ARTICLE IN PRESS

Computer-Aided Design ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Configuration products and quotients in geometric modeling

Saigopal Nelaturi, Vadim Shapiro *

Spatial Automation Laboratory, University of Wisconsin-Madison, 53706 Madison, United States

ARTICLE INFO

Keywords:
Configuration space
Minkowski sum
Sweeps
Geometric reasoning
Shape synthesis

ABSTRACT

The six-dimensional space $\mathbb{SE}(3)$ is traditionally associated with the space of configurations of a rigid solid (a subset of Euclidean three-dimensional space $\mathbb{R}^3$). But a solid itself can be also considered to be a set of configurations, and therefore a subset of $\mathbb{SE}(3)$. 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 $\mathbb{R}^3$. 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 approximated using a fast parallel sampling method on a GPU and provide error estimates for the approximation.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

1.1. Shape in configuration space

Swept sets (or sweeps) are one of the fundamental representation schemes in geometric and solid modeling [1]. Generally a swept solid 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 [2]. 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 survey in [3]. Most known methods assume a particular representation of the point set B and/or of the trajectory A, and tend not to be broadly applicable.

The Minkowski sum [4,5] 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. See Fig. 1 for an example. In this sense, the Minkowski sum can also be considered a sweep $g(A, B) = A \oplus B$, which is both generalized because A is no longer limited to a one-parameter family of transformations, 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 being restricted to translational motions/configurations, Minkowski operations are frequently applied in motion planning [6,7], containment and packaging [8] layout [9], image processing [4] and many other graphics and shape modeling applications [5,10]. Their popularity is largely due to the rich algebraic structure that forms the foundation of mathematical morphology [4]. The same algebraic structure has been shown to exist for the traditional sweeps [11], reinforcing the close relationship between sweeps and Minkowski operations.

In order to unify various 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 set of coordinate frames 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 $\mathbb{R}^3 \times \mathbb{SO}(3)$ [7]. The extension removes the artificial distinction between shapes and their transformations because both are now subsets of the configuration space. Furthermore, this view leads to the generalization of a swept set.

* Corresponding author.
E-mail addresses: saigopan@cae.wisc.edu (S. Nelaturi), vshapiro@engr.wisc.edu (V. Shapiro).

0010-4485/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2010.09.007

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

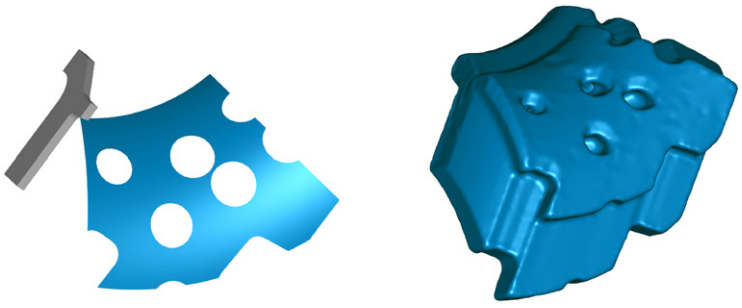

Fig. 1. Minkowski sum (shown on the right) of a solid and a surface (shown on the left) computed as a projection of the configuration product. Features from both input shapes can be seen in the Minkowski sum.

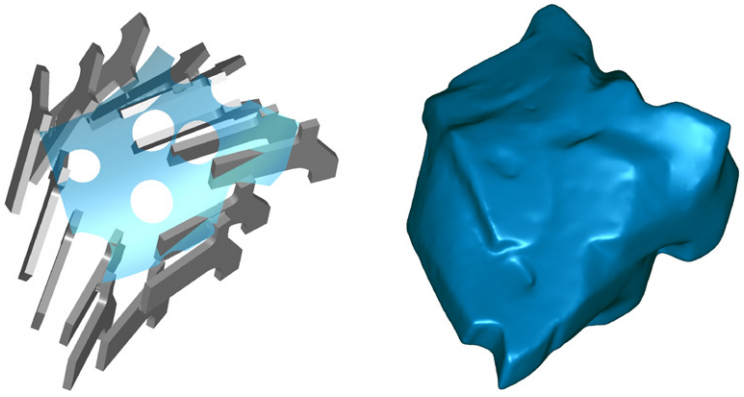

Fig. 2. Sweeping a solid over a surface computed as the projection of the configuration product of the two shapes. Left: the projection of the configuration product shown at discrete points on the surface boundary. Right: the projection of the configuration product corresponds to the sweep of the solid as it moves according to transformations defined by the surface.

Adopting the representation of configuration space as the Special Euclidean Group $\mathbb{SE}(3)$ [12] whose elements are 4×4 homogeneous transformation matrices, given a pair (A, B) of subsets of $\mathbb{SE}(3)$ the *configuration product* is a mapping $f : A \times B \rightarrow \mathbb{SE}(3)$ defined by

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

where $\cdot$ represents the group operation of matrix multiplication. Swept sets and Minkowski sums are then both sets of configurations $b \in B$ transformed by rigid transformations $a \in A$, 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 usually require computing the 6D space of configurations occupied by an object B as it is transformed according to the set A. Such problems reduce to a direct evaluation and representation of the configuration product $A \otimes B$, and subsume the classical problems of computing various instances of Minkowski sums and sweeps. For example, in manufacturing applications, it is often desirable to compute the sweep of a solid (tool) as it moves over a curve or surface while maintaining a particular orientation with respect to the curve/surface normals/tangents. An example is shown in Fig. 2. Another example of a 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 [13], but both are special cases of the configuration product.

Inverse problems typically impose the constraint $A \otimes B \subseteq C$, where C is a given subset of $\mathbb{SE}(3)$, and require computing the largest possible set of configurations A or the largest shape B that satisfies the constraint. In this sense, the inverse problems define A or B implicitly, and include common problems of packaging, where the set B must fit inside C under a set of transformations A, and motion planning where C plays the role of free (configuration) space. Formally, the inverse problems can be solved using operations dual to configuration product called *configuration quotients*, which are proper generalizations of the Minkowski difference and unsweep operations [11]. They are defined and studied in Section 2 of the paper.

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. The main contribution of this paper is to show that all these problems and many others involving general motions and relative configurations of solids may be effectively formulated using configuration products and quotients. We show that products and quotients may be rapidly approximated by sampling subsets of configuration space and computing all pairwise multiplications between the sampled sets. The inherently parallel computational procedure 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. We also derive sampling error estimates that can be used to develop effective sampling strategies and to measure deviations from exact computations.

1.2. Paper outline

Basic properties of configuration products and quotients are summarized in Section 2. The duality between configuration products and quotients subsumes the well known duality relationship

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

for Minkowski sum/difference [4], and for the `sweep/unsweep` operations [11]. The important observation is that quotients 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 some design and manufacturing process planning applications:

- 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.
- Computing the workspace of a mechanism.
- Computing the set of transformations of a weld gun that avoid collisions with surrounding tooling while maintaining contact with the work part.
- Defining constraints on the shape of a weld gun from constraints on its allowed transformations.
- Shape/motion design from workspace constraints.

In Section 4, we describe a simple parallel implementation to approximate the configuration products and quotients by sampling the primitive sets in $\mathbb{SE}(3)$ and computing pairwise multiplications of the sampled configurations. We also derive error bounds that can be used to control the quality of the approximation. Projections into $\mathbb{R}^3$ and $\mathbb{SO}(3)$ 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 Fig. 1. The concluding Section 5 follows the implementation and experimental results and discusses their significance for geometric modeling systems and applications.

1.3. Related work

The configuration space has often been exploited as the setting to formulate and solve problems involving relative position, contact, and interference of moving bodies, most notably for planning a collision free path of a robot in the presence of obstacles [6,7]. Typically, such problems require solving a direct problem by constructing a map called the C-space map that partitions configuration space into three disjoint components called the free space, contact space, and obstacle space [14–18]. Most computations of the C-space map involve computing just the contact space because it is a lower-dimensional hyper-surface that bounds both the free and obstacle spaces. In this section, we briefly discuss approaches for representing C-space maps, applications formulated as direct and inverse problems in configuration space, and the relationship between C-space maps, swept solids, and Minkowski operations. We do not attempt to make an exhaustive review of the research literature in these areas; more detailed reviews may be found in [3,14,19,8].

Direct problems in configuration space. The driving force for representing shapes in configuration space via the C-space map has been the robot motion planning problem, which focuses on developing boundary representations of obstacles in configuration space by analyzing contact conditions between the robot and obstacles [14]. The underlying idea of the configuration space formulation in robot motion planning is to map the obstacles into a configuration space so that planning the motion of the robot transforms into the problem of planning the motion of a point [7] around the six dimensional shape representing the obstacle space. The obstacle space is a semi-algebraic set bounded by the contact hyper-surface patches, where each patch represents a constraint on the robot's motion as it contacts an obstacle [16,17,14]. In practice the dimension and convexity (or the lack thereof) of the robot and obstacles significantly affect the difficulty of constructing the C-space map [14]. Approximations of the exact motion planning problem are often obtained by either projecting the obstacles into a lower-dimensional subspace of the configuration space thereby restricting the robot's degrees of freedom, or simplifying the shape and relying on more conservative motion planning. For example, when the robot and obstacles are polygons, the configuration space is three dimensional and the obstacle can be constructed as a sequence of slices in the configuration space where each slice represents all the positions achievable in a particular orientation [18,15]. In the special case when the robot is restricted to translate, the configuration space obstacle can be constructed as the Minkowski sum of the (reflected) robot and obstacles [7]. Following the advances in representing obstacle spaces for robot motion planning, other spatial planning problems have been formulated and solved using the configuration space formalism, but all such solutions are necessarily limited by the ability to compute and represent the required configuration spaces.

Inverse problems in configuration space. Configuration space formulations have been used to tackle inverse problems in shape design, given partial shape information and constraints on the relative motion of the contacting parts. For example, the design and modification of planar vibratory feeder boundaries based on motion constraints of the fed parts has been attempted in [18] using the configuration space formulation for planar robot motion planning. A similar technique is used to support the design of polygonal fixture/part pairs for form closure [20], i.e. part immobility under infinitesimal transformations. Form closure configurations are determined by intersection points of finitely many contact spaces derived from shapes of the fixtures and the part. Algorithms to compute and visualize C-space maps for such problems have largely been limited to three-dimensional configuration spaces of planar translations and rotations [15].

Generally speaking, the inverse problem of mapping the contact space to a pair of contacting moving shapes is under-constrained (many pairs of shapes can yield same contact space), but the problem may be properly constrained by fixing one of the two shapes. Then it is possible to construct a unique 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 [4,8] $C \ominus \hat{B}$ ($\hat{B}$ represents the reflection of B), which is defined in terms of the Minkowski difference. Additionally, it has been shown [11] 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 one-parameter sweep. This method of shape design is useful in designing mechanisms and solving packaging problems [21,22]. 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 [23,24] for planning work part accessibility by a coordinate measurement machine. In this paper, we remove all restrictions on the types of allowed motions (translation, rotation, single- or multi-parameter) and show that all inverse problems can be formulated using the notions of configuration quotients, leading to more general solutions for both shapes and transformations in configuration space.

Approximate computations. Early proposals to compute exact semi-algebraic boundaries of swept solids and configuration space obstacles [25–27] using the theory of envelopes and other algebraic methods [3,28] do not scale with the increase in dimension and complexity of realistic three-dimensional shapes [14,7]. The benefits of approximating swept solid boundaries by sampling and/or using polyhedral approximations [29,19,5,30,31] possibly in conjunction with graphics hardware [32,19] are being widely recognized and accepted. Similarly, methods that sample configuration

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

space are being adopted [33] as practical solutions to path planning problems. We will show that all direct and inverse problems may be solved via efficient sampling of primitives in configuration space and and computing pairwise products of transformations on the Graphics Processing Unit. This approach subsumes previous efforts to compute one-parameter sweeps and Minkowski sums in a sense that they are projections of the computed configuration product into $\mathbb{R}^3$. The brute force sampling approach may be further improved in some restricted situations via convolutions and the Fourier transform [34,35].

Group morphology. It is well known that Minkowski sums share the same group structure that forms the foundation of mathematical morphology [4]. The algebra for mathematical morphology was originally defined for subsets of $\mathbb{R}^n$, and later extended in [36] to subsets of any transformation group (rigid, affine, projective) acting on $\mathbb{R}^n$. It is easy to show that algebraic properties of one-parameter sweep and unsweep operations [11] is a special case the resulting group morphological structure. However, preserving the group structure through discretization, as proposed in [36] is problematic because the discrete subgroups of the general transformation groups (e.g. those involving translations and rotations) are limited in size or resolution [12,37]. Our approach to discretization, discussed in Section 4, relies on discretizing the primitive sets and not the underlying group, and therefore does not place restrictions on the sampling density.

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 $\mathbb{SE}(n) = \mathbb{R}^n \times \mathbb{SO}(n)$ [12]. Solids are represented as subsets of $\mathbb{R}^3$, and hence our focus is restricted to the group $\mathbb{SE}(3)$ 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, $\mathbb{SE}(3)$ is clearly a representation of configuration space [38].

Associating a position and a set of orientations to every point in a solid[1] facilitates a representation of the solid itself as a point set in $\mathbb{SE}(3)$ (more precisely as an element of $\mathbb{R}^3 \times \mathcal{P}(\mathbb{SO}(3))$, where $\mathcal{P}(\mathbb{SO}(3))$ is the power set of $\mathbb{SO}(3)$), such that the point set in Euclidean space is a projection of the point set in $\mathbb{SE}(3)$ onto its translational components via the map $\pi : \mathbb{SE}(3) \to \mathbb{R}^3$. This association may be conceptualized as attaching a set of coordinate frames at every point in the solid and is formally captured by an *embedding* $\gamma : \mathbb{R}^3 \to \mathbb{SE}(3)$ defined by

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

where $P = (x, y, z)$ is the position of a point in the solid and $\mathcal{R}$ is a set of rotation matrices. It follows that $\pi \circ \gamma = I$, where I represents the identity map.

The set of orientations $\mathcal{R}$ associated with a point of the solid can be the identity transformation I, a fixed orientation, a subset of the group of planar rotations $\mathbb{SO}(2)$ or a subset of the group of spatial rotations $\mathbb{SO}(3)$. The choice of embedding depends on how the solid is used in a particular application. The embedding may represent a solid's motion or sweep through a subset of configuration space $\mathbb{SE}(3)$. Such embeddings of moving shapes are encountered in many kinematic applications, for example when associating frames to robot links using the Denavit–Hartenberg convention [39], or when choosing a frame on a robot to compute its six dimensional workspace. In other applications, the solid itself may serve define a set of relative configurations for other solids. Below we consider two illustrative examples.

Example. Suppose A is an embedding of a solid with identity orientation assigned to all of its points. Then $A \otimes B$ is a translational sweep of set B over A. If B is also an embedding of a solid, then projection $\pi(A \otimes B)$ into Euclidean space is the usual Minkowski sum of the two solids. Notice, however, that $(A \otimes B)$ does not commute unless B is also embedded with the identify orientation. For example, the two shapes in Fig. 1 were embedded in $\mathbb{SE}(3)$ using identity orientations, and their Minkowski sum was computed via projection of their configuration product into $\mathbb{R}^3$. A similar approach may be used to define other solid modeling operations, including general sweeps, offset, and thickening operations.

Example. The orientation of a manufacturing tool (such as a paint or weld gun, or CMM probe) is always constrained with respect to the surface normals of the work part that the tool contacts. If the work part is embedded into $\mathbb{SE}(3)$ with boundary points oriented to align with corresponding surface normals, then the sweep of the tool over the surface of the work part is the projection $\pi(A \otimes B)$, where A is the embedding of the work part and B is embedding of the tool. For example, the sweep of a solid moving over the surface with normal contact in Fig. 2 was computed with $A(u, v)$ as the embedding of the work part surface $S(u, v)$ shown in Fig. 3. In this case, $A(u, v)$ was constructed from the surface $S(u, v)$ using orientations defined by the orthonormal vectors $[\frac{\partial S}{\partial u}, \frac{\partial S}{\partial v}, \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}]$. If the tool is allowed to change its orientation about the surface normal, each point of the work part surface would be embedded with $\mathbb{SO}(2)$ defined with respect to the surface normal, and so on.

Since both shapes and transformations can now be treated as subsets of a subgroup of $\mathbb{SE}(3)$, we are now in a position to define an algebra of sets in $\mathbb{SE}(3)$ that generalizes the sweep and Minkowski operations.

2.2. Products and quotients

We now highlight some algebraic properties of the configuration product and its dual configuration quotient operations. The algebra is conceptually similar to group morphology [36] and provides a natural generalization of properties of Minkowski algebra [4] and of one parameter sweeps [11], 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

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \tag{1}$$

$$(A \cup B) \otimes C = (A \otimes C) \cup (B \otimes C) \tag{2}$$

$$A \otimes (B \cup C) = (A \otimes B) \cup (A \otimes C). \tag{3}$$

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 the 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

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

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

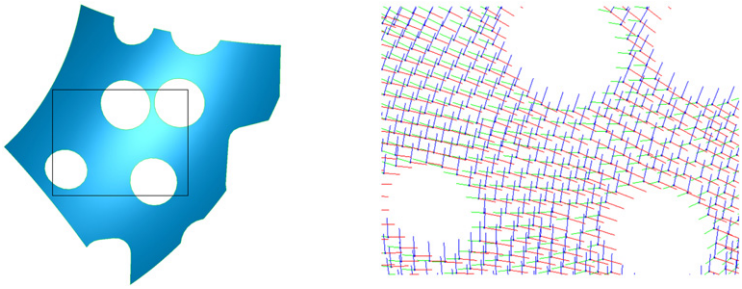

Fig. 3. The parametric surface $S(u, v)$ on the left is embedded in $\mathbb{SE}(3)$ by assigning each point of the surface an orientation defined by the orthonormal vectors $[\frac{\partial S}{\partial u}, \frac{\partial S}{\partial v}, \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}]$. Right: the resulting coordinate frames are shown at sampled positions in the region of the surface highlighted by the rectangle on the left.

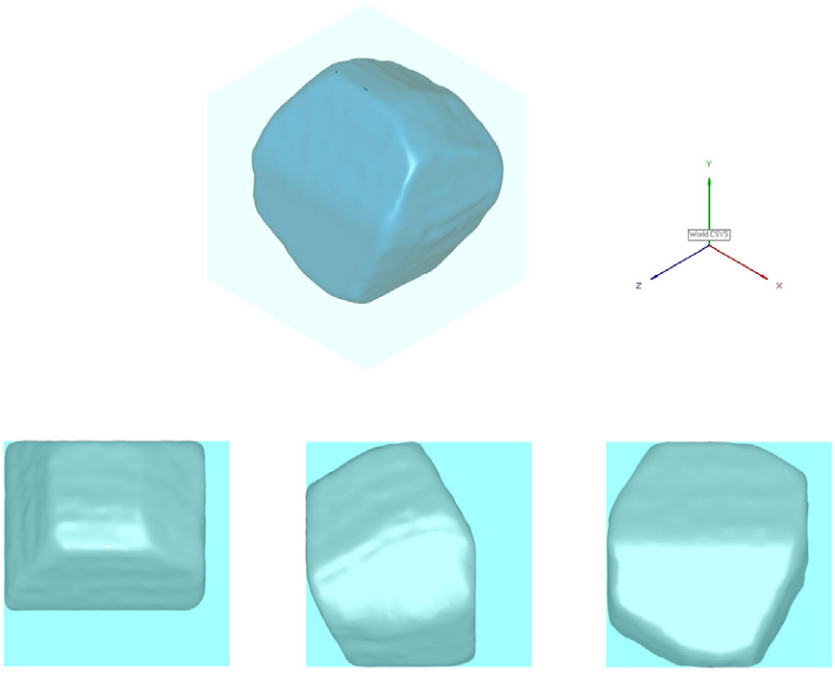

Fig. 4. Isometric view (top) and three orthographic projections (bottom) of the right configuration quotient $T \oslash C$, where C is a unit cube with side 2 centered at the origin, and T is a three parameter family $Rot_x(\phi) \otimes Rot_z(\theta) \otimes Trans(-x^2, 0, -x)$, $0 \le \theta, \phi \le -\frac{\pi}{6}$, $0 \le x \le 0.5$. The transformations $Rot_x(\phi)$ and $Rot_z(\theta)$ indicate a rotation of θ and ϕ about the x and z axes respectively, and the transformation $Trans(-x^2, 0, -x)$ indicates a translation by a vector $(-x^2, 0, -x)$.

unsweep, respectively, also generalize. This is indeed the case, but the lack of commutativity of matrix multiplication results in two dual operations, which we will call, respectively, *left* and *right quotients*.

The *left configuration quotient* of sets $A, B \in \mathbb{SE}(3)$ is a mapping $f : A \times B \to \mathbb{SE}(3)$ defined as

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

In the special case when $A, B \subset \mathbb{R}^3 \times I$, the left configuration quotient becomes the 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 [4]. Intuitively, the left configuration quotient corresponds to iteratively intersecting over the set B of configurations to which the set A of transformations is applied. We shall see in Section 2.3 that the left quotient is useful in determining the maximal subspace of transformations according to which a shape can move without leaving a given envelope.

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

$$A \osl B = \bigcap_{a \in A} a \otimes B. \tag{5}$$

Intuitively, the right configuration quotient corresponds to iteratively intersecting over the set A of all transformations that are being applied to a set B of configurations. Fig. 4 shows an example. When A is a path (a one parameter family of transformations) in configuration space and $B \subset \mathbb{R}^3 \times \mathbb{SO}(3)$, $\pi(A \osl B) = \texttt{unsweep}(B, \hat{A})$, 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 [11] for detailed analysis of the `unsweep` operation). It should not be surprising that the right quotient can be used to determine the maximal shape that fits within an envelope for a given set of configurations (see Section 2.3).

In general, $A \oslash B \neq A \osl B$, 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 duality property that generalizes the well known dualities of Minkowski operations and `sweep`/`unsweep`.

Proposition 2.1 (*Duality of Products and Quotients*). $(A \osl B^c)^c = A \otimes B = (A^c \oslash B)^c$.

Proof. Use de Morgan's laws to prove the statements.

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

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

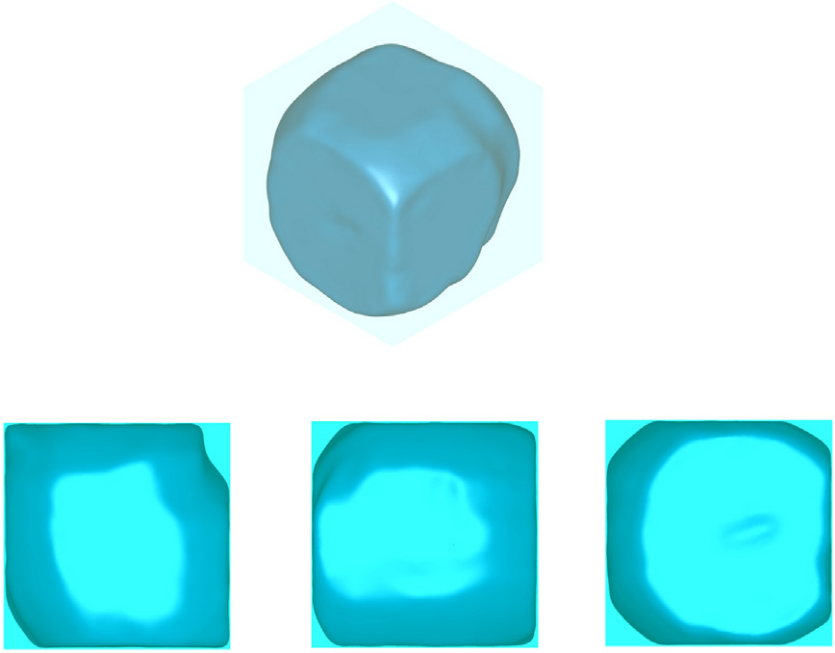

Fig. 5. The isometric view (top) and orthographic projections (bottom) of the product of the inverted family $\hat{T}$ of transformations with the right quotient computed in Fig. 4. By the containment property, if $\hat{T}$ is the set of transformations applied to a set X so that $\hat{T} \otimes X \subseteq C$ where C is the embedded cube, then the maximal set is $T \oslash C$ which is the right quotient computed in Fig. 4.

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

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

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

Proposition 2.1 is important because it holds for *arbitrary* sets of transformations in $\mathbb{SE}(3)$. 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 quotients 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.

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

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

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

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

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

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

2.3. 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 subgroups of dimensions m, n in $\mathbb{SE}(3)$, 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 $B \subset I \times \mathbb{SO}(2)$, the product $A \otimes B$ is clearly a subset of the four dimensional subgroup $\mathbb{R}^3 \times \mathbb{SO}(2)$. Since $\dim(A \times B) \leq \dim(A) + \dim(B)$ [40] and $\dim(A \otimes B) \leq 6$, the dimension of $A \otimes B$ is the lesser of $m + n$ and 6. If A and B are both embedded in the same subgroup of $\mathbb{SE}(3)$ 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 set of 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.

Proposition 2.2 (*Containment Property*). *Given sets $B, C \subset \mathbb{SE}(3)$, $A_1 = C \oslash \hat{B}$ is the largest subset of $\mathbb{SE}(3)$ such that $A_1 \otimes B \subseteq C$. Moreover, $A_2 = \hat{B} \oslash C$ is the largest subset of $\mathbb{SE}(3)$ such that $B \otimes A_2 \subseteq C$.*

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

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

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 $\mathbb{SE}(3)$ such that $B \otimes A_2 \subseteq C$. $\square$

Suppose A_2 is the set to be computed so that $B \otimes A_2 \subseteq C$ 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

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

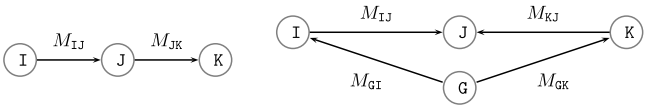

Fig. 6. Left: graph of a simple three node serial device that corresponds to a two link mechanism attached to the ground I. Right: graph of a simple four node parallel device corresponding to a found bar mechanism attached to the ground G.

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 [4]. More generally, when B and/or C are embedded in a 6D subset of $\mathbb{SE}(3)$, $A_1 = C \oslash \hat{B}$ represents the set of all transformations which, when applied to B, do not result in a collision with the complement of C. For example, Fig. 5 shows that the product $\hat{T} \otimes (C \oslash T)$, where T is the transformation used to compute the space in Fig. 4, is a subset of the cube C. The problem of finding a collision free path for a robot in the presence of obstacles [7] 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. One popular method of implementing this computation relies on the dual formulation in terms of $\otimes$, as described in Section 1.3.

3. Direct and inverse problems in design and manufacturing

3.1. Workspace of reconfigurable devices

Informally, the set of all configurations where a device can function is called its workspace. The concept of the workspace is used widely in robotics and manufacturing for planning effective device configurations for a variety of manufacturing applications, and techniques for either generating or representing workspaces have been proposed by a number of researchers [41–47,34]. Most techniques constructing workspaces are restricted to devices with either planar movement, specific joint types, or particular connectivity between components. The inverse problem of designing shapes of device components from their workspaces has not been widely studied. All workspace-related direct and inverse problems for arbitrary devices may be formulated and solved using configuration products and quotients.

Direct problems. A device's workspace is determined by the arrangement and transformations of its individual components. In modern commercial CAD systems this information is typically represented by an assembly graph, whose nodes store local coordinate frames on individual components and edges store relative transformations between components. Such a representation is well suited for generating individual positions and orientations in the device workspace via composition of transformations, as dictated by graph traversal. Devices with moving components may be classified on the basis of the connectivity between components as either *serial* or *parallel* devices [48]. Fig. 6(left) shows the graph representation of a simple serial device such as the two-link mechanism shown in Fig. 8, and Fig. 6(right) shows the graph of a parallel device such as the four-bar linkage in Fig. 7. A serial device's workspace is derived from basic forward kinematic expressions [41,43,45] and may be written as a composition of configuration products. For example, if q is a frame representing the local coordinate 'end-effector' on the device component K in Fig. 6(left), the positions and orientations occupied by q as the device operates is $M_{IJ} \otimes M_{JK} \otimes q$. Assuming both joints in the mechanism admit unconstrained planar rotational motion, the positions reachable by the mechanism will be an annulus defined by the projection of the configuration product $M_{IJ} \otimes M_{JK} \otimes q$.

Workspaces of parallel devices are usually studied on a case by case basis [49] depending on the available degrees of freedom, but can also be easily computed using configuration products. Every parallel device graph is a collection of serial device graphs joined at some common nodes. This implies that the workspace achievable at the component represented by a common node must be a subset of the intersection of the workspaces of all serial devices joined at the component. For example, the workspace of a frame q located on the coupler of a four bar chain is given by the expression $(M_{GI} \otimes M_{IJ} \otimes q) \cap (M_{GK} \otimes M_{KJ} \otimes q)$. Fig. 7 shows the projections of the 6D workspaces into $\mathbb{R}^3$. The two disconnected components of the workspace shown correspond to two distinct arrangements of the mechanism that are reflections of each other. The orientations achieved by the frame q may be obtained by projecting the 6D workspace into $\mathbb{SO}(3)$. Projections of the 6D workspace into $\mathbb{R}^3$ and $\mathbb{SO}(3)$ are termed the reachable and dextrous workspace, respectively [13]. These simple graph traversal rules may be used to compute the workspace of an arbitrary device constructed as a combination of serial and parallel devices.

Inverse problems. Components of reconfigurable devices can be redesigned to accommodate changes in the workspace, by systematically decomposing the workspaces based on the containment properties of left and right quotients. Fig. 8 illustrates the procedure for re-designing a two link serial mechanism to

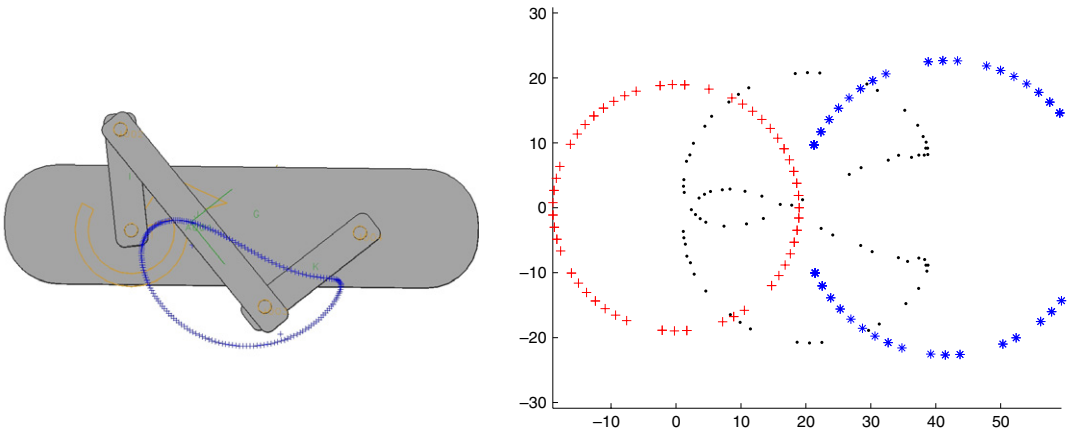

Fig. 7. Left: four bar chain with coupler reachable workspace, computed in NX 4, Right: Crank, Coupler and Follower reachable workspaces obtained by intersecting sampled workspaces of the serial chains.

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

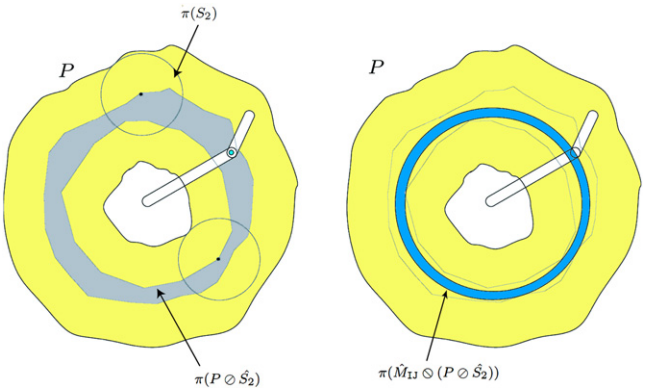

Fig. 8. Design modification in the 2-link device to cover the maximum subset of P. The grey space on the left is the set $\pi(P \oslash \hat{S}_2)$ of target positions for the first link; on the right, the set of all positions $\pi(\hat{M}_{IJ} \oslash (P \oslash \hat{S}_2))$ that can be achieved by the changes in the geometry of the first link.

reach the maximal set of positions contained within the desired workspace P (which could be modified based on presence of obstacles, or desired accessibility constraints). Recall that the reachable workspace of a *given* two-link mechanism with the graph shown in Fig. 6(left) is an annulus computed by a composition of configuration products that follows the graph. Suppose we would like to increase the workspace of the device to cover as much of the set P as possible, by redesigning the first link of the mechanism. The workspace P is embedded as the set $\gamma(P)$ into the subgroup $\mathbb{R}^2 \times \mathbb{SO}(2)$ with all orientations in $\mathbb{SO}(2)$ associated to each position. Labeling the workspace of the second link as $S_2 = M_{JK} \otimes q$, the reference frame on the first link can only be located in the space defined by the set $\gamma(P) \oslash \hat{S}_2$. Furthermore, since the first link being modified is restricted to rotate about the pivot attached to the ground, the maximal subset of $\gamma(P) \oslash \hat{S}_2$ where the frame on the first link can be located is defined by $X = \hat{M}_{IJ} \oslash (\gamma(P) \oslash \hat{S}_2)$, and its translational component constitutes an annulus as shown in Fig. 8. We conclude that X is a set of configurations that represents the increased workspace of the first link, which requires changes in the link's geometry and/or its motion. If we assume that the link's motion M_{IJ} is determined by a revolute joint and cannot be modified, we have no choice but to modify that geometry of the link that determines the set F of reference frames for the motion of the second link, so that $M_{IJ} \otimes F = X$. The intersection of $X \oslash M_{IJ}$ with the link's shape yields the maximal subset of configurations that can be included in F, for example, by cutting a slot according to dimensions defined by the annulus and the width of the first link.

3.2. Weld process planning

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 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 gun has only one degree of freedom

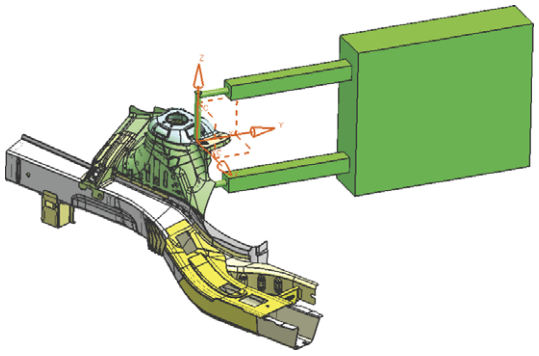

Fig. 9. A typical work assembly and weld gun.

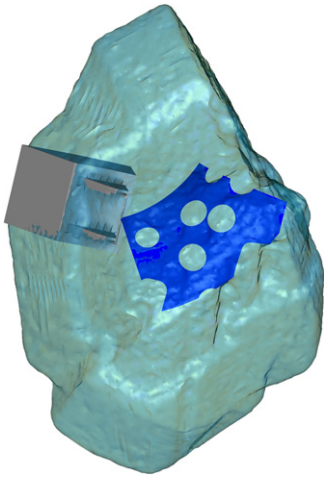

Fig. 10. The weld gun in Fig. 9 swept by the clearance range of orientations, and subsequently swept over the part surface whose embedding is described in Fig. 3.

corresponding to rotation about the normal to maintain valid contact. A typical welding setup is shown in Fig. 9.

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 $\mathbb{SO}(2)$ about the 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 Fig. 10. Suppose now that T is the set of transformations in $\mathbb{R}^3 \times \mathbb{SO}(3)$ 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 $\mathbb{SE}(3)$ 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

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

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 the surface, with orientation controlled by the surface normal. Fig. 10 shows 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 the figure.

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 $W \subset \mathbb{R}^3 \times \mathbb{SO}(2)$ by associating all rotations in $\mathbb{SO}(2)$ defined by the surface normal at the weld location to every position. 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 $\mathbb{R}^3 \times \mathbb{SO}(2)$. 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 $\mathbb{R}^3 \times \mathbb{SO}(2)$. 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 $\mathbb{SO}(2)$. 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 work part. The work part must now be embedded in all of $\mathbb{R}^3 \times \mathbb{SO}(3)$, 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 $\mathbb{SE}(3)$ 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 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 [50].

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 $\mathbb{R}^3 \times \mathbb{SO}(2)$ by associating all of $\mathbb{SO}(2)$ 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 $\pi(X) = \texttt{unsweep}(\pi((iW)^c), M)$. When X is unbounded, it is more practical to compute $\pi(X)$ as using the dual `sweep` operation. Fig. 11 shows an approximation of $\texttt{sweep}(W, \hat{M})$ as a sweep of work part configurations.

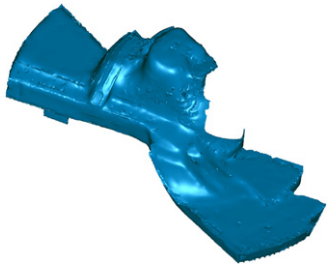

Fig. 11. The complement of the shown $\texttt{sweep}(\pi(W), \hat{M})$ is the $\texttt{unsweep}(\pi(W^c), M)$ which defines the maximal shape for any tool that moves according to M without interference. Here $\pi(W)$ is the assembly with the work part shown in Fig. 9 and M is a 30 degree rotation about the part surface normal at the weld location.

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 $\mathbb{R}^3 \times \mathbb{SO}(3)$, as described above.

4. GPU implementation via sampling

Computing the configuration product exactly for general subsets $A, B \subset \mathbb{SE}(3)$ is clearly quite challenging. Representing boundaries of configuration products as algebraic or semi-algebraic sets is difficult even for non-convex polyhedra, as evident from efforts in computing configuration space obstacles. 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. For the well understood special case of one-parameter sweeps and Minkowski sums of polyhedra, exact solutions are rare [5,30], and approximate sampling techniques have been widely advocated [35,19,31,29].

We adopt the sampling solution for computing configuration products, in several stages that are similar to those described in [31] 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. Fig. 13 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 Figs. 1, 2 and 13.

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

4.1. Sampling strategies

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 $\mathbb{SE}(3)$ is accomplished by sampling the solids in $\mathbb{R}^3$ and embedding the sampled set in $\mathbb{SE}(3)$ 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. To make the notion of a dense sampling of configurations more precise, it is necessary to have a distance function in $\mathbb{SE}(3)$.

Since $\mathbb{SE}(3)$ is a Riemannian manifold [38], any two configurations $x, y \in \mathbb{SE}(3)$ can be joined by a length minimizing geodesic such that the length of the geodesic is the Riemannian distance function [51] $d(x, y) = \|\ln(x^{-1}y)\|$ where ln represents the matrix logarithm [52] 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 $\mathbb{SE}(3)$ 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 $\mathbb{SE}(3)$, while observing that other metrics are possible [38] and are useful for different applications.

Once the metric is selected, the notion of a sufficiently dense sampling may be formalized. An *ϵ-neighborhood* $B_\epsilon(a)$ located at $a \in A$, for any $A \subset \mathbb{SE}(3)$, is defined as intersection of an open ball with the set A:

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

Representing the sampling of a set A as S_A, following [31] 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

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

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

A good sampling strategy should assure that the product $S_A \otimes S_B$ of ϵ-coverings S_A, S_B, is an ϵ-covering of $A \otimes B$, which is true if and only if

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

In turn, Eq. (24) will be satisfied if and only if the following relations are satisfied

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

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

Eq. (25) always holds true by construction. However, we show below that Eq. (26) is not always satisfied because it may be 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. See an example in Fig. 12.

We will now estimate the bound on the maximum deviation from an ϵ-covering induced on approximating $A \otimes B$ by the product

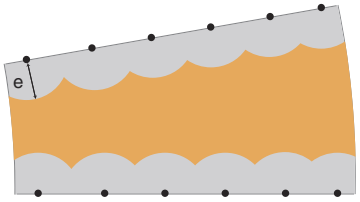

Fig. 12. A is a one parameter family of planar rotations approximated by an ϵ-covering S_A consisting of two rotation matrices. B is a line segment along the x-axis far from the origin approximated by an ϵ-covering S_B of its embedding in $\mathbb{SE}(3)$. The projection $\pi(A \otimes B)$ is the one parameter sweep consisting of the union of all shaded (gray and orange) regions, and the projection $\pi(S_A \otimes S_B)$ 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. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of ϵ-coverings S_A, S_B. It is sufficient to establish this bound for the case when both $S_A = \{a\}$ and $S_B = \{b\}$ are single point ϵ-coverings. In such a case $A = B_\epsilon(a)$, $B = B_\epsilon(b)$ and $S_A \otimes S_B = ab$ is the approximation of $A \otimes B$. Consider points $a_* \in B_\epsilon(a)$ and $b_* \in B_\epsilon(b)$ such that $d(a, a_*) < \epsilon$ and $d(b, b_*) < \epsilon$. We will characterize the distances of all points in $\{a, a_*\} \otimes \{b, b_*\}$ from ab and thereby determine the maximum deviation of points in the exact product $B_\epsilon(a) \otimes B_\epsilon(b)$ from the approximate product. Excluding the point ab whose distance to itself is trivially zero, the three other points obtained in the product $\{a, a_*\} \otimes \{b, b_*\}$ are $\{ab_*, a_*b, a_*b_*\}$. The following propositions establish tight bounds on these three distances in general and several important special situations.

Proposition 4.1. $d(ab, ab_*) < \epsilon$.

Proof. The distance of the point ab_* from ab is ϵ since $d(ab, ab_*) = \|\ln(b^{-1}a^{-1}ab_*)\| = \|\ln(b^{-1}b_*)\| = d(b, b_*) < \epsilon$. □

Proposition 4.2. $d(ab, a_*b) \leq \epsilon\|b\|^2$.

Proof. The distance of the point a_*b from the transformed center ab is given by

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

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

In order to simplify Eq. (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 [52,53] (see Appendix B for a proof)

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

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 $\epsilon\|b\|^2$. □

Proposition 4.2 applies to a common situation where a_* includes a rotation, as illustrated in Fig. 12. However, there are two

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

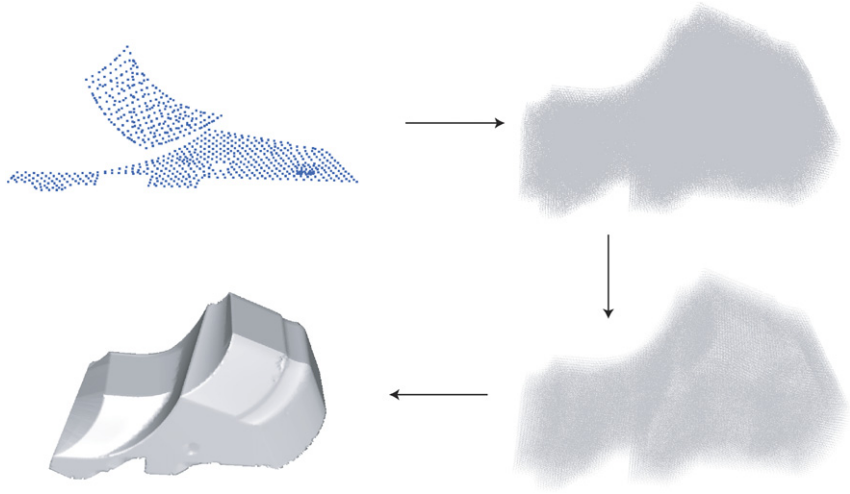

Fig. 13. Approximate Minkowski sum computed as a configuration product of samplings of surfaces. Top right: projection of the point set obtained after computing the product on the GPU, computation time = 0.07186 s. Bottom right: the remaining points after filtering out interior points using flood fill, 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.

special cases where the product $d(ab, a_*b)$ will always be ϵ, depending on the subgroups of $\mathbb{SE}(3)$ into which both primitive sets are embedded. These two cases are summarized in the following two propositions.

Proposition 4.3. *When S_A and S_B are single point ϵ-coverings of subsets of Abelian subgroups i.e. translations $\mathbb{R}^3$ or planar rotations $\mathbb{SO}(2)$, $d(ab, a_*b) < \epsilon$.*

Proof. Owing to the commutative relation $xy = yx$ it follows that $y^{-1}xy = x$. Therefore Eq. (28) immediately reduces to

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

Thus in this special case, $B_\epsilon(a) \otimes b$ will always be an ϵ-ball centered around ab. Similarly $B_\epsilon(b) \otimes a$ will be an ϵ-ball centered around ab. □

Proposition 4.4. *When S_A is a single point ϵ-covering of a set in $\mathbb{SE}(3)$ and S_B is a single point ϵ-covering of a set in $\mathbb{SO}(3)$, $d(ab, a_*b) < \epsilon$.*

Proof. The proof of this theorem is given in Appendix A of this paper. □

Proposition 4.5. $d(ab, a_*b_*) \leq \epsilon(1 + \|b\|^2)$.

Proof. Using the triangle inequality induced by the metric, we assert that $d(ab, a_*b_*) \leq d(ab, a_*b) + d(a_*b, a_*b_*)$. Analogous to computing $d(ab, ab_*)$ in Proposition 4.1, we can see that $d(a_*b, a_*b_*) < \epsilon$. Therefore in general $d(ab, a_*b_*) \leq \epsilon(1 + \|b\|^2)$ if both S_A and S_B are epsilon coverings. For the special cases when $d(ab, a_*b) < \epsilon$, we have $d(ab, a_*b_*) < 2\epsilon$. □

The computed error bounds may be used to devise effective sampling strategies to guarantee that the product of two ϵ-coverings is an ϵ-covering of the exact configuration product. For example, the bound in Proposition 4.3 underlies the observation that an epsilon-covering of the boundary of Minkowski sum $\partial(A \oplus B)$ can be computed as Minkowski sum of the epsilon-coverings of the individual boundaries ∂A and ∂B, as is shown in [31] for polyhedral models. Similarly, Proposition 4.4 guarantees that applying a solid sampled as an ϵ-covering in $\mathbb{R}^3 \times I$ to a set of non-trivial orientations sampled as an ϵ-covering in $I \times \mathbb{SO}(3)$ always results in an ϵ-covering of the configuration solid in $\mathbb{R}^3 \times \mathbb{SO}(3)$. This property may be useful in sampling embedded solids as ϵ-coverings, where an identical copy of a subset of $\mathbb{SO}(3)$ is associated to every point in the solid. More sophisticated sampling strategies may be required in general situations. For example, it may be prudent to choose a very fine sampling density such that $\epsilon \to 0$, and scale the transformations closer to the origin such that $\epsilon\|b\|^2 \approx \epsilon$, in order to compensate for the increasing deviation from the ϵ-covering for transformations further from the origin as illustrated in Fig. 12.

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 [54] 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. 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 Fig. 13, 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. Our current implementation is simple but 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

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

similar to algorithms described in [35,34] which has complexity $O(n \log n)$ as opposed to $O(n^2)$ for the pairwise multiplication implementation, and can be implemented in parallel.

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 $\mathbb{SO}(3)$. 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 [55] 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 [31]. 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 Geomagic software to generate bounding surfaces in Figs. 1, 2, 4, 5, 10 and 11, and the robust cocone algorithm developed by Dey and Goswami [56] in Fig. 13. All CPU computations were performed on an Intel Core 2 Duo workstation (3 GHz, 3 GHz, 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 [57].

5. Conclusions

The main contribution of this paper is to show that configuration products and quotients generalize and unify a variety of computations in geometric modeling involving motions and relative configurations. By treating solids and rigid transformations as point sets in the configuration space $\mathbb{SE}(3)$, we have shown that formulations of most 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, determination of feasible (configuration) spaces for shapes under general motions, and workspace computations.

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 [35,34]. We note that many steps in the sampling reconstruction process may be improved further; for example, a marching cubes algorithm to compute a triangulation of a surface is now available with the NVIDIA CUDA SDK and could be adapted to compute surfaces from noisy point data.

The 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.

Acknowledgements

We would like to thank our anonymous reviewers for their comments and suggestions. We would also like to thank Tamal Dey for permitting the use of the robust cocone software. This research was supported in part by the National Science Foundation grants CMMI-0500380 and CMMI-0621116.

Appendix A. Proof of Proposition 4.4

Proposition. *For $A \subset SE(3)$, $B \subset SO(3)$, given $(a, a_*) \in A$ such that $d(a, a_*) < \epsilon$ and $b \in B$, it follows that $d(ab, a_*b) < \epsilon$.*

Proof. We know $d(ab, a_*b) < \|b^{-1} \ln(a^{-1}a_*)b\|$. Here $b \in SO(3)$ and $\ln(a^{-1}a_*) \in \mathfrak{se}(3)$ where $\mathfrak{se}(3)$ is the Lie algebra of $SE(3)$. Consider a typical element in $\mathfrak{se}(3)$ written as

$$\begin{pmatrix} \Omega & \mathbf{v} \\ 0 & 0 \end{pmatrix}$$

where Ω is a skew symmetric matrix of the form

$$\Omega = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

corresponding to an element of the Lie algebra of the rotation group,[2] and $\mathbf{v}$ is a translation. From Eq. (31), we know $d(ab, a_*b) = \|b^{-1} \ln(a^{-1}a_*)b\|$ where $\mathsf{X} = \ln(a^{-1}a_*)$ is an element of the Lie algebra $\mathfrak{se}(3)$ of $\mathbb{SE}(3)$. The expression $b^{-1}\mathsf{X}b$ is called the *action* of b on X [12]. The action of an element of $SO(3)$ on the Lie algebra element X may be computed (without losing generality) as follows

$$\begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \Omega & \mathbf{v} \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} R^t & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R\Omega R^t & R\mathbf{v} \\ 0 & 0 \end{pmatrix}. \tag{A.1}$$

To show that $d(ab, a_*b) < \epsilon$ we need to show that the Frobenius norm of the matrix on the right side of Eq. (A.1) is the same as that of X. First we will show $\|R\Omega R^t\| = \|\Omega\|$. For any vector $\mathbf{x} \in \mathbb{R}^3$, we have $\Omega\mathbf{x} = \omega \times \mathbf{x}$ where $\omega = (\omega_x, \omega_y, \omega_z)^T$, and $\times$ represents cross product. It follows that $\|\Omega\| = \|\omega\|\sqrt{2}$. Any rotation matrix R can be partitioned into three mutually orthogonal vectors defined by the rows of the rotation matrix (or its columns since the inverse of a rotation matrix is its transpose). Thus we can write

$$R = \begin{pmatrix} \mathbf{r}_i^t \\ \mathbf{r}_j^t \\ \mathbf{r}_k^t \end{pmatrix}$$

with the induced cyclic relations $\mathbf{r}_i \times \mathbf{r}_j = \mathbf{r}_k$, $\mathbf{r}_j \times \mathbf{r}_k = \mathbf{r}_i$, $\mathbf{r}_k \times \mathbf{r}_i = \mathbf{r}_j$. Then we have $R^t = (\mathbf{r}_i\ \mathbf{r}_j\ \mathbf{r}_k)$ and

$$\begin{aligned} R\Omega R^t &= R(\omega \times \mathbf{r}_i\ \ \omega \times \mathbf{r}_j\ \ \omega \times \mathbf{r}_k) \\ &= \begin{pmatrix} 0 & \mathbf{r}_i \cdot (\omega \times \mathbf{r}_j) & \mathbf{r}_i \cdot (\omega \times \mathbf{r}_k) \\ \mathbf{r}_j \cdot (\omega \times \mathbf{r}_i) & 0 & \mathbf{r}_j \cdot (\omega \times \mathbf{r}_k) \\ \mathbf{r}_k \cdot (\omega \times \mathbf{r}_i) & \mathbf{r}_k \cdot (\omega \times \mathbf{r}_j) & 0 \end{pmatrix}. \end{aligned}$$

The scalar triple product $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$ may be equivalently rearranged cyclically as $\mathbf{b} \cdot (\mathbf{c} \times \mathbf{a})$. Therefore we have

[2] The logarithm of a rotation matrix gives an element of its Lie algebra, which is a skew symmetric matrix.

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

$$\begin{pmatrix} 0 & \mathbf{r}_i \cdot (\omega \times \mathbf{r}_j) & \mathbf{r}_i \cdot (\omega \times \mathbf{r}_k) \\ \mathbf{r}_j \cdot (\omega \times \mathbf{r}_i) & 0 & \mathbf{r}_j \cdot (\omega \times \mathbf{r}_k) \\ \mathbf{r}_k \cdot (\omega \times \mathbf{r}_i) & \mathbf{r}_k \cdot (\omega \times \mathbf{r}_j) & 0 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & \omega \cdot (\mathbf{r}_j \times \mathbf{r}_i) & \omega \cdot (\mathbf{r}_k \times \mathbf{r}_i) \\ \omega \cdot (\mathbf{r}_i \times \mathbf{r}_j) & 0 & \omega \cdot (\mathbf{r}_k \times \mathbf{r}_j) \\ \omega \cdot (\mathbf{r}_i \times \mathbf{r}_k) & \omega \cdot (\mathbf{r}_j \times \mathbf{r}_k) & 0 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & -\mathbf{r}_k^t \cdot \omega & \mathbf{r}_j^t \cdot \omega \\ \mathbf{r}_k^t \cdot \omega & 0 & -\mathbf{r}_i^t \cdot \omega \\ -\mathbf{r}_j^t \cdot \omega & \mathbf{r}_i^t \cdot \omega & 0 \end{pmatrix}.$$

Observing that $R \cdot \omega = (\mathbf{r}_i^t \cdot \omega, \mathbf{r}_j^t \cdot \omega, \mathbf{r}_k^t \cdot \omega)^t$, it follows that $\|R\Omega R^t\| = \|R \cdot \omega\|\sqrt{2}$. Since R is an orthogonal transformation, $\|R \cdot \omega\| = \|\omega\|$ and therefore $\|R\Omega R^t\| = \|\omega\|\sqrt{2} = \|\Omega\|$. Similarly $\|R\mathbf{v}\| = \|\mathbf{v}\|$. Therefore the norm of the action is the same as the norm for the $\mathfrak{se}(3)$ element X. □

Appendix B. Properties of matrix exponential and logarithm

This section summarizes standard definitions of the matrix exponential and logarithm [52] and gives the common derivation of Eq. (29). The exponential e^M of a matrix M is defined as

$$e^M = I + M + \frac{M^2}{2!} + \frac{M^3}{3!} + \cdots. \tag{B.1}$$

The logarithm $\ln(X)$ of a matrix X is defined as

$$\ln(X) = (X - I) - \frac{(X-I)^2}{2} + \frac{(X-I)^3}{3} - \frac{(X-I)^4}{4} + \cdots. \tag{B.2}$$

Unfortunately if X is not near the identity the matrix logarithm either does not converge, or converges so slowly that it is not of practical use [52,58]. The standard way of dealing with this problem is to use the square root operator repeatedly to bring X near the identity [58]

$$\ln(X) = 2^k \ln(X^{1/2^k}). \tag{B.3}$$

As k increases $X^{1/2^k} \to I$, so for sufficiently large k standard methods to compute the logarithm (such as truncating the series in Eq. (B.2)) may be applied. This procedure reverses the more common scaling and squaring method of evaluating the matrix exponential $e^M = e^{(\frac{M}{2^k})^{2^k}}$ [58]. These computational methods ensure $e^{\ln(X)} = X$ and $\ln(e^M) = M$ up to specified accuracy [59].

Now we show that $e^{B^{-1}AB} = B^{-1}e^A B$ which is a restatement of Eq. (29). We use the proof derived in [52]. First we check that

$$(B^{-1}AB)^n = (B^{-1}AB)(B^{-1}AB)\ldots(B^{-1}AB) \tag{B.4}$$
$$= B^{-1}A^nB. \tag{B.5}$$

Using the above relation, and the fact that $B^{-1}(C+D)B = B^{-1}CB + B^{-1}DB$, we get

$$e^{B^{-1}AB} = \sum_{n=0}^{\infty} \frac{(B^{-1}AB)^n}{n!} \tag{B.6}$$
$$= B^{-1}\left(\sum_{n=0}^{\infty} \frac{A^n}{n}\right)B \tag{B.7}$$
$$= B^{-1}e^A B. \tag{B.8}$$

References

[1] Requicha AAG. Representations for rigid solids: theory, methods, and systems. ACM Computing Surveys 1980;12(4):437–64.
[2] Shapiro V. Solid modeling. In: Handbook of computer aided geometric design. 2001.
[3] Malek K, Yang J, Blackmore D, Joy K. Swept volumes: foundations, perspectives, and applications. International Journal of Shape Modeling 2004;2(1):87–127.
[4] Serra J. Image analysis and mathematical morphology. Orlando (FL, USA): Academic Press, Inc.; 1983.
[5] Kaul A, Rossignac J. Solid-interpolating deformations: construction and animation of PIPs. Computers & Graphics 1992;16(1):107–15.
[6] Lozano-Perez T. Spatial planning: a configuration space approach. IEEE Transactions on Computers 1983;100(32):108–20.
[7] Latombe JC. Robot motion planning. Kluwer Academic Publishers; 1991.
[8] Middleditch AE. Application of vector sum operator. Computer-Aided Design 1988;20(4):183–8.
[9] Boissonnat JD, De Lange E, Teillaud M. Minkowski operations for satellite antenna layout. In: Proceedings of the thirteenth annual symposium on computational geometry. New York (NY, USA): ACM; 1997. p. 7–76.
[10] Ghosh Pijush, Jain PK. An algebra of geometric shapes. IEEE Computer Graphics and Applications 1993;13(5):50–9.
[11] Ilies HT, Shapiro V. Dual of sweep. Computer Aided Design 1999;31(3):185–201.
[12] Selig JM. Geometric fundamentals of robotics. Springer Verlag; 2005.
[13] Agrawal SK. Workspace boundaries of in-parallel manipulator systems. In: Advanced robotics, 1991. 'Robots in unstructured environments', 91 ICAR., Fifth International Conference on. 1991. p. 1147–52.
[14] Wise KD, Bowyer A. A survey of global configuration-space mapping techniques for a single robot in a static environment. The International Journal of Robotics Research 2000;19(8):762.
[15] Sacks E, Bajaj C. Sliced configuration spaces for curved planar bodies. The International Journal of Robotics Research 1998;17(6):639.
[16] Donald BR. A search algorithm for motion planning with six degrees of freedom. Artificial Intelligence 1987;31(3):295–353.
[17] Canny J. The complexity of robot motion planning ACM doctoral dissertation awards. MIT Press; 1988.
[18] Caine M. The design of shape interactions using motion constraints. In: 1994 IEEE international conference on robotics and automation. 1994. Proceedings. 1994. p. 366–71.
[19] Varadhan G, Manocha D. Accurate Minkowski sum approximation of polyhedral models. Graphical Models 2006;68(4):343–55.
[20] Brost RC, Goldberg KY. A complete algorithm for designing planar fixtures using modular components. Robotics and Automation, IEEE Transactions on 1996;12(1):31–46.
[21] Ilieş HT, Shapiro V. On shaping with motion. Journal of Mechanical Design 2000;122:567.
[22] Ilies H, Shapiro V. A class of forms from function: the case of parts moving in contact. Research in Engineering Design 2001;13(3):157–66.
[23] Spyridi AJ, Requicha AAG. Accessibility analysis for the automatic inspection of mechanicalparts by coordinate measuring machines. In: 1990 IEEE international conference on robotics and automation, 1990. Proceedings. 1990. p. 1284–9.
[24] Spitz SN, Spyridi AJ, Requicha AAG. Accessibility analysis for planning of dimensional inspection with coordinate measuring machines. IEEE Transactions on Robotics and Automation 1999;15(4):714–27.
[25] Martin RR, Stephenson PC. Sweeping of three-dimensional objects. Computer-Aided Design 1990;22(4):223–34.
[26] Leu MC, Lu F, Blackmore D. Simulation of NC machining with cutter deflection by modelling deformed swept volumes. CIRP Annals-Manufacturing Technology 1998;47(1):441–6.
[27] Bajaj CL, Kim MS. Generation of configuration space obstacles: the case of moving algebraic curves. Algorithmica 1989;4(1):157–72.
[28] Lee IK, Kim MS, Elber G. Polynomial/rational approximation of Minkowski sum boundary curves. Graphical Models and Image Processing 1998;60(2):136–65.
[29] Kim YJ, Varadhan G, Lin MC, Manocha D. Fast swept volume approximation of complex polyhedral models. Computer-Aided Design 2004;36(11):1013–27.
[30] Hachenberger P. Exact Minkowksi sums of polyhedra and exact and efficient decomposition of polyhedra in convex pieces. Lecture Notes in Computer Science 2007;4698:669.
[31] Lien JM. Covering Minkowski sum boundary using points with applications. Computer Aided Geometric Design 2008;25(8):652–66.
[32] Schroeder WJ, Lorensen WE, Linthicum S. Implicit modeling of swept surfaces and volumes. In: Proceedings of the conference on visualization'94. IEEE Computer Society Press; 1994. p. 40–5.
[33] Hsu D, Latombe JC, Motwani R. Path planning in expansive configuration spaces. International Journal of Computational Geometry and Applications 1999;9(4):495–512.
[34] Chirikjian GS, Kyatkin AB. Engineering applications of noncommutative harmonic analysis. 2001.
[35] Kavraki LE. Computation of configuration-space obstacles using the fast Fourier transform. IEEE Transactions on Robotics and Automation 1995;11(3):408–13.
[36] Roerdink JBTM. Group morphology. Pattern Recognition 2000;33(6):877–95.
[37] Miller W. Symmetry groups and their applications. Academic Press; 1972.
[38] Zefran M, Kumar V, Croke C. Metrics and connections for rigid-body kinematics. International Journal of Robotics Research 1999;18(2):243–58.
[39] Denavit J, Hartenberg RS. A kinematic notation for lower-pair mechanisms based on matrices. Journal of Applied Mechanics 1955;22(2):215–21.
[40] Wage ML. The dimension of product spaces. Proceedings of National Academy of Sciences 1978;75(10):4671–2.
[41] Tsai YC, Soni AH. An algorithm for the workspace of a general $n - r$ robot. Transactions of the ASME 1983;105(March):52–7.

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007

[42] Merlet JP. Determination of 6d workspaces of Gough-type parallel manipulator and comparison between different geometries. International Journal of Robotics Research 1999;18(9):902–16.
[43] Davidson JK, Hunt KH. Robot workspace of a tool plane: part 1 — a ruled surface and other geometry. Transactions of the ASME 1987;109(March):50–9.
[44] Yang DCH, Lee TW. On the workspace of mechanical manipulators. Transactions of the ASME 1983;105(March):62–9.
[45] Botturi D, Fiorini P, Martelli S. A geometrical method for robot workspace computation. In: International conference on advanced robotics. 2003.
[46] Vijaykumar R, Tsai M, Waldron K. Geometric optimization of manipulator structures for working volume and dexterity. In: Proceedings of the 1985 IEEE international conference on robotics and automation. 1985.
[47] Chen I-Ming, Yang Guilin. Inverse kinematics for modular reconfigurable robots. In: Proceedings of the 1998 IEEE international conference on robotics and automation. 1998. p. 1647–52.
[48] McCarthy JM, Joskowicz L. Kinematic synthesis. Formal Engineering Design Synthesis 2001;321.
[49] Merlet JP. Parallel robots. Kluwer Academic Publishers; 2006.
[50] Nelaturi S, Abhyankar A, Shapiro V, Tilove R. Feasible spaces in weld gun selection. In: IEEE international conference on automation science and engineering, 2008. CASE 2008. 2008. p. 870–5.
[51] Burns KH, Gidea M. Differential geometry and topology: with a view to dynamical systems. Chapman & Hall/CRC; 2005.
[52] Curtis ML. Matrix groups. Springer; 1984.
[53] Gurarie D. Symmetries and Laplacians: introduction to harmonic analysis, group representations, and applications. North Holland; 1992.
[54] NVIDIA. Compute unified device architecture programming guide. 2007.
[55] Foley JD, Van Dam A. Fundamentals of interactive computer graphics. Boston (MA, USA): Addison-Wesley Longman Publishing Co., Inc.; 1982.
[56] Dey TK, Goswami S. Provable surface reconstruction from noisy samples. In: Proceedings of the twentieth annual symposium on computational geometry. New York (NY, USA): ACM; 2004. p. 330–9.
[57] Alpern B, Carter L, Grayson M, Pelkie C. Orientation maps: techniques for visualizing rotations (a consumer's guide). In: Proceedings of the 4th conference on visualization'93. 1993. p. 183–8.
[58] Cheng SH, Higham NJ, Kenney CS, Laub AJ. Approximating the logarithm of a matrix to specified accuracy. SIAM Journal on Matrix Analysis and Applications 2001;22:1112.
[59] Kenney CS, Laub AJ. A Schur–Frechet algorithm for computing the logarithm and exponential of a matrix. SIAM Journal on Matrix Analysis and Applications 1998;19(3):640–63.

Please cite this article in press as: Nelaturi S, Shapiro V. Configuration products and quotients in geometric modeling. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.09.007