Configuration Products in Geometric Modeling

Saigopal Nelaturi
Spatial Automation Laboratory
University of Wisconsin-Madison
saigopan@cae.wisc.edu

Vadim Shapiro
Spatial Automation Laboratory
University of Wisconsin-Madison
vshapiro@engr.wisc.edu

ABSTRACT

The six-dimensional space $SE(3)$ is traditionally associated with the space of configurations of a rigid solid (a subset of Euclidean three-dimensional space E^3). But a solid can be also considered to be a set of configurations, and therefore a subset of $SE(3)$. This observation removes the artificial distinction between shapes and their configurations, and allows formulation and solution of a large class of problems in mechanical design and manufacturing. In particular, the *configuration product* of two subsets of configuration space is the set of all configurations obtained when one of the sets is transformed by all configurations of the other. The usual definitions of various sweeps, Minkowski sum, and other motion related operations are then realized as projections of the configuration product into E^3 . Similarly, the dual operation of *configuration quotient* subsumes the more common operations of unsweep and Minkowski difference. We identify the formal properties of these operations that are instrumental in formulating and solving both direct and inverse problems in computer aided design and manufacturing. Finally, we show that all required computations may be implemented using a fast parallel sampling method on a GPU.

1. INTRODUCTION

1.1 Shape in configuration space

Swept sets (or sweeps) are one of the fundamental representation schemes in geometric and solid modeling [29]. Generally a swept set S may be represented by a pair (A, B) of sets and a mapping $g : A \times B \to \mathbb{R}^3$, such that $S = g(A, B)$. Typically A is a one parameter family of rigid transformations, B is a point set in $\mathbb{R}^3$, and

$$g(A, B) = \texttt{sweep}(B, A) = \bigcup_{a \in A} B^a$$

where B^a denotes set B transformed by a [33]. Most commercial CAD systems provide (limited) functionality for constructing swept solids, for example, in a form of a two-dimensional cross section moving on a space trajectory that is transversal to the plane of the cross section. The problems of constructing, approximating, and representing these and other types of sweeps have been studied extensively, e.g. see a recent survey in [24]. Most known methods assume particular representation of the point set B and/or of the trajectory A, and tend not to be broadly applicable.

Minkowski sum [32, 19] of two subsets A, B of $\mathbb{R}^3$ is defined as the direct sum $A \oplus B$, with A and B being treated as a collection of (vector) positions. In this sense, Minkowski sum can be also considered a form of a sweep $g(A, B) = A \oplus B$, which is both generalized because A is no longer limited to a one-parameter family, and restricted because A contains only translations but does not allow any rotations. Notice also that the symmetry of Minkowski sum with respect to sets A and B is no longer obvious when it is viewed as a swept solid (but of course, it is still true). Despite this restriction, Minkowski operations are used extensively in motion planning [23, 21], containment and packaging [26] layout [3], image processing [32] and many other graphics and shape modeling applications [19, 15]. Their popularity is largely due to the rich algebraic structure that forms the foundation of mathematical morphology [32]. The same algebraic structure has been shown to exist for the traditional sweeps [18], reinforcing the close relationship between sweeps and Minkowski operations. For an overview of Minkowski sum applications see [26], and for a recent review of Minkowski sum computations see [22].

In order to unify sweeps and Minkowski operations within a single, more general and hence more powerful computational framework, we will consider a solid in terms of the positions *and* orientations associated with its points. This view is equivalent to specifying a coordinate system at each point in the solid, thus implying the solid can be treated as a set of rigid transformations relative to an absolute coordinate system, and therefore as a subset of the six dimensional configuration space $\mathbb{R}^3 \times SO(3)$ [21]. As a consequence, the artificial distinction between solids and rigid body transformations disappears, because both are now subsets of the configuration space. Furthermore, this view leads to a natural generalization of a swept set.

Given a pair (A, B) of sets in the configuration space, a *configuration product* is a mapping $f : A \times B \to \mathbb{R}^3 \times SO(3)$ defined by

$$f(A, B) = A \otimes B = \bigcup_{a \in A, b \in B} a \cdot b$$

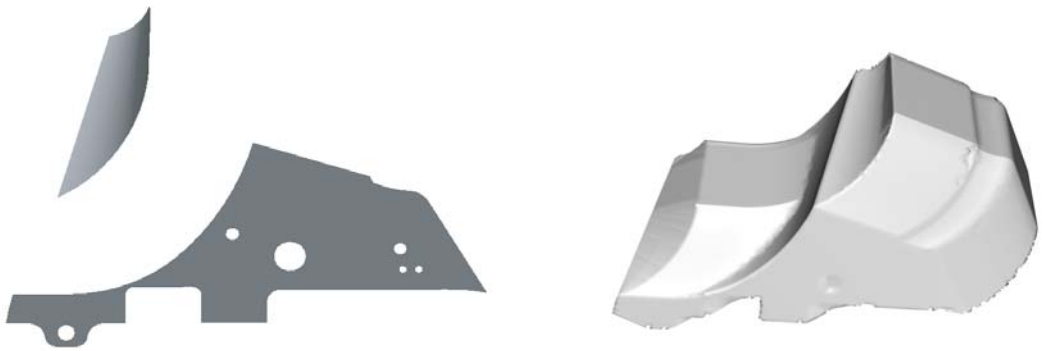

Figure 1: Selected surfaces (left) for which the Minkowski sum (right) is computed. Features from both input surfaces can be seen in the Minkowski sum

where $\cdot$ represents matrix multiplication. Swept sets and Minkowski sums are then both sets of configurations b transformed by rigid transformations a, and projected as point sets into the Euclidean space $\mathbb{R}^3$.

This paper argues that configuration product is a key geometric modeling operation that allows the formulation and solution of many problems in spatial design/planning involving relative configuration and/or motion constraints. Broadly, all such problems can be classified as either direct or inverse.

Direct Problems. These problems require computing 6d space (the positions and orientations) occupied by a moving object. Such problems reduce to a direct evaluation and representation of the configuration product $A \otimes B$. In addition to the classical problems of computing a sweep and Minkowski sum, computing the space occupied by an object while contacting a surface in various orientations, for example the orientations defined by the normal and tangent at each point, is an example of a direct problem. Another example of direct problem is the determination of a mechanism's (e.g. robot's) workspace, where it is required to explicitly compute all the positions and orientations achievable by a mechanism. It is common to distinguish between reachable (position) and dextrous (orientation) workspaces [1], but both are special cases of the configuration product.

Inverse Problems. Such problems impose constraints on a shape or on an allowed set of configurations implicitly by the expression $A \otimes B \subseteq C$, where C is a given subset of $\mathbb{R}^3 \times SO(3)$, and require computing the largest possible set of configurations A or the largest shape B that satisfy the constraint. Inverse problems include packaging, where set B must fit inside C under a set of transformations A, and motion planning problems where C plays the role of free (configuration) space. Formally, the inverse problems can be solved using an operation dual to configuration product called *configuration quotient*, which is a proper generalization of the Minkowski difference and unsweep operations.

Several of the direct and inverse problems described above, such as sweeps over manifolds (curves and surfaces), design of maximal shapes under arbitrary motion constraints, and determination of maximal transformations of shapes to satisfy containment constraints, are difficult to formulate and solve except in special cases. All these problems and many others involving general motions and relative configurations of solids may be effectively solved using configuration products and quotients, as we will show in this paper.

1.2 Paper Outline

Basic properties of configuration products and quotients, are summarized in in Section 2. The duality between configuration products and quotients subsumes the well known duality relationship for Minkowski sum/ difference, and for `sweep`/ `unsweep` operations [18]. The important observation is that a quotient can be used to define both a maximal shape that remains within a set under a specified set of transformations, and also the maximal set of transformations of a shape to satisfy containment constraints.

In Section 3 we demonstrate applications of configuration products and quotients in manufacturing process planning applications. The specific problems formulated and solved include:

1. Computing the 6d space occupied by a weld gun robot moving over the surface of a work part as the gun maintains specified contact constraints.
2. Computing the set of transformations of a weld gun that avoid collisions with surrounding tooling while maintaining contact with the work part.
3. Defining constraints on the shape of a weld gun from constraints on its allowed transformations

In Section 4, we describe a simple fast parallel implementation to sample the configuration product by sampling its primitive sets in $\mathbb{R}^3 \times SO(3)$ and computing pairwise multiplications of the sampled transformations. The inherent parallelism is mapped onto the GPU architecture where the configuration products can be computed at a fraction of the computational cost associated with a similar sampling algorithm on the CPU. Projections into $\mathbb{R}^3$ and $SO(3)$ allow rapid approximate computations of Minkowski sums, sweeps, and other special cases of configuration products and quotients. This technique was used to compute the Minkowski sum in Figure 1.

1.3 Related Work

Configuration spaces have been used to model a number of problems involving kinematic relationships between shapes and their motions, most notably for planning a collision free path of a robot in the presence of obstacles [23, 21]. Typically, such problems require computing a map called the C-space map that partitions configuration space into three disjoint components - obstacle, contact, and free space [38, 30, 13, 8, 7]. The C-space map is usually computed using one of two standard approaches. One approach is to represent free space as a semi-algebraic set bounded by contact surface patches, where each patch represents a constraint on the robot's motion when it contacts an obstacle [13, 8, 38]. The other approach is to represent free space as a sequence of slices in configuration space where each slice represents the contact constraints when the object translates with fixed orientation [7, 30]. In either case, it is evident that the map is determined by the moving shape and its allowed transformations. The C-space map is also clearly applicable in workspace computation [25]. The workspace can be thought of as a stack of slices where each slice represents all the positions achievable in a particular orientation.

Configuration spaces have been studied in the context of designing shapes from their motion constraints. Caine [7] uses the configuration space obstacles to design and modify planar vibratory feeder boundaries based on motion constraints of the parts that are fed. For planar motion, the configuration space obstacle is a solid in the 3d configuration space constructible by stacking obstacles for each part orientation. The obstacle for a single part orientation is the Minkowski sum of the feeder and the reflection of the part. A similar technique is used by Brost [4] who uses configuration spaces to support the design of polygonal fixture/part pairs for form closure, i.e. part immobility under infinitesimal transformations. Form closure configurations are determined by intersection points of finitely many contact manifolds derived from shapes of the fixtures and the part. The contact manifolds are the boundaries of configuration space obstacles. Sacks [30] demonstrates an algorithm to compute and visualize the C-space map for such problems where the configuration space is three dimensional, determined by translations and rotations in the plane.

Caine [6] identifies design of shape from motion constraints to be the inverse problem of mapping a contact manifold to a pair of shapes whose interaction represents the contact manifold. He also observes such a mapping is clearly under-constrained (many pairs of shapes can yield same contact manifold), and so the design problem may be properly constrained by fixing one of the two shapes to generate the other. Then it is possible to create a maximal shape that is guaranteed not to violate contact constraints. For example, it is well known that the maximal shape A that maintains contact with a shape B while constrained to translate within a larger static shape C is the erosion [32, 26] $C \ominus \hat{B}$ ($\hat{B}$ represents the reflection of B), which is defined in terms of the Minkowski difference. Ilies and Shapiro [18] show that the maximal shape that is constrained to move in a one parameter family of transformations within an envelope is defined by the `unsweep` operation, which is the dual of the standard `sweep`. Another class of inverse problems, accessibility problems in manufacturing process planning are often formulated using Minkowski operations to determine a set of feasible configurations. This approach was used in [35, 34] for planning work part accessibility by a coordinate measurement machine.

Most of the above problems may be reformulated in terms of configuration products and quotients, leading to more general solutions for both sets and transformations in configuration space. We demonstrate an approximate computation of configuration products in Section 4 by sampling the primitive sets in configuration space and computing pairwise products of transformations on the Graphics Processing Unit. The sampling approach has been used by others to compute both sweeps and Minkowski sums in $\mathbb{R}^3$; for example, an approach for sampling Minkowski sum is recently proposed by [22]. A related work by Kavraki [20] describes computation of configuration space obstacles by sampling primitive sets and computing their convolution using a Fast Fourier Transform.

2. CONFIGURATION SPACE ALGEBRA

2.1 Solids in Configuration space

General affine transformations in $n-$ dimensional Euclidean space can be represented as linear transformations in projective space via homogenous coordinates and $(n+1) \times (n+1)$ matrices. The subset of such transformations that preserve distances and orientations are called rigid transformations, and they form a Lie group called the Special Euclidean group $SE(n) = \mathbb{R}^n \times SO(n)$ [31]. Solids are represented as subsets of $\mathbb{R}^3$, and hence our focus is restricted to the group $SE(3)$ of rigid transformations in $\mathbb{R}^3$. Since the position and orientation of a solid may be abstracted by a coordinate frame represented as a rigid transformation relative to an absolute coordinate system, $SE(3)$ is clearly a representation of configuration space [39]. The choice of the reference coordinate frame on any given solid is not unique but is often selected based on convenience or some physically meaningful considerations, as is the case, for example, with the Denavit-Hartenberg convention for mechanisms [11].

Associating a position and orientation to every point in a solid [1] facilitates a representation of the solid itself as a point set in $SE(3)$, such that the point set in Euclidean space is a projection of the point set in $SE(3)$ onto its translational components via the map $\pi : \mathbb{R}^3 \times SO(3) \rightarrow \mathbb{R}^3$. This association may be conceptualized as attaching a coordinate system at every point in the solid and is formally captured

[1] As usually, we assume that the solid is an r-set [29].

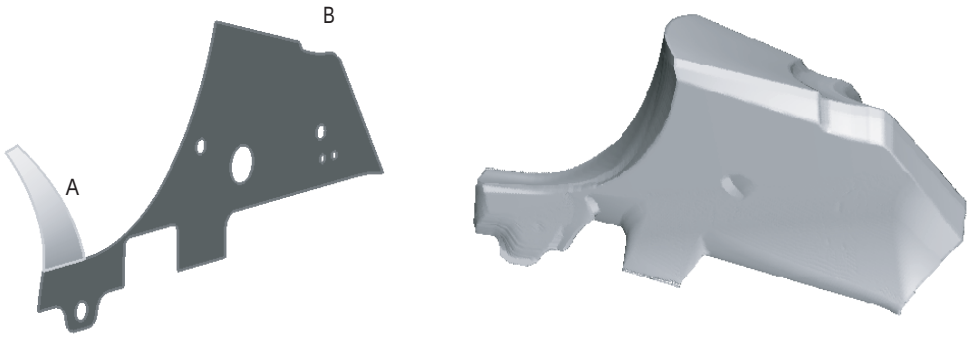

Figure 2: Sweeping one surface by another is computed as a configuration product of the surfaces. Left: The surfaces A, B for which the product $\gamma(A) \otimes \gamma(B)$ is computed. The embedding $\gamma(A)$ is defined by transformations that locate a coordinate system along the normal and principal directions of curvature for each point in A. The set $\gamma(B)$ is defined by associating the identity orientation with each position in B. Right: The projection $\pi(\gamma(A) \otimes \gamma(B))$ corresponds to the sweep of surface B as it moves along the transformations defined by $\gamma(A)$

by an *embedding* $\gamma : \mathbb{R}^3 \rightarrow SE(3)$ defined by

$$\gamma(T) = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$$

where $T = (x, y, z)$ is the position of a point in the solid and R is a rotation matrix. It follows that $\pi \circ \gamma = I$, where I represents the identity transformation.

Just as there is no unique way to choose a single coordinate frame to represent a solid's configurations, there is no unique embedding of a solid in $SE(3)$. However, it is possible to choose a physically meaningful embedding by observing that a solid's movement by a continuous set of transformations is always restricted to some connected subgroup of $SE(3)$ defined by the solid's degrees of freedom. For example, if a solid is constrained to rotate in a plane, it is not physically meaningful to embed the solid with arbitrary orientation to its points – the representation of the solid in the configuration space will be inconsistent with its physically achievable configurations. On the other hand, since the product of two transformations in a subgroup will always remain within the subgroup, embedding the solid as a set of transformations within the subgroup defined by its degrees of freedom is a physically meaningful choice. In the special case where a solid is restricted to translate, the projection of the configuration product of the solid and its allowed translations should be the Minkowski sum of the solid and its trajectory. So it is required to define the embeddings of the solid and its trajectory to be consistent with the Minkowski sum and its properties. In particular, since the Minkowski sum commutes, the rotation component of both embeddings must be the identity rotation. If the embeddings include non trivial rotations representing orientations, $\pi(A \otimes B) \neq \pi(B \otimes A)$ unless $\gamma(A), \gamma(B) \subset \mathbb{R}^3 \times I$. For example the two surfaces in Figure 1 were embedded in $SE(3)$ using identity orientations, and their Minkowski sum was computed via projection of their configuration product in $\mathbb{R}^3$.

Assigning the identity orientation to point positions in the solid ensures that the embedding is always transformed within a subgroup of $SE(3)$ that represents the solid's available degrees of freedom. This means that configuration products may be used to perform many traditional solid modeling operations such as offsets, sweeps, and Minkowski sums.

But non-identity embeddings allow more general computations with configuration product that may be difficult to formulate any other way. For example, Figure 2 shows the sweep (sampled and triangulated respectively) of a surface A by a surface B , where B is required to maintain contact along normals in A. In this case, surface A is embedded in $SE(3)$ using orientations defined by the surface normal and the directions of the principal curvatures. Surface B is embedded with the identity coordinate system. With physically meaningful embeddings, this computation is useful in many applications, e.g. when computing a multi-parameter sweep of a manufacturing tool (or robot) moving along the workpart surface.

2.2 Products and Quotients

We now highlight some algebraic properties of a configuration product and its dual, configuration quotient operation. The algebra is a natural generalization of properties of Minkowski algebra [32] and of one parameter sweeps [18], but the generalization is complete in the sense that it encompasses properties of all other motion/ rigid transformation related operations on solids.

Configuration products follow basic properties of associativity and distributivity over set union

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \quad (1)$$

$$(A \cup B) \otimes C = (A \otimes C) \cup (B \otimes C) \quad (2)$$

$$A \otimes (B \cup C) = (A \otimes B) \cup (A \otimes C) \quad (3)$$

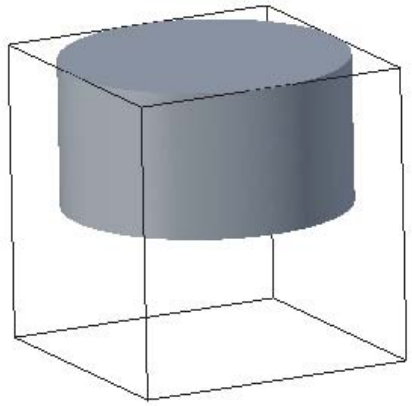

Figure 3: Right configuration quotient of the two parameter family of transformations defined by $[0, \frac{l}{2}] \times SO(2)$, with the cube of side l embedded in the subspace $\mathbb{R}^3 \times I$. The cylinder represents the projection of the quotient and is the set of points that do not leave the cube under any of the transformations in the family

Associativity of the configuration product follows directly from associativity of matrix multiplication. The distributivity of configuration products over set unions is useful for computational purposes because it implies that configuration product may be computed in a piecewise fashion over any partitioned subset of configuration space. This strategy is used in our implementation described in Section 4.

Because the configuration product is a generalization of the Minkowski sum and the sweep operations, it is reasonable to expect that the dual operations of Minkowski difference and unsweep, respectively, also generalize. This is indeed the case, but the lack of commutativity results in two dual operations, which we will call, respectively, *left* and *right quotients*.

Left configuration quotient of sets $A, B \in SE(3)$ is a mapping $f : A \times B \to SE(3)$ defined as

$$A \oslash B = \bigcap_{b \in B} A \cdot b \tag{4}$$

In the special case when $A, B \subset \mathbb{R}^3 \times I$, the left configuration quotient becomes Minkowski difference via projection to $\mathbb{R}^3$. In other words, $\pi(A \oslash B) = \pi(A) \ominus \pi(B)$ where $\ominus$ stands for the Minkowski difference [32]. Intuitively, the left configuration quotient corresponds to iteratively intersecting over the set of configurations to which the set of transformations A is being applied to.

Similarly, *right configuration quotient* of sets $A, B \in SE(3)$ is a mapping $f : A \times B \to SE(3)$ defined as

$$A \oslash B = \bigcap_{a \in A} a \cdot B \tag{5}$$

Intuitively, the right configuration quotient corresponds to iteratively intersecting over the set of all transformations that are being applied to a set of configurations B. Figure 3 shows an example[2] with a cube of side l. Observe that if the sets A, B are not embedded in $\mathbb{R}^3 \times I$, rotations will be factored into the quotients. When A is a path (a one parameter family of transformations) in configuration space and $B \subset \mathbb{R}^3 \times I$, $\pi(A \oslash B) = \texttt{unsweep}(B, \hat{A})$, where $\hat{A} = \bigcup_{a \in A} a^{-1}$ is the inverted trajectory in the configuration space (see further explanation in section 2.3 and [18] for detailed analysis of the `unsweep` operation).

In general, $A \oslash B \neq A \oslash B$, but strictly speaking, only one of the quotients is needed, because each quotient may be expressed in terms of the other. The precise nature of this relationship follows the general duality property that generalizes the well known dualities of Minkowski operations and `sweep`/`unsweep` relationship.

Duality Property.$(A \oslash B^c)^c = A \otimes B = (A^c \oslash B)^c$.
Proof. Use de Morgan's laws to prove the statements.

$$\bigcup_{a \in A} a \cdot B = A \otimes B = \bigcup_{b \in B} A \cdot b \tag{6}$$

$$\Rightarrow \bigcup_{a \in A} (a \cdot B^c)^c = A \otimes B = \bigcup_{b \in B} (A^c \cdot b)^c \tag{7}$$

$$\Rightarrow (\bigcap_{a \in A} (a \cdot B^c))^c = A \otimes B = (\bigcap_{b \in B} (A^c \cdot b))^c \tag{8}$$

$$\Rightarrow (A \oslash B^c)^c = A \otimes B = (A^c \oslash B)^c \tag{9}$$

This property is important because it holds for *arbitrary* sets of transformations in $SE(3)$. Additionally, the property implies that it is sufficient to compute just one of $\otimes, \oslash, \oslash$ since the other two may be directly derived via complementation. In particular, it should be clear that $A \oslash B = (A^c \oslash B^c)$, but we choose to retain both left and right quotient because both arise in applications frequently and independently, to reflect very distinct physical constraints. The three operations are related by a number of useful algebraic equalities which we state here without the proofs, that amount to straightforward algebraic substitutions.

$$(A \oslash B) \oslash C = A \oslash (B \otimes C) \tag{10}$$

$$A \oslash (B \oslash C) = (A \otimes B) \oslash C \tag{11}$$

$$(A \cap B) \oslash C = (A \oslash C) \cap (B \oslash C) \tag{12}$$

$$A \oslash (B \cap C) = (A \oslash B) \cap (A \oslash C) \tag{13}$$

$$(A \cap B) \oslash C = (A \oslash C) \cap (B \oslash C) \tag{14}$$

$$A \oslash (B \cap C) = (A \oslash B) \cap (A \oslash C) \tag{15}$$

[2]The transformations in $[0, \frac{l}{2}] \times SO(2)$ correspond to all matrices in the two parameter family

$$\begin{pmatrix} cos(\theta) & -sin(\theta) & 0 & 0 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

for $0 \leq \theta \leq 2\pi$ and $0 \leq x \leq \frac{l}{2}$

2.3 Set and topological properties

The algebraic and duality properties help us understand some basic topological properties of sets constructed from configuration products and quotients. In particular, when A, B are compact sets and embedded in subspaces of dimensions m, n in $SE(3)$, it is clear that the dimension of $A \otimes B$ (represented as $dim(A \otimes B)$ is at most $dim(A \times B)$. For example, if a three dimensional set $A \subset \mathbb{R}^3 \times I$ is combined with a one dimensional set $B \subset I \times SO(2)$, the product $A \otimes B$ is clearly a subset of the four dimensional subspace $\mathbb{R}^3 \times SO(2)$. Since $dim(A \times B) \leq dim(A) + dim(B)$ [37] and $dim(A \otimes B) \leq 6$, the dimension of $A \otimes B$ is the lesser of $m + n$ and 6. In addition, if A and B are both embedded in the same subspace of $SE(3)$ with dimension $p \leq 6, dim(A \otimes B) = p$. Similar compactness and dimensionality arguments extend to quotients, by duality. It is also true that if A, B are connected, then $A \otimes B$ will be connected, but the reverse statement is not necessarily true: it is possible to choose one of A, B as a disconnected set such that $A \otimes B$ is connected. By duality, the quotient of a connected set $A \otimes B$ with a connected set may be disconnected.

An important utility of `unsweep` and Minkowski difference that makes them particularly useful for motion planning, packaging problems, and shape design, is their ability to define maximal configuration spaces that satisfy the specified containment constraints. These include maximal shapes contained within an envelope under specified one parameter motion, or maximal translations of a shape to stay within an envelope. Since quotients are generalizations of the Minkowski difference and `unsweep`, analogous properties without imposing restrictions on the allowed transformations would make them the operator of choice to formulate all spatial containment problems.

Containment Property Given sets $B, C \subset SE(3)$, $A_1 = C \oslash \hat{B}$ is the largest subset of C such that $A_1 \otimes B \subseteq C$. Moreover, $A_2 = \hat{B} \oslash C$ is the largest subset of C such that $B \otimes A_2 \subseteq C$.

Proof. Given $B, C \subset SE(3)$ the set A_1 of configurations such that $A_1 \otimes B \subseteq C$ is given by

$$A_1 = \{a | a \cdot b \in C, \quad \forall b \in B\} \tag{16}$$

It follows that for all $a \in A_1$, $a \in C \cdot b^{-1}$ for all $b \in B$. This implies $A_1 = \bigcap_{b \in B} C \cdot b^{-1}$ implying $A_1 = C \oslash \hat{B}$. Similar arguments show that $A_2 = \hat{B} \oslash C$ is the largest subset of C such that $B \otimes A_2 \subseteq C$.

Suppose A_2 is the set to be computed so that $B \otimes A_2 \subset C$ where B is a motion, and C is a subset of $\mathbb{R}^3 \times I$, then $\pi(A_2) = \pi(\hat{B} \oslash C) =$ `unsweep`(C, B). Also, for sets $C, B \subset \mathbb{R}^3 \times I$ the maximal set of transformations A_1 such that $A_1 \otimes B \subseteq C$ is defined by the property as $A_1 = C \oslash \hat{B}$. Since both $C, B \subset \mathbb{R}^3 \times I$, A_1 is a set of translations and therefore $C \oslash \hat{B} = C \ominus \hat{B}$ is the well known erosion operation [32]. More generally, when B and/or C are embedded in a 6d subset of $SE(3)$, $A_1 = C \oslash \hat{B}$ represents the set of all transformations which, when applied to B, do not result in a collision with C. The problem of finding a collision free path for a robot in the presence of obstacles [21] is now equivalent to finding a path between starting/ending configurations in the free space $A_1 = C \oslash \hat{B} = (C^c \otimes \hat{B})^c$, which is the complement of the configuration space obstacle. A preferred method of implementing this computation usually relies on the dual formulation in terms of $\otimes$.

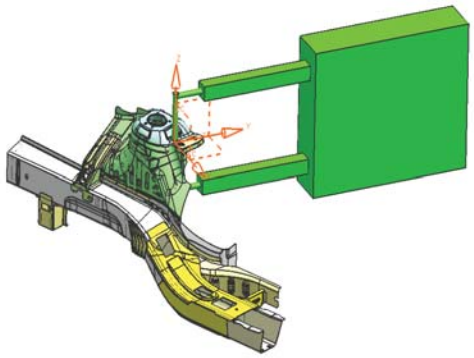

Figure 4: A typical work assembly and weld gun

3. EXAMPLES OF DIRECT AND INVERSE PROBLEMS IN MANUFACTURING

In this section we will demonstrate applications of configuration products and quotients by formulating and solving direct and inverse problems in welding process planning for a typical automotive assembly plant. An important requirement of any robot assisted welding process is to ensure that a set of weld locations on a work part can be reached by the robot weld gun assembly without colliding with the work part or surrounding tooling. The general problem of finding a path for the robot carrying the weld gun in the presence of multiple obstacles may be formulated via the containment property as discussed in Section 2. We will now formulate and solve related simpler but important problems. Putting the robot motion planning problem aside, we assume that once a weld location is reached, a *valid contact* between a gun and work-part occurs along the part surface normal at the weld location. This is a common manufacturing constraint for spot welding, which implies that, at any given location, the the gun has only one degree of freedom corresponding to rotation about the normal to maintain valid contact. A typical welding setup is shown in Figure 4.

Direct problem: surface sweep. Typically, a set of weld locations is defined on the work part and the space occupied by a weld gun at these locations needs to be determined in order to plan the placement of surrounding tooling to avoid collisions with the gun. Some static tooling may be pre planned in order to partially fixture the work part but movable tooling is generally planned in areas inaccessible by the weld gun if possible. Since it is practically not possible to place a weld gun at a weld location repeatedly in exactly one orientation, the weld gun is allowed a clearance range of orientations at each location. The clearance range at each weld location is chosen sufficiently small to avoid/minimize collisions with the work assembly and can be represented as a range of planar rotational transformations (for valid contact). Denoting the embedding of the weld gun in $\mathbb{R}^3 \times I$ as G, the clearance range of rotations in $SO(2)$ about the

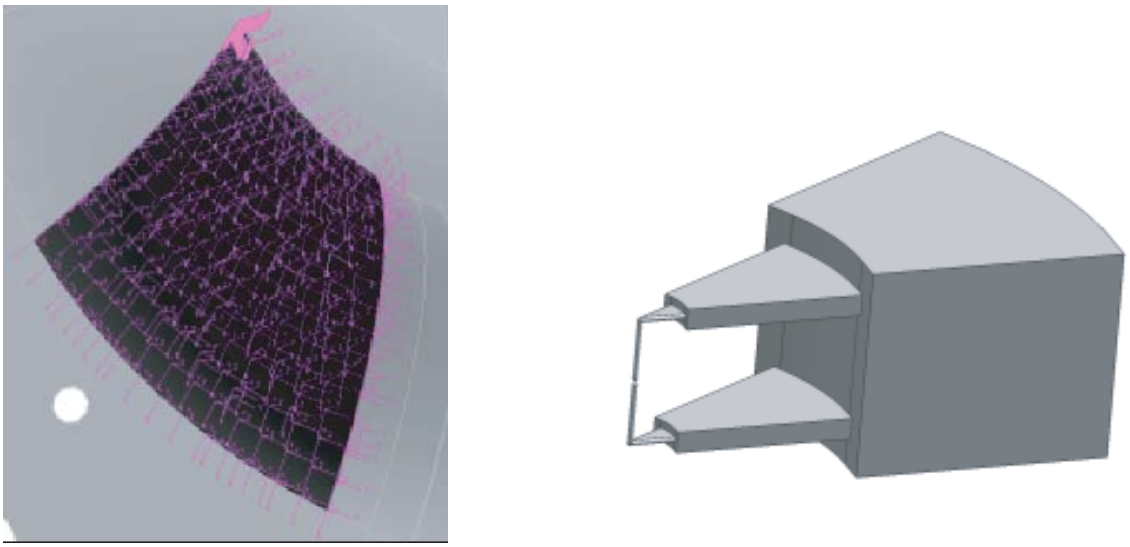

Figure 5: Left: A patch of a surface on which the weld gun is constrained to move while maintaining normal contact. Coordinate frames are defined at each location on the surface by the surface normal and the directions of the principal curvatures. Right: The weld gun in Figure 4 swept by the clearance range of orientations

weld gun electrode (aligned with the part surface normal at the weld location) as C, the $6d$ space occupied by the weld gun at any given location is the configuration product $C \otimes G$ whose projection into Euclidean space is simply a rotational sweep as shown in Figure 5 (right). Suppose now that T is the set of transformations in $\mathbb{R}^3 \times SO(3)$ that locate the weld locations and orient the gun with the surface normal. Such a set of transformations can be defined as discussed in Section 2 by associating a coordinate system at each weld location defined by the surface normal and two other orthogonal vectors whose cross product is the surface normal. Then the total space in $SE(3)$ occupied by the weld gun is the configuration product $T \otimes C \otimes G$. Note that the configuration product is six dimensional in this case, and the same formulation applies to finite and infinite sets of weld locations. If T is defined by a surface to be welded, then the configuration product in this case defines the sweep of the weld gun over surface, with orientation controlled by the surface normal. Figure 6 shows a sparse sampling of such a swept solid realized as the projection $\pi(T \otimes C \otimes G)$. The primitive sets T and $\pi(C \otimes G)$ are shown in Figure 5.

Inverse problem: feasible space of a weld gun. We will say that a configuration of a weld gun is *feasible* if the gun does not collide with the work assembly while maintaining valid surface contact at that location. The set of all such configurations is called the *feasible space* at the weld location. Computing the feasible space of a weld gun at a single weld location may be formulated as an inverse problem. To do so, the work assembly is embedded as the set $W \subset \mathbb{R}^3 \times SO(2)$ by associating all rotations in $SO(2)$ to every position and the gun is embedded as the set $G \subset \mathbb{R}^3 \times I$ to ensure that any transformation applicable to the weld gun is within $\mathbb{R}^3 \times SO(2)$. The set of all transformations T of a weld gun G that avoid collisions with the work part W must satisfy

$$T \otimes G \subseteq (iW)^c \tag{17}$$

where iW represents the interior of the work part since contact with the points on the boundary ∂W of the work part must be allowed. Using the containment property, the set of all non colliding transformations T is

$$T = (iW)^c \oslash \hat{G}, \tag{18}$$

which has the dimensionality of $\mathbb{R}^3 \times SO(2)$. These transformations correspond to all translations and planar rotations of the weld gun that do not collide with the work part. Of particular interest are the transformations at weld locations with the surface normal corresponding to the axis of rotation defining $SO(2)$. The subset of T at any position on ∂W with such normals is the feasible space at that position.

It is straightforward to extend the above formulation to computing gun's feasible transformations over surfaces of an arbitrary workpart. The work part must now be embedded in all of $\mathbb{R}^3 \times SO(3)$, because the weld gun is now treated as an object with six degrees of freedom. However not all points in the embedding have orientations corresponding to all possible rotations. The boundary of the work part must be embedded into $SE(3)$ in such a way that the possible orientations at each point must correspond to planar rotations about the part surface normal to maintain valid contact. All other points in $\mathbb{R}^3$ are embedded with all possible rotations because every position in the complement of the work part boundary can be reached in multiple orientations. Using exactly the same formulation as described for the feasible space at a single location, the set of all possible valid non colliding configurations is given by the same expression (18). The rotations of elements in T at each position on the boundary correspond to the feasible space at that location.

The above formulation reduces to well known and easily

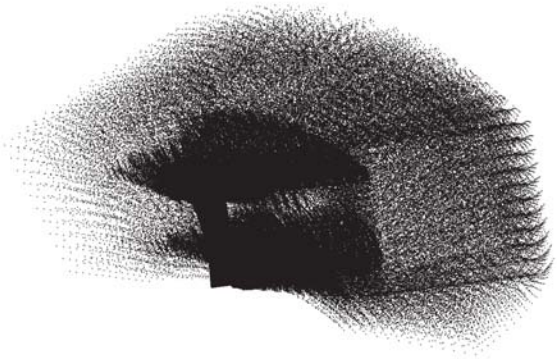

Figure 6: The swept gun moving over the part surface is a projection of a configuration product into Euclidean space visualized as a sparse point cloud

computable special cases. Expression (18) is simply an expression for the free space (its dual is the complement of the configuration space obstacle). On the other hand, expression (17) can be applied directly in situations where $T \otimes G$ can be computed. For example, at a single weld location, this expression corresponds to the intersection of gun's rotational sweep by 360° and the set $(iW)^c$. A more detailed explanation of this procedure and supporting computations may be found in [27].

Inverse problem: feasible weld gun. Constructing the feasible space is a key utility in verifying the weld gun's performance with respect to a particular process plan. But it is less useful when the number of weld guns is large or when the weld gun needs to be (re)designed. In this case, given the work part and tooling W, a weld location, and some desired range of contact/clearance configurations C, we want to determine spatial constraints on the shape of the gun X. The work part is embedded in $\mathbb{R}^3 \times SO(2)$ by associating all of $SO(2)$ with each position because C must be a set of planar rotations. The maximal weld gun X is a set of all configurations that must be contained within $(iW)^c$ under all transformations in C, i.e.

$$C \otimes X \subseteq (iW)^c, \tag{19}$$

implying that

$$X \subseteq \hat{C} \oslash (iW)^c. \tag{20}$$

When C is a one parameter set of rotations, it follows that $\pi(X) = \texttt{unsweep}((iW)^c, M)$. In a CAD system, it is more practical to compute $\pi(X)$ as using the dual `sweep` operation simply because W is a solid (R-set) and W^c is not. Figure 7 shows an approximation of $\texttt{sweep}(W, \hat{M})$ as a finite union of work part configurations.

Once again, the formulation applies to more general sets with minimal modifications. A maximal feasible weld gun X for a surface of work part W is defined by the same expressions, provided that W is embedded in all of $\mathbb{R}^3 \times SO(3)$, as described above.

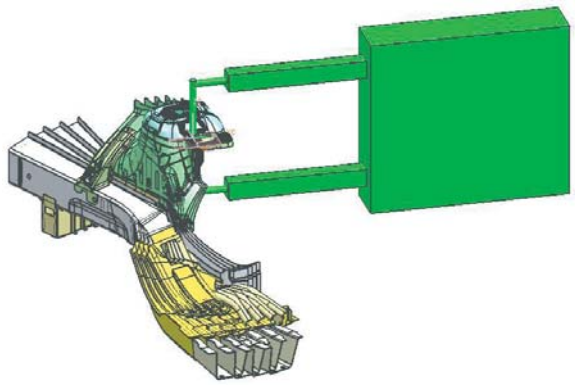

Figure 7: $\texttt{unsweep}(W^c, M)$ computed as the dual $\texttt{sweep}(W, \hat{M})$. M is a 30 degree rotation about the part surface normal at the weld location

4. GPU IMPLEMENTATION VIA SAMPLING

Computing the configuration product exactly for general subsets $A, B \subset SE(3)$ is clearly quite challenging. Commercial CAD systems provide only limited functionality to compute one parameter sweeps (typically for 2D cross sections), and no support for other classes of configuration products. Approximating configuration products and quotients via finite unions and intersections of solids becomes impractical for non trivial shapes (such as the work part/weld gun) and transformations (such as a sweep over the surface). Therefore in order to compute and visualize general configuration products, we must look for alternate computational strategies that do not rely significantly on CAD systems. Even for the well understood special case of Minkowski sums of polyhedra, exact solutions are rare [19, 17], and approximate sampling techniques have been widely advocated [20, 36, 22].

4.1 Sampling strategy

We adopt the sampling solution for computing configuration products, in several stages that are similar to those described in [22] for computing Minkowski sums. First, discrete approximations of the primitive sets are obtained by sampling them in configuration space. Then pairwise matrix multiplications of the sampled configurations are computed rapidly on the Graphics Processing Unit (GPU), yielding a discrete approximation of the configuration product, that can be visualized by several methods, depending on the dimension of the computed result. When the final set of configurations represent a three-dimensional solid, it is visualized by projecting the sampled configuration product into $\mathbb{R}^3$, filtering out interior points in the projection, and reconstructing a water-tight surface homeomorphic to the sampled surface around the remaining points. Figure 9 shows the pipeline of computations required to represent a Minkowski sum as a projection of the configuration product of two sets. The same pipeline was used to compute Figures 1, 2, 9.

Primitive sampled sets in configuration space are generated by sampling discrete configurations extracted from the underlying set parameterizations. To sample solids embedded in configuration space we use the parameterizations in the solid's boundary representation. Sampling the embedding

of solids in $SE(3)$ is equivalent to sampling the solids in $\mathbb{R}^3$ and embedding the sampled set in $SE(3)$ by associating every position with the appropriate orientation. For all other sets of transformations, an underlying parametrization is expected and a discretization is obtained by sampling configurations corresponding to discrete parameter values. It is important to sample the primitive sets densely in order to obtain a faithful representation of their configuration product. In order to make the notion of a dense sampling of configurations more precise, it is necessary to have a distance function in $SE(3)$.

Since $SE(3)$ is a Riemannian manifold [39], any two configurations $x, y \in SE(3)$ can be joined by a length minimizing geodesic such that the length of the geodesic is the Riemannian distance function [5] $d(x,y) = \| ln(x^{-1}y) \|$ where ln represents the matrix logarithm [10] and the norm chosen is the Frobenius norm. Under this metric, the distance between two points in $\mathbb{R}^3 \times I$ becomes the standard Euclidean distance. The distance between any two points in $SE(3)$ is always greater or equal to the distance between their projections into Euclidean space, or any other subspace. For the purposes of this paper we will adopt the Riemannian distance function as the metric on $SE(3)$, while observing that other metrics are possible [39] and are useful for different applications.

Once the metric is selected, the notion of a sufficiently dense sampling may be formalized by defining the $\epsilon-$*ball* centered at $a \in A$ for any $A \subset SE(3)$

$$B_\epsilon(a) = \{x \in A \mid d(a,x) < \epsilon\} \tag{21}$$

Representing the sampling of a set A as S_A, following [22] we say that S_A is an ϵ-*covering* of A if for every point $x \in A$, there exists a point $a \in S_A$ such that $d(a,x) < \epsilon$. It follows that

$$A = \bigcup_{a \in S_A} B_\epsilon(a) \tag{22}$$

$$A \otimes B = \bigcup_{\substack{a \in S_A \\ b \in S_B}} B_\epsilon(a) \otimes B_\epsilon(b) \tag{23}$$

Given ϵ-coverings S_A, S_B, the product $S_A \otimes S_B$ will be an $\epsilon-$covering of $A \otimes B$ if and only if

$$\bigcup_{\substack{a \in S_A \\ b \in S_B}} B_\epsilon(ab) = A \otimes B \tag{24}$$

In turn, Equation 24 will be satisfied if and only if the following relations are satisfied

$$\bigcup_{\substack{a \in S_A \\ b \in S_B}} B_\epsilon(ab) \subset A \otimes B \tag{25}$$

$$\bigcup_{\substack{a \in S_A \\ b \in S_B}} B_\epsilon(ab) \supset A \otimes B \tag{26}$$

Equation 25 always holds true by construction. However, Equation 26 is not satisfied in general because it is often possible to pick some $x \in A \otimes B$ such that the distance to any $y \in B_\epsilon(ab)$ for all $a \in S_A, b \in S_B$ is greater than ϵ. More formally, the deviation from an $\epsilon-$covering induced on approximating $A \otimes B$ by the product of $\epsilon-$coverings S_A, S_B is bounded. It is sufficient to establish this bound for the case when both $S_A = \{a\}$ and $S_B = \{b\}$ are single point $\epsilon-$coverings. In such a case $S_A \otimes S_B = ab$ is the approximation of $A \otimes B$. Consider a point $a_* \in B_\epsilon(a)$ such that $d(a, a_*) < \epsilon$. The distance of the point a_*b from the transformed center ab is given by

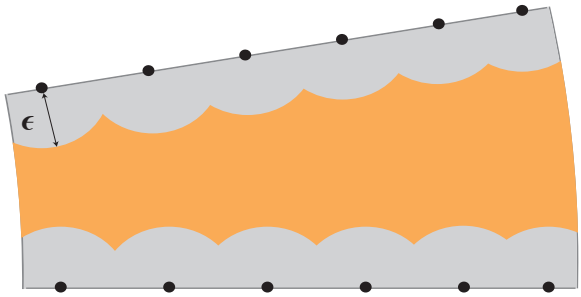

Figure 8: A is a one parameter family of planar rotations approximated by an $\epsilon-$covering S_A consisting of two rotation matrices. B is a line segment along the $x-$axis far from the origin approximated by an $\epsilon-$covering S_B of its embedding in $SE(3)$. The figure shows the projection $\pi(A \otimes B)$ which is the one parameter sweep consisting of the union of all shaded (gray and orange) regions. The figure also shows the projection $S_A \otimes S_B$ which is all the sample points. When $A \otimes B$ is approximated by $S_A \otimes S_B$ the regions in orange correspond to points in $\pi(A \otimes B)$ for which the distance to the closest projected sample point is greater than ϵ.

$$d(ab, a_*b) = \| ln((ab)^{-1} a_* b) \| \tag{27}$$

$$= \| ln(b^{-1} a^{-1} a_* b) \| \tag{28}$$

In order to simplify Equation 28 and thereby compute the bound, we exploit a standard fact connecting Lie groups and Lie algebras. Every Lie group acts on itself by the group conjugation $g^{-1}xg$ for group elements g, x which can be expressed as [10, 16]

$$g^{-1} e^{\mathsf{X}} g = e^{g^{-1}\mathsf{X}g} \tag{29}$$

where $\mathsf{X} = ln(x)$ and $e^{\mathsf{X}} = x$ represents its matrix exponential. Therefore

$$\| ln(b^{-1}a^{-1}a_*b) \| = \| ln(e^{b^{-1} ln(a^{-1}a_*) b}) \| \tag{30}$$

$$= \| b^{-1} ln(a^{-1}a_*) b \| \tag{31}$$

$$\leq \| ln(a^{-1}a_*) \| \| b \|^2 \tag{32}$$

$$\leq \epsilon \| b \|^2 \tag{33}$$

We have used the fact that $\| xy \| \leq \| x \| \| y \|$ and that for any rigid transformation x, $\| x \| = \| x^{-1} \|$. Thus the distance of any point $a_*b \in B_\epsilon(a) \otimes b$ to the transformed sample point ab is bounded by $\epsilon \| b \|^2$. Figure 8 illustrates a situation where the product of two $\epsilon-$coverings is not an $\epsilon-$covering, and shows the increasing deviation from the $\epsilon-$covering for transformations further from the origin.

In the special case when both A, B belong to an Abelian (commutative) subgroup of $SE(3)$ namely translations ($\mathbb{R}^3$) or planar rotations ($SO(2)$), owing to the commutative relation $xy = yx$ it follows that $y^{-1}xy = x$. Therefore Equation

28 immediately reduces to

$$\| ln(b^{-1}a^{-1}a_*b) \| = \| ln(a^{-1}a_*) \| = \epsilon \quad (34)$$

Thus in this special case, $B_\epsilon(a) \otimes b$ will always be an $\epsilon-$ball and the product $S_A \otimes S_B$ will always be an $\epsilon-$covering of $A \otimes B$. This is verified for the case when $A \otimes B$ is the Minkowski sum in [22]. In all other cases when $B_\epsilon(a) \otimes b$ will not be an $\epsilon-$ball, it is prudent to either choose a very fine sampling density such that $\epsilon \to 0$, or scale the transformations closer to the origin such that $\epsilon \| b \|^2 \approx \epsilon$.

4.2 Product on a GPU

Once the sets are sampled, we now directly follow the definition of the configuration product and compute pairwise matrix multiplications of elements (configurations) in the primitive sets. Each pairwise multiplication is independent of others, allowing straightforward parallelization of the configuration product using the Single Instruction Multiple Data (SIMD) computational model. The GPU is especially well suited to address such problems that can be expressed as data-parallel computations, where the same program is executed on many data elements in parallel with high arithmetic intensity (the ratio of arithmetic operations to memory operations). Our prototype implementation relies on NVIDIA's CUDA API [28] and is intended to demonstrate feasibility of the configuration product computations.

The parallelism is at the thread level where each thread computes a unique pairwise product and writes back into a unique position in global memory. In our implementation we have assumed the rotation component of each transformation is represented by a 3×3 orthogonal matrix and the translational component by a 3×1 vector. The distributive properties of the configuration product allow partitioning the sampled sets, and to compute the results in several data input/output cycles with the GPU operating at full capacity.

For the Minkowski sum shown in Figure 9, the product $S_A \otimes S_B$ contained 1350195 points and was computed in one pass on the GPU in 71.816 ms including all global memory copies. GPU Computations were performed on an NVIDIA GTX 280 GPU. Using the described representation of transformation matrices the GPU can compute roughly 22 million pairwise products in parallel for one data input/output cycle. Our current implementation is simple but is sufficient to demonstrate the practicality of approximating the configuration product by simple pairwise matrix multiplications on the GPU. The basic GPU computation of the configuration product can be improved in several ways. It is possible to increase the number of matrices computed in each pass by representing each rotation matrix with three parameters such as Euler angles, or four parameters using unit quaternions. Computational efficiency can also be increased if we represent the configuration product as the convolution of two sets which can be computed using the Fast Fourier Transform similar to the algorithm described in [20] which has complexity $O(n\ log\ n)$ as opposed to $O(n^2)$ for the pairwise multiplication implementation, and can be implemented in parallel. However, this algorithm has large storage requirements making it somewhat impractical for the computing products in full 6d configuration space.

4.3 Visualization

The subsets of configuration space, both sampled and computed, can be used directly for membership queries and planning purposes. Visualizing these 6d sets is more challenging, but reduces to visualization of their projections into $\mathbb{R}^3$ and $SO(3)$. The projection of the sampled configuration product in $\mathbb{R}^3$ is obtained by simply extracting the translational components of the matrices.

The resulting projection in Euclidean space is a dense 3D set that contains both interior and boundary points. The interior points are not required to visualize the boundary of the swept solid generated by the configuration product, and can be filtered out using variety of techniques. For our implementation, we used the simple flood fill algorithm commonly used in computer graphics [14] to remove all points whose neighborhoods ("buckets") are interior to the set of points. More sophisticated algorithms can be used to filter out interior points in the special case of computing the Minkowski sum as demonstrated in [22]. Depending on the bucket sizes used to filter the points, the remaining set of retained boundary points correspond to a noisy sampling of set's boundary. A number of techniques can be used to reconstruct the triangulated boundary from this sampling. We used the robust cocone algorithm developed by Dey and Goswami [12] to generate bounding surfaces in Figures 1, 2, 9. All CPU computations were performed on an Intel Core 2 Duo workstation (3GHz, 3GHz, 2.75 GB RAM). We have not implemented a method for visualizing the orientation of configurations, but a number of approaches have been suggested. For example, rotations are commonly visualized in terms of the angle-axis parameterization that is easily obtained from the eigenvectors and eigenvalues of the rotation matrix [2].

5. CONCLUSIONS

The main contribution of this paper is to show that configuration products and quotients generalize and unify a variety of computations in solid modeling involving motions and relative configurations. By treating solids and rigid transformations as point sets in the configuration space SE(3), we have shown that most formulations of kinematic planning/design problems are derivable from this general formulation. The usual Minkowski operations and variety of sweep operations are special cases of configuration products and quotients, but many useful modeling operations require full generality of configuration space algebra. These include sweeps over manifolds (curves, surfaces), design of moving shapes, and determination of feasible (configuration) spaces for shapes under general motions.

We have also shown that the formulation in the configuration space should not be viewed as an impediment to practical implementations. Sampling the configuration products and quotients is computationally tractable using parallel algorithms and architectures, as is clearly demonstrated by our brute force preliminary GPU implementation. It may be possible to further expedite the computation of configuration products using algorithms described in [20, 9]. We note that many steps in the sampling reconstruction process may be improved further; for example, marching cubes algorithm to compute a triangulation of a surface is now available with the NVIDIA CUDA SDK and could be adapted to compute

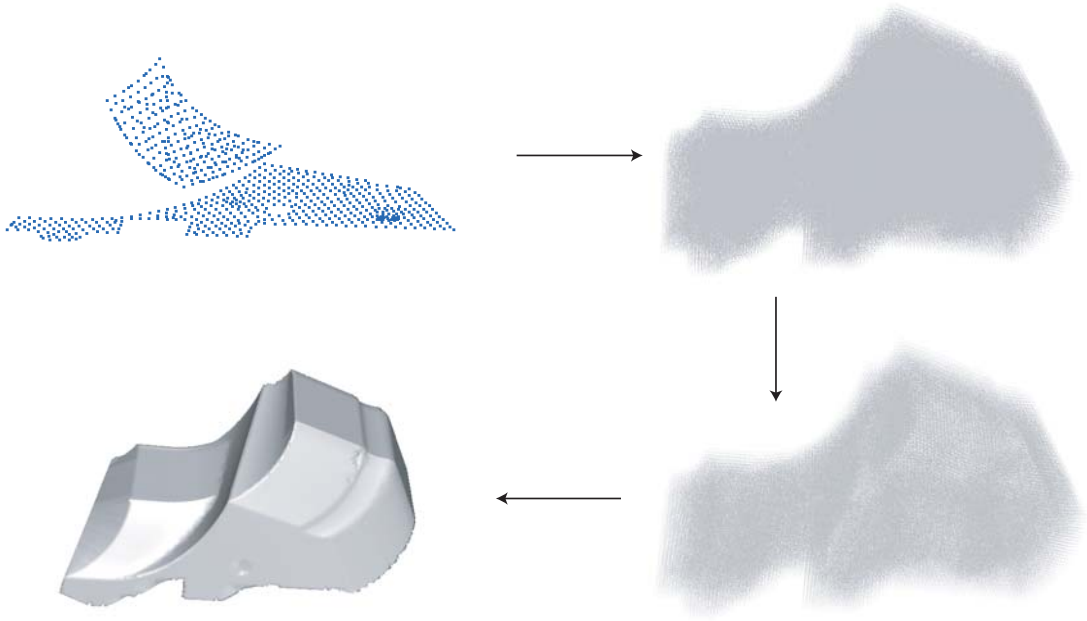

Figure 9: Approximate Minkowski sum computed as a configuration product of samplings of surfaces. Top right: Projection of the point set obtained after computing the product on the GPU, computation time = 0.07186 s. Bottom right: The remaining points after filtering out interior points using flood fill, computation time = 60 s. Bottom left: Water-tight surface reconstructed using the robust cocone software, reconstruction time = 337.61 s. The total time to compute the boundary of the Minkowski sum of the two sampled surfaces = 397.681 s.

surfaces from noisy point data.

The theoretical generality, computational tractability, and wide scope of applications of the configuration space algebra suggest that it should become an integral part of future solid modeling and CAD systems. Our preliminary results also point to a number of important research issues. These include recasting fundamental solid modeling algorithms directly in configuration space, investigating how the properties of configuration space algebra can improve modeling techniques, development of efficient parallel algorithms, and fleshing out specific applications in computer aided design and manufacturing.

6. ACKNOWLEDGEMENTS

We would like to thank Tamal Dey and Joshua Levine for permitting the use of, and suggesting useful visualization strategies for their robust cocone software. We would also like to thank Mikola Lysenko, Tom Grim, and Atul Abhyankar for helping with aspects of implementation and data generation. Finally we would like to thank reviewers for their comments. This research was supported in part by the National Science Foundation grants CMMI-0500380 and CMMI-0621116.

7. REFERENCES

[1] S. Agrawal. Workspace boundaries of in-parallel manipulator systems. In *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1147–1152, 1991.

[2] B. Alpern, L. Carter, M. Grayson, and C. Pelkie. Orientation maps: techniques for visualizing rotations (a consumer's guide). In *Proceedings of the 4th conference on Visualization'93*, pages 183–188, 1993.

[3] J. Boissonnat, E. De Lange, and M. Teillaud. Minkowski operations for satellite antenna layout. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 67–76. ACM New York, NY, USA, 1997.

[4] R. Brost and K. Goldberg. A complete algorithm for designing planar fixtures using modular components. *Robotics and Automation, IEEE Transactions on*, 12(1):31–46, Feb 1996.

[5] K. Burns and M. Gidea. *Differential geometry and topology: with a view to dynamical systems*. Chapman & Hall/CRC, 2005.

[6] M. Caine. The design of shape from motion constraints, PhD. Thesis, 1993.

[7] M. Caine. The design of shape interactions using motion constraints. In *1994 IEEE International Conference on Robotics and Automation, 1994. Proceedings.*, pages 366–371, 1994.

[8] J. Canny. *The complexity of robot motion planning*

ACM doctoral dissertation awards. MIT Press, 1988.

[9] G. Chirikjian and A. Kyatkin. Engineering Applications of Noncommutative Harmonic Analysis, 2001.

[10] M. Curtis. *Matrix Groups*. Springer, 1984.

[11] J. Denavit and R. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22(2):215–221, 1955.

[12] T. Dey and S. Goswami. Provable surface reconstruction from noisy samples. In *Proceedings of the twentieth annual symposium on Computational Geometry*, pages 330–339. ACM New York, NY, USA, 2004.

[13] B. Donald. A search algorithm for motion planning with six degrees of freedom. *Artificial Intelligence*, 31(3):295–353, 1987.

[14] J. Foley and A. Van Dam. *Fundamentals of interactive computer graphics*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1982.

[15] P. Ghosh and P. K. Jain. An algebra of geometric shapes. *IEEE Computer Graphics and Applications*, 13(5):50–59, 1993.

[16] D. Gurarie. *Symmetries and Laplacians: Introduction to Harmonic Analysis, Group Representations, and Applications*. North Holland, 1992.

[17] P. Hachenberger. Exact Minkowksi sums of polyhedra and exact and efficient decomposition of polyhedra in convex pieces. *Lecture Notes in Computer Science*, 4698:669, 2007.

[18] H. Ilies and V. Shapiro. Dual of sweep. *Computer Aided Design*, 31(3):185–201, 1999.

[19] A. Kaul and J. Rossignac. Solid-interpolating deformations: construction and animation of PIPs. *Computers & graphics*, 16(1):107–115, 1992.

[20] L. Kavraki. Computation of configuration-space obstacles using the fast Fourier transform. *IEEE Transactions on Robotics and Automation*, 11(3):408–413, 1995.

[21] J. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[22] J. Lien. Covering Minkowski sum boundary using points with applications. *Computer Aided Geometric Design*, 25(8):652–666, 2008.

[23] T. Lozano-Perez. Spatial planning: A configuration space approach. *IEEE transactions on computers*, 100(32):108–120, 1983.

[24] K. Malek, J. Yang, D. Blackmore, and K. Joy. Swept volumes: Foundations, perspectives, and applications. *International Journal of Shape Modeling.*, 2(1):87–127, 2004.

[25] J. McCarthy and L. Joskowicz. Kinematic synthesis. *Formal Engineering Design Synthesis*, page 321, 2001.

[26] A. Middleditch. Application of vector sum operator. *Computer-Aided Design*, 20(4):183–188, 1988.

[27] S. Nelaturi, A. Abhyankar, V. Shapiro, and R. Tilove. Feasible spaces in weld gun selection. In *IEEE International Conference on Automation Science and Engineering, 2008. CASE 2008*, pages 870–875, 2008.

[28] NVIDIA. *Compute Unified Device Architecture Programming Guide*, 2007.

[29] A. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys*, 12(4):437–464, December 1980.

[30] E. Sacks and C. Bajaj. Sliced configuration spaces for curved planar bodies. *The International Journal of Robotics Research*, 17(6):639, 1998.

[31] J. Selig. *Geometrical Methods in Robotics*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1996.

[32] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc. Orlando, FL, USA, 1983.

[33] V. Shapiro. Solid modeling. *Handbook of Computer Aided Geometric Design*, 2001.

[34] S. Spitz, A. Spyridi, and A. Requicha. Accessibility analysis for planning of dimensional inspection withcoordinate measuring machines. *IEEE Transactions on robotics and automation*, 15(4):714–727, 1999.

[35] A. Spyridi and A. Requicha. Accessibility analysis for the automatic inspection of mechanicalparts by coordinate measuring machines. In *1990 IEEE International Conference on Robotics and Automation, 1990. Proceedings.*, pages 1284–1289, 1990.

[36] G. Varadhan and D. Manocha. Accurate Minkowski sum approximation of polyhedral models. *Graphical Models*, 68(4):343–355, 2006.

[37] M. Wage. The dimension of product spaces. *Proceedings of National Academy of Sciences*, 75(10):4671–4672, 1978.

[38] K. Wise and A. Bowyer. A survey of global configuration-space mapping techniques for a single robot in a static environment. *The International Journal of Robotics Research*, 19(8):762, 2000.

[39] M. Zefran, V. Kumar, and C. Croke. Metrics and connections for rigid-body kinematics. *International Journal of Robotics Research*, 18(2):243–258, 1999.