus, how $P^f_e$ reorients to $P^f_{\theta'}$. We will briefly discuss this mechanism (see [6] for more details).

Let us denote the allowed $\varphi$-interval as $[\varphi_l, \varphi_u]$, where the lower and upper bound $\varphi_l$ and $\varphi_u$ depend on various aspects such as track tilt, part-blade friction, desired feed-rate and the amount of track available for the blade device. Furthermore, the convex hull of $P^f_e(H)$ features one or more stable edges which correspond to a set of $\theta$. Let $\Theta^f_e(H) = \{\theta_1, \ldots, \theta_k\}$ be the ordered sequence of stable $\theta$ of $P^f(H)$ (see Fig. 6).

The general idea of the reorientation mechanism is as follows. For a given $h = H$, the blade angle range $[\varphi_l, \varphi_u]$ can be split in sub-intervals, where each subinterval reorients $P^f_e(H)$ from $\theta^f_e$ to one $\theta_i$ in $\Theta^f_e(H)$ (see Fig. 6). Expressing this mechanism in terms of track and blade poses, we can summarize the reorienting effect of $B$ as follows: For a given $P^f_e$, the set of blades at $h = H$ can be grouped into subsets which correspond to continuous sub-intervals of $[\varphi_l, \varphi_u]$; all blades in one subset map $P^f_e$ to the the same $P^f_\theta$ with $\theta$ in $\Theta^f_e(H)$.

Next, let us review an interesting side effect of the reorientation phase. By choosing the right $\varphi$ and $h$, multiple track poses with the same floor contact can be reoriented into one blade pose. This effect is important to take into account as otherwise the algorithm may discard potential solutions. Moreover, this phenomenon can be used to substantially increase the feed-rate of the blade. E.g. in our experiment, four $P^f_e$ have the same floor contact as the $P^f_\theta$ outputted by $B$; through the right blade design, we can reorient all four $P^f_e$ to this output $P^f_\theta$, hence substantially increasing the feed-rate of $B$.

We close this subsection with a practical note. To guide the part from the reorientation segment to the rejection segment without changing its blade pose, we connect these segments by a curve in the spirit of Brokowski *et al.* [7]. Their approach accommodates both convex and concave parts.

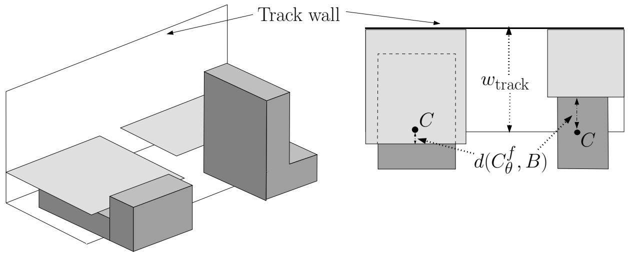

Fig. 7. Side view and top view of two blade poses from our experimental setup, where the left orientation is outputted by the experimental blade. The depicted blades illustrate the critical blade width $w_{\text{track}} - d(C^f_\theta, B)$ for both poses.

### D. Part Rejection

At this point, the blade has reoriented $P^f_e$ into a $P^f_\theta$ which then arrives at the rejection segment. Recall that $P^i_\theta$ is rejected when the floor does not support the center of mass of $P^f_\theta$ anymore. More precisely, rejection occurs when the orthogonal projection of $C$ on the floor plane is no longer contained within the track floor. Although rejection can already occur on the reorientation segment, we only consider rejection of blade poses on the rejection segment, because the former may occur if and only if the latter occurs.

The rejection of $P^f_\theta$ depends on the blade width. We capture this with the notion of *critical blade width*, which is the maximum blade width that $P^f_\theta$ survives. The critical blade width is defined as $w_{\text{track}} - d(C^f_\theta, B)$, where $w_{\text{track}}$ denotes the track width and $d(C^f_\theta, B)$ denotes the distance between the projection of $C$ and the rejection segment in $P^f_{\theta\perp}$ (see Fig. 7). The value of $w_{\text{track}}$ is chosen such that any blade pose resting against the rejection edge is free of the track wall. Furthermore, note that $C$ can be underneath the blade, in which case $d(C^f_\theta, B)$ is negative. We see in Fig. 7 that if the blade width is larger than $w_{\text{track}} - d(C^f_\theta, B)$, then $C$ is no longer supported; hence, the corresponding blade rejects $P^f_\theta$.

As a brief digression to a scenario *with* friction, we note that certain combinations of blade designs and part poses may yield a risk that the blade prevents the part from falling back into the bowl. For an intuitive understanding of such events, imagine the leftmost case in Fig. 7 with a slightly larger blade width such that the depicted track pose would become unstable. In our frictionless setting, however, such events do not occur.

## III. Procedure for Blade Design

In the first subsection, we provide an intuitive sketch of our procedure. The second subsection then zooms in on the building blocks of the procedure, which are the functions that model the effect of the blade parameters on a given track pose. The third and last subsection discusses the use of these functions to identify valid solution set $S$.

### A. Overview Procedure

As explained before, a blade is defined by three parameters: height $h$, blade angle $\varphi$ and width $w$. These parameters span a three-dimensional space, the *blade space*, where each point describes a unique blade. We will forge the concepts from the previous section into a function in blade space, a function that describes the impact of the blade parameters on a given track pose. Let us consider an arbitrary $P_e^f$.

Depending on the choice of $\varphi$ and $h$, the reorientation segment transfers $P_e^f$ into some blade pose $P_\theta^f$. As explained in section II-D, once we have $P_\theta^f$, we can easily calculate the corresponding critical blade width. Concatenating these two steps, we can conclude that the choice of $\varphi$ and $h$ directly determines the critical blade width that $P_e^f$ can survive. We capture this relation in the *critical function* $\omega_e^f(\varphi, h)$ that maps $\varphi$ and $h$ to the corresponding critical blade width.

Observe that a critical function describes a surface in the blade space. This surface features a convenient property, namely a point above the surface corresponds to a blade rejecting $P_e^f$, while any point below the surface corresponds to a blade that $P_e^f$ survives. The next subsection provides a more detailed explanation on the construction and properties of these surfaces, which we refer to as *critical surfaces*.

Let us summarize, as this is an important aspect of our procedure. Each critical surface corresponds to *one* specific track pose and specifies the impact of all possible blade parameterizations on this specific track pose. There exist $O(n^2)$ track poses, thus the blade space contains $O(n^2)$ critical surfaces. This construction allows us to determine for any blade which track poses will survive.

Recall that our goal is to reject all but one track pose. In terms of our geometric model, this goal translates to finding a point in the blade space such that all but one critical surfaces are below this point. Let us refer to such a point as a *valid solution $s$* and to the set of such points as the *valid solution set $S$*. The third subsection discusses the valid solution set in more detail.

### B. Critical Surfaces

In the previous section, we have introduced the concept of critical functions. We now turn to the generation of $\omega_e^f(\varphi, h)$ which is the product of three subsequent modeling phases.

In the first phase, we abstract from blade contacts by constructing a three-dimensional *push function* $\rho^f$ for $P^f$. This function models the effect of reorienting $P^f$ by a push action with a line and is defined in the *push space*. This three-dimensional space is spanned by $h$, the height of the push line, and the $\theta$ of $P^f$ before and after the push action.

In the second modeling phase, we tailor the push function to model the reorientation of $P_e^f$ by the blade, resulting in the *reorientation function* $\delta_e^f$. More precisely, this function maps $P_e^f$ for a given $\varphi$ and $h$ to a track pose $P_\theta^f$. The space in which $\delta_e^f$ is defined, is spanned by the blade height $h$, blade angle $\varphi$ and the roll orientation $\theta$ of $P_\theta^f$.

Finally, in the third modeling phase, we model the rejection of the reoriented $P_e^f$ by transforming its reorientation function into a critical function $\omega_e^f$. As already explained, $\omega_e^f$ is defined in the blade space which is spanned by $h$, $\varphi$ and blade width $w$.

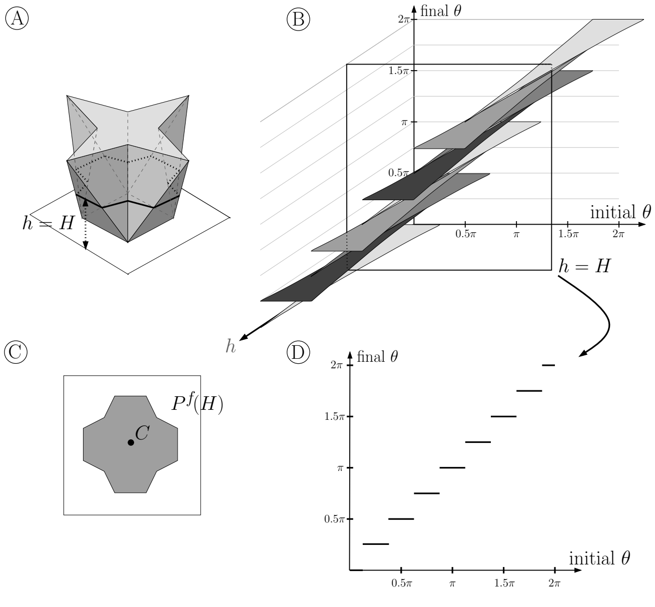

Fig. 8. Figure B illustrates $\rho^f$ for $P^f$ in figure A. Furthermore, figure D illustrates the planar push function for cross-section $P^f(H)$ of $P^f$ in figure C. The cross-section of $\rho^f$ at $h = H$ shows the relation between $\rho^f$ and the planar push function.

*Push Function*

Let us start off by considering the blade at a fixed height $h = H$ and study the effect of the reorientation segment on the cross-section $P^f(H)$ of $P^f$. For this planar case, the effect of the reorientation segment on $P^f(H)$ resembles that of a fence mounted along a conveyor belt. As proposed by Peskin and Sanderson [5] for such fences, this effect can be modeled as pushing $P^f(H)$ with a line until the polygon assumes a stable orientation against this push line. The effect of push actions can be modeled by a push function [1], [2], [27] which are step functions that map an initial to a final orientation, both expressed relative to the push line. Hence, for a given $h$, the reorientation by the blade can be captured as a step function that maps an initial $\theta$ to a final stable $\theta$ of $P^f(H)$.

Let us consider the effect of $h$ on the push function. Intuitively, the result is a "stack" of continuously varying push functions, where each cross-section at $h = H$ is a standard push function. As $h$ changes, line segments of step function pop up, grow, shrink, and disappear. Between their appearance and disappearance, these segments sweeps out a continuous surface in the push space (see Fig. 8). We refer to these surfaces as *push patches*. They will eventually form the elementary units in the critical surfaces. Recapitulating on the push function, given an initial $\theta$ of $P^f$ and push height $h$, $\rho^f(\theta, h)$ outputs the final stable $\theta$ of $P^f$.

As a remark on the complexity of $\rho^f$, we note that each $\rho^f$ consists of $O(n)$ push patches of $O(1)$ complexity.

*Reorientation Function*

Next, we use $\rho^f(\theta, h)$ as a basis to construct the reorientation functions $\delta_e^f(\varphi, h)$ for all track poses $P_e^f$ with floor contact

$P^f$. Given a blade angle and height, $\delta_e^f(\varphi, h)$ specifies for $P_e^f$ into which blade pose $P_\theta^f$ it will be reoriented.

The step from $\rho^f$ to $\delta_e^f$ effectively is nothing more then a change in perspective. Just as the push function, $\delta_e^f$ maps a combination of any given initial orientation and height to a final stable orientation. However, where $\rho^f$ takes as input the orientation of $P^f$ relative to the fixed push line, $\delta_e^f$ changes perspective and takes as input the orientation of the push line relative to the fixed $P_e^f$. In the latter case, the push line is the reorientation segment and its orientation is blade angle $\varphi$. As with $\rho^f$, the output of $\delta_e^f$ is the final stable $\theta$ of $P^f$.

Henceforth, $\varphi$ of $\delta_e^f$ uniquely maps to a push angle $\theta$ of $\delta^f$. Similarly, the interval of possible blade angles $[\varphi_l, \varphi_u]$ of $\delta_e^f$ uniquely maps to an equally sized $\theta$-interval of $\delta^f$.

In summary, the height intervals of $\rho^f$ and $\delta_e^f$ are the same, the $\varphi$-domain of $\delta_e^f$ directly maps to some $\theta$-interval of $\rho^f$, and both functions specify the stable $\theta$ of $P^f$ as output. In other words, $\delta_e^f$ is a "slice" of $\rho^f$ as defined within the above described $\theta$-interval. As a final remark, we note that the complexity of $\delta_e^f$ is $O(n)$, which is the same complexity as that of its corresponding $\rho^f$.

*Critical Function*

Recall that given $\varphi$ and $h$, $\delta_e^f$ maps $P_e^f$ to a stable blade pose $P_\theta^f$. As explained in section II-D, the critical blade width for $P_\theta^f$ can be determined using basic geometry. Henceforth, the transformation from $\delta_e^f$ to $\omega_e^f$ is a one-to-one mapping from blade pose to critical blade width. With $\omega_e^f$, we now have the description of the critical surface of $P_e^f$. Recall that the push surface was comprised of patches. Due to the one-to-one mapping, these patches are conserved and thus also comprise the critical surfaces. We refer to the latter type of patches as *critical patches* (see Fig. 9). Finally, the one-to-one mapping preserves the complexity of $\delta_e^f$, hence $\omega_e^f$ consists of $O(n)$ critical patches of $O(1)$ complexity.

### C. Solution

The blade space contains $O(n^2)$ critical surfaces of complexity $O(n)$, each corresponding to a $P_e^f$. Hence, the total complexity of the critical arrangement is $O(n^3)$. Recall that a point which lies above all but one surface in terms of blade width represents a valid solution; that is, it represents a blade that rejects all but one track pose. The valid solution set $S$ is the blade space between the upper envelope and the critical surfaces one level down, where the upper envelope is the surface comprised of all (partial) critical patches that are maximal with respect to $w$ [28]. In geometry, this space is referred to as the *n-level* of an arrangement of surfaces [28].

In our surface arrangement, a level typically consists of many subspaces due to the $O(n^2)$ surfaces interacting with each other in or near the upper envelope. Consequently, the valid solution set $S$ typically consists of many subsets, each featuring valid solutions with specific characteristics.

Once $S$ is generated, the algorithm can reports this set of valid blade designs. No valid solution exists when the n-level is empty; in other words, when the entire upper envelope consists of coinciding critical surfaces. The occurrence of the latter is of course extremely unlikely. More general, we observe that a $(\varphi, h)$-domain for which two or more critical surfaces coincide, cannot serve as basis for a valid solution set, as the corresponding set of blades would allow more than one blade pose to survive. Blades may not exist for parts with geometric symmetries; characterizing the class of parts for which blades are guaranteed to exist is a subject for future research.

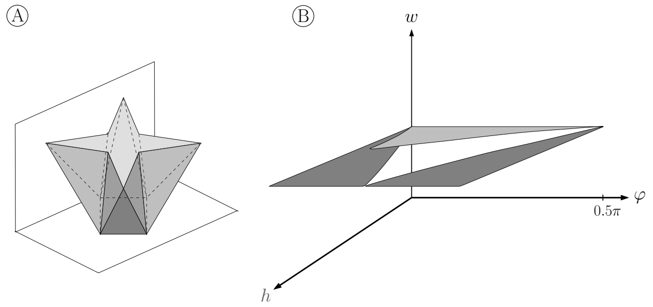

Fig. 9. Figure B shows $w_e^f$ for the track pose in figure A and $[\varphi_l, \varphi_r] = \langle 0, \pi/2 \rangle$. The increase in slope of $w_e^f$ is explained by the increase of the diameter of $P^f$. Observe that the surface of $w_e^f$ is a slice of $\rho^f$ (Fig. 8) where the final $\theta$ orientation is transformed to the critical blade width.

Having said this, let us revisit the phenomenon of multiple track poses collapsing to one blade pose. Let $P_{e1}^f$ and $P_{e2}^f$ be two such track poses. In terms of their critical surfaces, this phenomenon corresponds to the partial coincidence of one or more critical patches of the critical surface of $P_{e1}^f$ and $P_{e2}^f$ . Coinciding critical surfaces in general eliminate valid solutions, hence it is essential to take this phenomenon into account when extracting the n-level of the critical surface arrangement. In short, we deal with this by attaching to each critical patch an identifier that expresses the corresponding feature of $P^f$. When two (partially) coinciding critical patches with the same identifier are detected, then the corresponding blades will not be discarded as valid solutions.

Geometric algorithms exist to calculate the n-level of our critical surface arrangement [28]; these algorithms typically apply a divide-and-conquer approach in combination with vertical cell decomposition. Their polynomial worst-case running times can in theory be of relatively high degree, but we anticipate that the running time in practice remains far below below the worst-case bounds.

## IV. Conclusion

We have presented a new geometric primitive for vibratory bowl feeders, which receives a stream of identical polyhedral parts in arbitrary stable pose as input and outputs parts in one single pose. A blade first reorients the parts from their arbitrary track poses to an intermediate blade pose, after which all but one blade poses are rejected back into the bowl. Under the assumption that the motion of the part is quasi-static and parts do not topple during blade interaction, our complete procedure reports either all possible single blade solutions or that no solution exists. We have experimentally shown the practical applicability by constructing a blade that feeds a fluorescent fixture mount.

We anticipate that the proposed procedure generalizes from polyhedra to semi-algebraic 3D parts. The ground for this anticipation is work by Rao and Goldberg [29] on 2D grasp functions for non-polygonal parts. These functions are very similar to 2D push functions which form the basis of our critical function. Furthermore, we anticipate that the assumption of zero friction between the part and the blade can be partially relaxed by using a result by Berretty *et al.* [30]. Berretty *et al.* model Coulomb friction for planar part-fence interaction which is very similar to our part-blade interaction.

Our aim in this paper has been to take a first step in the design of complete algorithms for three-dimensional parts in the context of vibratory bowls. Near future research is the adaptation of existing geometric algorithms such that the valid solution space can be generated as efficiently as possible. A second subject to be addressed is the assumption that interaction between part and blade does not change the floor contact of the part; this assumption may be unrealistic for certain combinations of thin parts and large blade heights.

Further down the road, we identify three additional challenging problems. A first interesting problem is the application of a sequence of blades instead of one blade. We conjecture that this extension allows a further optimization of for example feed rate and insensitivity to tolerances. Secondly, in light of reconfigurable part feeders, it is of interest to augment the proposed procedure to generate blade designs within limitations imposed by an adjustable hardware implementation of the blade. And thirdly, given the complete solution set $S$ output by our procedure, a promising research direction lies in the design of algorithms to post-process $S$ in order to identify blade designs that best satisfy criteria such as feed-rate optimization or insensitivity to tolerances.

In conclusion, blades introduce a powerful new geometric primitive for feeding 3D parts. By combining properties of fences and traps, blades can actively reorient or reject orientations to feed a broad class of industrial parts. Further experiments and research are needed so that blades can be incorporated into a truly algorithmic part feeding system.

## Acknowledgment

Onno C. Goemans acknowledges the support of the Netherlands Organisation for Scientific Research (NWO). Ken Goldberg thanks Brian Carlisle for useful discussions and acknowledges support from CITRIS, UC Berkeley's Center for Information Technology Research in the Interest of Society. A. Frank van der Stappen is partially supported by the Dutch BSIK/BRICKS-project.

## References

[1] K. Goldberg, "Orienting polygonal parts without sensors," *Algorithmica*, vol. 10, pp. 201–225, 1993.

[2] J. Wiegley, K. Goldberg, M. Peshkin, and M. Brokowski, "A complete algorithm for designing passive fences to orient parts," *Assembly Automation*, vol. 17, no. 2, pp. 129–136, 1997.

[3] S. Akella and M. Mason, "Posing polygonal objects in the plane by pushing," *IEEE ICRA*, pp. 2255–2262, 1992.

[4] K. Lynch and M. Mason, "Stable pushing: Mechanics, controllability, and planning," *International Journal of Robotics Research*, vol. 15, no. 6, pp. 533–556, 1996.

[5] M. Peshkin and A. Sanderson, "The motion of a pushed sliding workpiece," *IEEE J. of Robotics and Automation*, vol. 4, no. 6, pp. 569–598, 1988.

[6] R.-P. Berretty, K. Goldberg, M. Overmars, and A. van der Stappen, "Computing fence designs for orienting parts," *Computational Geometry: Theory and Applications*, vol. 10, no. 4, pp. 249–262, 1998.

[7] M. Brokowski, M. Peshkin, and K. Goldberg, "Optimal curved fences for part alignment on a belt," *ASME Trans. of Mechanical Design*, vol. 117, pp. 27–34, 1995.

[8] T. Zhang, G. Smith, R.-P. Berretty, K. Goldberg, and M. Overmars, "The toppling graph: Designing pin sequences for part feeding," *IEEE ICRA*, pp. 139–146, 2000.

[9] S. Akella, W. Huang, K. Lynch, and M. Mason, "Parts feeding on a conveyor with a one joint robot," *Algorithmica*, vol. 26, no. 3, pp. 313–344, 2000.

[10] M. Erdmann and M. Mason, "An exploration of sensorless manipulation," *IEEE J. of Robotics and Automation*, pp. 367–279, 1988.

[11] B. Natarajan, "Some paradigms for the automated design of parts feeders," *Int. Journal of Robotics Research*, vol. 8, no. 6, pp. 89–109, 1989.

[12] K.-F. Bohringer, V. Bhatt, B. Donald, and K. Goldberg, "Algorithms for sensorless manipulation using a vibrating surface," *Algorithmica*, vol. 26, pp. 389–429, 2000.

[13] G. Boothroyd, *Automatic Assembly and Product Design*. Taylor & Francis Ltd, 2005.

[14] *Private conversation with Warren Wilson of TechnaVibes*, Placerville, CA, 2002.

[15] D. Berkowitz and J. Canny, "Designing part feeders using dynamic simulation," *IEEE ICRA*, pp. 1127–1132, 1996.

[16] M. Jakiela and J. Krishnasamy, "Computer simulation of vibratory part feeding and assembly," $2^{nd}$ *Int. Conf. on Discrete Element Methods*, pp. 403–411, 1993.

[17] G. Maul and M. Thomas, "A systems model and simulation of the vibratory bowl feeder," $2^{nd}$ *International Conference on Discrete Element Methods*, vol. 16, no. 5, pp. 309–314, 1997.

[18] J. Selig and J. Dai, "Dynamics of vibratory bowl feeders," *IEEE ICRA*, pp. 3299–3304, 2005.

[19] L. Lim, B. Ngoi, S. Lee, S. Lye, and P. Tan, "A computer-aided framework for the selection and sequencing of orientating devices for the vibratory bowl feeder," *Int. J. of Production Research*, vol. 32, no. 11, pp. 2513–2524, 1994.

[20] A. Christiansen, A. Edwards, and C. Coello, "Automated design of parts feeders using a genetic algorithm," *IEEE ICRA*, pp. 846–851, 1996.

[21] M. Caine, "The design of shape interactions using motion constraints," *IEEE ICRA*, pp. 366–371, 1994.

[22] P. Agarwal, A. Collins, and J. Harer, "Minimum trap design," *IEEE ICRA*, pp. 2243–2248, 2001.

[23] R.-P. Berretty, K. Goldberg, M. Overmars, and A. van der Stappen, "Trap design for vibratory bowl feeders," *International Journal of Robotics Research*, vol. 20, pp. 891–908, 2001.

[24] O. Goemans, A. Levandowski, K. Goldberg, and A. van der Stappen, "On the design of guillotine traps for vibratory bowl feeders," *IEEE CASE*, pp. 79–86, 2005.

[25] M. Mason, *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[26] R.-P. Berretty, M. Overmars, and A. van der Stappen, "Orienting polyhedral parts by pushing," *Comput. Geom.*, vol. 21, no. 1-2, pp. 21–38, 2002.

[27] R. Brost, "Automatic grasp planning in the presence of uncertainty," *Journal of Robotic Research*, vol. 7, no. 1, pp. 3–17, 1988.

[28] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry – Algorithms and Applications*, 1997.

[29] A. Rao and K. Goldberg, "Friction and part curvature in parallel-jaw grasping," *Journal of Robotic Systems*, vol. 12, no. 6, pp. 365–382, 1995.

[30] R.-P. Berretty, K. Goldberg, M. Overmars, and A. van der Stappen, "Algorithms for fence design," *Robotics, the algorithmic perspective*, pp. 279–295, 1998.

[31] Y.-B. Chen and D. Ierardi, "The complexity of oblivious plans for orienting and distinguishing polygonal parts," *Algoritmica*, vol. 14, pp. 367–397, 1995.

[32] M. Sharir and P. Agarwal, *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.