

# Corner-Operated Tran-Similar (COTS) maps, patterns, and lattices

JAREK ROSSIGNAC, Georgia Institute of Technology

The planar **COTS map** proposed here takes the unit square to a region  $R$  bounded by four log-spiral edges. It is **Corner-Operated** (controlled by the four corners of  $R$ ) and **Tran-Similar** (it maps translations to similarities). The tiles of the COTS map of a regular pattern are similar to each other. It may facilitate intuitive design and algorithmic optimization of procedural models of complex, possibly multi-resolution, lattices, because it affords constant-cost algorithms for Point-In-Lattice testing and for Total-Area-Calculations. We provide simple, closed-form expressions for evaluating the COTS map and its inverse from the positions of its corners. We conjecture that the COTS map may be useful in a variety of applications in Engineering, Architecture, and Art and we provide a few illustrative examples of its possibilities. We compare it to related, previously proposed, planar maps and discuss several variations and extensions.

CCS Concepts: • Applied computing → Computer-aided design; • Computing methodologies → Parametric curve and surface models; Texturing; Procedural animation; Representation of mathematical functions; Image processing;

Additional Key Words and Phrases: Planar Map; Deformation; Distortion; Similarity; Log-Spiral; Lattice Structure; Area Computation; Tiling.

## ACM Reference format:

Jarek Rossignac. 2019. Corner-Operated Tran-Similar (COTS) maps, patterns, and lattices. *ACM Trans. Graph.* ??, ??, Article ?? (?? 2019), 14 pages.

[https://doi.org/0000001.0000001\\_2](https://doi.org/0000001.0000001_2)

## 1 INTRODUCTION

### 1.1 Motivation

Our primary motivation is to simplify the design of procedural models of extremely complex lattice structures [Wang and Rosen 2002] and to accelerate queries on them. We propose a compromise between: (1) **regular** lattices, which are axis-aligned matrices of translated copies of a template shape, for which queries, such as **PIT (Point-Inclusion-Testing)** [Tilove 1980] or **TAC (Total-Area-Calculation)**, can be answered in constant time and (2) Free-Form Deformations (FFD) [Sederberg and Parry 1986] of regular lattices, which broaden the design space considerably, but have a higher query cost. Reducing that cost, while still supporting a useful, although limited, range of deformations is an important goal towards supporting the parameterization and optimization of high-definition material micro-structures [Schaedler and Carter 2016] for Additive Manufacturing [Chu et al. 2008; Regli et al. 2016]. The planar **COTS lattice**, proposed here, is the image of a regular lattice by a COTS map, which is controlled by four corners points and which affords

ACM acknowledges that this contribution was authored or co-authored by an employee, or contractor of the national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only. Permission to make digital or hard copies for personal or classroom use is granted. Copies must bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. To copy otherwise, distribute, republish, or post, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2019 Association for Computing Machinery.

0730-0301/2019/??-ART?? \$15.00

[https://doi.org/0000001.0000001\\_2](https://doi.org/0000001.0000001_2)

constant (space and time) PIT and TAC queries. A **COTS map** has 8 degrees-of-freedom. Its four-dimensional distortion-space (ignoring position, orientation, and scaling) is illustrated in Fig. 1.

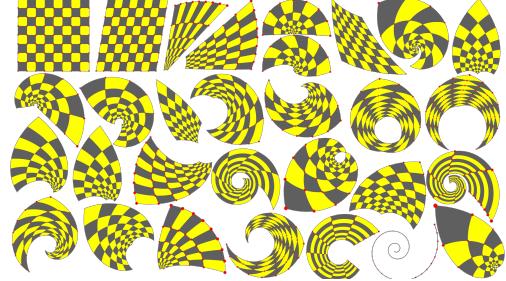


Fig. 1. Examples of the 4 DoF distortion-space of the COTS map.

Examples of COTS lattices are shown in Fig. 2.

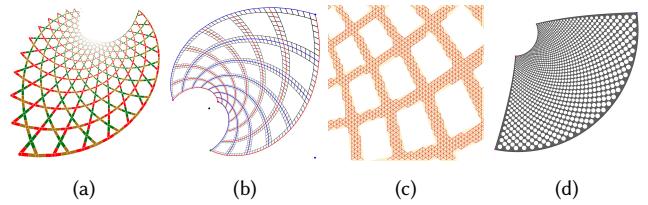


Fig. 2. A COTS lattice with Kagome connectivity with three COTS hub-patterns in red, green, and gold (a). A procedurally trimmed, multi-scale COTS lattice (b). A detail of a Lattice-in-Lattice in which a fine, regular lattice is trimmed [Liu and Shapiro 2018] efficiently to its part inside a coarse COTS lattice (c). A material structure defined by a COTS matrix of groups of two circular holes that has a constant homogenized mass (d).

### 1.2 Key contributions

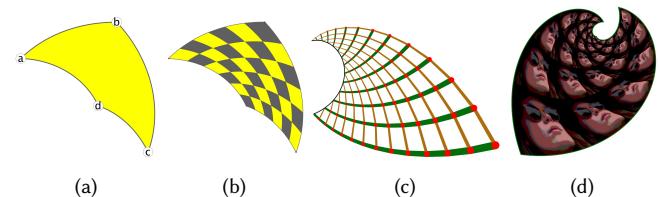


Fig. 3. The range,  $R$ , (yellow) of a COTS map and its four control corners (a), its tiles, which are similar to each other (b), its use to approximate a Michell truss (c), and its use for a “twist” on *Shot Marilyn* in the form of a Tran-Similar matrix of a template image (d).

The COTS map may: (1) Accelerate the processing of procedural models of a useful set of planar steady lattices [Gupta et al. 2019], (2) Provide an easy-to-use and effective interactive tool for designing

or animating a simple warp (of a square domain) that distributes similarity transformation steadily (adjacent tiles along each row are related by a constant similarity) [Rossignac and Vinacua 2011] and the residual distortion evenly. (3) Support, we conjecture, the design and beautification [Langbein et al. 2004] of geometric patterns [Liu et al. 2004; Mitra et al. 2013] that have a pleasing variation in element position, orientation, and scale along a curved coordinate system.

COTS maps the parametric domain,  $D = [0, 1]^2$ , to a planar region,  $R$ , bounded by 4 log-spiral segments. It is fully controlled by the four **corners** ( $a, b, c$ , and  $d$ ) of  $R$  (Fig. 3a). It maps (Fig. 3b) the square cells of an  $m \times n$  regular tiling of  $D$  onto a matrix (two-pattern) of similar tiles,  $T_{i,j}$ , for which opposite angles are identical. We provide simple and exact, closed-form expressions for computing the COTS map and its inverse and for identifying, in constant time (regardless of  $m$  and  $n$ ), the tile that contains a query point. Finally, we conjecture that COTS lattices may offer a useful generalization of some variants of Michell trusses [Skelton and de Oliveira 2010], which minimize weight for a given load (Fig. 3c), and that the COTS map may be useful for creating artistic arrangements of steady patterns (Fig. 3d).

We define our notation and terminology in Sec. 2. We compare the COTS map to prior art in Sec. 3. We define the COTS map in Sec. 4. We discuss queries on COTS matrices in Sec. 5. We discuss properties of the COTS map in Sec. 6. We discuss variants in Sec. 7. We discuss the application to lattices in Sec. 8. Proofs are in Appendix A.

## 2 TERMINOLOGY AND REVIEW OF SIMILARITIES

Here, we clarify our notation and terminology, review planar similarities, and define steady versions of “row” or “matrix” patterns. To increase insight, we formulate affine transformations in terms of points and vectors, rather than in terms of homogeneous matrices.

### 2.1 Notation

Technical terms appear in bold when first introduced. We use the following, slightly non-standard, but convenient notation.

**Integers** are in lowercase ( $i$  or  $j$ ). **Scalars** are in italics ( $u$  or  $\lambda$ ). **Points** are in bold lowercase ( $f$  or  $p_i$ ) or in parentheses using Cartesian coordinates (“ $(x, y)$ ”). **Vectors** are in lowercase bold with an overhead arrow ( $\vec{v}$ ) or in brackets using Cartesian components (“ $(x, y)$ ”). Vector from  $a$  to  $c$  is written  $\vec{ac}$ . Sum  $p + \vec{v}$  denotes  $p$  translated by  $\vec{v}$ . The **angle** between  $\vec{u}$  and  $\vec{v}$  is written  $\vec{u}^\vee \vec{v}$ . The version of  $\vec{v}$  **rotated** by angle  $\alpha$  is denoted  $\vec{v}^\circ \alpha$ . Shortcut  $\vec{v}^\perp$  stands for  $\vec{v}^\circ \frac{\pi}{2}$ . The **dot product** of  $\vec{u}$  and  $\vec{v}$  is denoted  $\vec{u} \cdot \vec{v}$ .

The **labelled edge** (closed line segment) from  $a$  to  $b$  is written  $ab$ .  $|ab|$  denotes the length of  $ab$ , and hence also the distance between  $a$  and  $b$ . The **labelled triangle** with vertices  $a, b$ , and  $c$  is written  $abc$ .  $abc \sim def$  indicates that these triangles are **similar** (i.e., related by a similarity). Other **shapes** (disk, beam) are in bold caps ( $R$  or  $T_{i,j}$ ).

**Transformations** are in curly caps ( $\mathcal{U}$  or  $\mathcal{M}$ ). **Composition** of  $\mathcal{U}$  and  $\mathcal{V}$  is written  $\mathcal{U} \cdot \mathcal{V}$ . Application of  $\mathcal{U}$  to point  $p$  (resp. shape  $X$ ) is written  $\mathcal{U} \cdot p$  (resp.  $\mathcal{U} \cdot X$ ).  $\mathcal{U} \cdot \mathcal{V} \cdot \mathcal{W} \cdot p$  means  $\mathcal{U} \cdot (\mathcal{V} \cdot (\mathcal{W} \cdot p))$ .  $\mathcal{U}^t$  is the **power** of  $\mathcal{U}$ ,  $\mathcal{U}^{-1}$  is its inverse, and  $\mathcal{U}^0$  is the identity.

Complex numbers are in bold italics and may represent points, vectors, or similarities. Let  $p$  and  $q$  represent points  $p$  and  $q$ ,  $u$  and  $w$  represent vectors  $\vec{u}$  and  $\vec{w}$ , and  $s$  represent similarity  $S$ . Vector  $\vec{u} = \langle \lambda_u \cos \alpha_u, \lambda_u \sin \alpha_u \rangle$ , where  $\lambda_u = |\vec{u}|$  and  $\alpha_u$  is the

angle from  $\langle 0, 1 \rangle$  to  $\vec{u}$ , has two **complex forms**:  $u = x_u + iy_u$  and  $u = \lambda_u e^{i\alpha_u}$ . The complex form of a similarity,  $S$ , that preserves the origin and takes vector  $\langle 1, 0 \rangle$  to vector  $\langle \lambda \cos \alpha, \lambda \sin \alpha \rangle$  is  $s = \lambda e^{i\alpha}$ . The complex form of  $S \cdot \vec{u}$  is the product  $su$ . The complex form of a similarity that takes  $\vec{w}$  to  $\vec{u}$  is  $u/w$ .

### 2.2 Proper Similarities

We focus on **planar** similarities. A **similarity**, i.e., a constrained version of an affine transformation, is the arbitrary composition of the following primitive transformations. **Translation**  $\mathcal{T}_{\vec{v}}$  by vector  $\vec{v}$  maps point  $p$  to point  $p + \vec{v}$  and maps a vector to itself. **Rotation**  $\mathcal{R}_{f,\alpha}$  by angle  $\alpha$  around  $f$  maps  $\vec{v}$  to  $\vec{v}^\circ \alpha$  and maps  $p$  to  $f + \vec{fp}^\circ \alpha$ . **Dilation**  $\mathcal{D}_{f,\lambda}$  by a uniform scaling  $\lambda$  about  $f$  maps  $\vec{v}$  to  $\lambda \vec{v}$  and maps  $p$  to  $f + \lambda \vec{fp}$ . Recall that a similarity preserves angles (i.e., is conformal) and collinearity.

Let  $S_{a,c}$  be a similarity taking  $(0, 0)$  to point  $a$  and  $(1, 0)$  to point  $c$ . If  $p = (x, y)$ , then  $S_{a,c} \cdot p = a + x \vec{ac} + y \vec{ac}^\perp$ .

**Definition 2.1.** A **proper similarity** is a similarity that is not a translation.

A proper similarity,  $S$ , has a unique **fixed point**,  $f$ , and a **canonical decomposition**:  $\mathcal{R}_{f,\alpha} \cdot \mathcal{D}_{f,\lambda}$ . Hence,  $S$  can be represented by the triplet  $\langle f, \lambda, \alpha \rangle$ . If  $S = \langle f, \lambda, \alpha \rangle$ , then  $S \cdot \vec{v} = \lambda \vec{v}^\circ \alpha$  and

$$S \cdot p = f + \lambda \vec{fp}^\circ \alpha. \quad (1)$$

### 2.3 Steady Animations and Matrices

We will define (Def. 4.4) the COTS map as the commutative composition of two Steadily-Animated Similarities.

**Definition 2.2.** A time parameterized similarity  $S(t)$  is a **Steadily-Animated Similarity (SAS)**, iff  $\exists S, \forall t, S(t) = S^t$ .

When  $S = \langle f, \lambda, \alpha \rangle$ ,  $S^t = \langle f, \lambda^t, t\alpha \rangle$ , and therefore

$$S^t \cdot p = f + \lambda^t \vec{fp}^\circ (t\alpha). \quad (2)$$

A **pattern** is a periodic arrangement of instances of a **template** shape. A one-pattern is called a **row**. A two-pattern (row of rows) is called a **matrix**. **Steady** patterns have been defined in terms of constant incremental transformations (affinities in [Rossignac and Vinacua 2011], similarities in [Gupta et al. 2019], translations in [Pasko et al. 2011]). We focus here on their planar versions.

**Definition 2.3.** Set  $\{X_m\}$  of shapes is a **steady row** iff there exist a **similarity**,  $S$ , and a template,  $X$ , such that, for each valid  $m$ ,  $X_m = S^m \cdot X$ .

**Definition 2.4.** A **translation row** is a steady row for which  $S$  is a translation.

**Definition 2.5.** The set  $\{X_{m,n}\}$  of  $m \times n$  shapes is a **steady matrix** iff there exist similarities,  $\mathcal{I}$  and  $\mathcal{J}$ , and a template,  $X$ , such that, for each valid  $(m,n)$  pair,  $X_{m,n} = \mathcal{J}^n \cdot \mathcal{I}^m \cdot X$ .

Note that we underscore ( $m$  or  $n$ ) the matrix “dimensions” (i.e., the total numbers of elements in a row along a direction).

Given a matrix  $\{X_{m,n}\}$ , let  $X_{r,*}$  (resp.  $X_{*,r}$ ) denote the **n-row** (resp. **m-row**) of its elements for which  $m=r$  (resp.  $n=r$ ).

**THEOREM 2.6.** In a steady matrix  $\{X_{m,n}\}$ , for each valid  $r$ , row  $X_{r,*}$  and row  $X_{*,r}$  are each steady. (Proof in Appendix A.1.)

**Definition 2.7.** A **Tran-Similar matrix** (abbreviated **TS matrix**) is a steady matrix for which  $\mathcal{I}$  and  $\mathcal{J}$  commute.

**Definition 2.8.** A **translation matrix** is a steady matrix for which  $\mathcal{I}$  and  $\mathcal{J}$  are translations.

**Definition 2.9.** A **regular matrix** is a translation matrix for which  $\mathcal{I}$  translates by  $\langle 1, 0 \rangle$  and  $\mathcal{J}$  translates by  $\langle 0, 1 \rangle$ .

### 3 COMPARISON TO PRIOR ART

#### 3.1 COTS maximizes evenness of distortion

A vast amount of prior art is devoted to the design and animation of planar maps that minimize some measure of distortion. Some authors focused on conformal maps, which always exist (Riemann Mapping Theorem) [Segall and Ben-Chen 2016; Vaxman et al. 2015]. Others focused on solutions with a bounded distortion, provided that the map does not produce fold-overs or other visual artifacts [Chen et al. 2013; Poranne and Lipman 2014]. A restriction to polygonal regions has been studied extensively (see [Schneider and Hormann 2015] for an example and further bibliography). Various solutions have been proposed under the term Generalized Barycentric Coordinates [Hormann and Floater 2006; Weber et al. 2009]. Some solutions improve the quality of the mapping by giving up interpolation of the boundaries [Weber and Gotsman 2010]. Several approaches minimize quasiconformal distortion [Weber et al. 2012], for example by decoupling the rotation and scaling to simplify minimization [Chern et al. 2015].

Solutions to the problem of producing a good quality animation of a planar or space warp include the As-Rigid-As-Possible shape interpolation, which strives to compute the least distorting morphs between two planar [Chen and Gotsman 2016; Chien et al. 2016] or volumetric [Alexa et al. 2000] shapes. These approaches have been extended to extreme deformations that may include changes in topology [Zhu et al. 2017]. For example, the authors of [Nian and Chen 2016] start with a Teichmüller map (see [Jin et al. 2009]) and then tune it to produce a linear variation of conformal distortion over time.

We argue that the conformal distortion is not the best tool for measuring the quality of a planar map when that map is used to design the distortion of an image (or of a regular matrix of tiles or shapes). To support this statement, in Fig. 4, we show the circumcircle of the corners of domain,  $D$ , warped by four different COTS maps. (Examples of the warps of that circle by the restricted FPI map are shown in [Lipman et al. 2012].) We do not believe that the measure of conformal distortion captures the nature and amount of the dissimilarity between these warped circles, nor that it adequately captures the similarity of the distortions of the different tiles of the map.

We do not claim that the COTS map minimizes a particular measure of distortion between a shape and its image. In fact, we believe that doing so is not a useful goal for planar warps, when these are designed precisely to distort an image or pattern. Instead, we advocate that a second, arguably more important measure of the quality of a map is the **evenness of distortion**: We want all parts of the

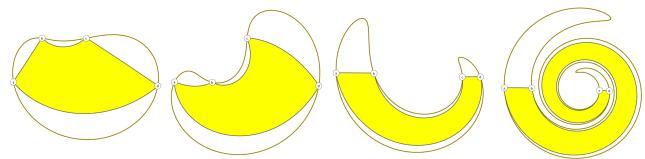


Fig. 4. Images by four COTS maps of the circle through the corners of the unit square. Range  $R$  is filled in yellow. The disparity of their distortions cannot be captured by a single scalar measure.

image (or all tiles of the regular matrix) to be distorted identically. We therefore suggest that, for some applications, COTS may be the best CO map, because (as implied by Props. 2 and 3):

**THEOREM 3.1.** COTS maximizes evenness of distortion.

#### 3.2 COTS is the first TS map that is also CO

The COTS map is the only map known to be both TS and CO. By TS (Tran-Similar), we mean that it maps translations to similarities. In other words, that if two shapes are related by a translation, then their COTS images are related by a similarity. By CO (Corner-Operated), we mean that the user (or an optimization or animation program) may fully control the map using the four corners of its range  $R$  in a manner that is affine-invariant (independent of the choice of the global frame) and symmetric (not affected by cyclic permutations or reversions of the labels associated with these vertices).

Simple TS, but non-CO, maps include Isometries, Similarities, Affinities, and Log-Polar maps. All these are special cases of the Augmented COTS map (Sec. 6.1). To appreciate the benefits of extending these simple TS maps to the more general COTS map, we consider the total number of **Degrees-of-Freedom (DoF)** of a map and separate it into the **degrees of distortion (DoD)**, and the (up to) 4 DoF for the similarity that controls the position, uniform scaling, and orientation. The distortion DoD is 0 for a similarity and 2 for an affinity. The benefit of the COTS map over simple TS maps is that it has a DoF of 8 and hence a DoD of 4. We may think for example that  $a$  and  $b$  control the similarity and that  $c$  and  $d$  control the distortion of the COTS map. To appreciate what such an increase of the DoD means in practice, assume that we quantize each coordinate of the control corners to 100 possible values. An affine map can generate 10,000 possible distortions. The COTS map can generate 100,000,000 possible distortions. Its superior distortion range may have significant benefits for design creativity and for the optimization of architected (material) structures.

#### 3.3 Examples of non-TS maps

Consider, the bilinear map:  $\mathcal{B}(x, y) = \mathcal{I}(\mathcal{I}(a, y, b), x, \mathcal{I}(d, y, c))$ , where  $\mathcal{I}(a, y, b)$  is the LERP  $a + y\vec{ab}$ . Its drawbacks are that the tiles may have drastically different distortions (Fig. 5a) and that its range,  $R$ , may fold (Fig. 5c). COTS of course does not suffer from these defects (Fig. 5b and 5d).

The **Möbius map** is defined (using complex numbers) by  $\mathcal{M}(z) = (pz + q)/(rz + s)$ , with  $ps \neq qr$ . It is popular because of its beautiful mathematical properties: It preserves angle (between directions from the same point) and it preserves clines (curves that are either

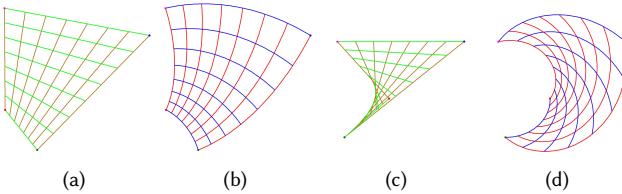


Fig. 5. Bilinear (a) and COTS map (b) for the same quadruplet. The bilinear map may fold (c). The COTS map does not (d).

a circle or a line). But it is not TS. Furthermore, it is not CO and has a DoD of only 2. Indeed, given corners  $a$ ,  $b$ , and  $c$ , corner  $d$  is constrained to a particular point along the circumcircle of  $a$ ,  $b$ , and  $c$ . In Fig 6, we compare it to a COTS map with the same corners and illustrate the additional distortion DoFs of the COTS map.

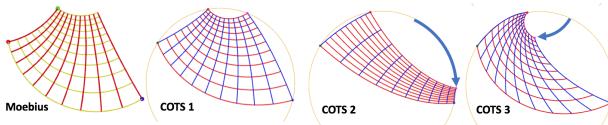


Fig. 6. Möbius is controlled by three corners:  $a_m$  (red),  $b_m$  (green), and  $d_m$  (blue). Its fourth corner,  $c_m$ , lies on their circumcircle,  $C$ . Three COTS maps with corners  $a = a_m$ ,  $b = b_m$ ,  $c$ , and  $d = d_m$ : COTS 1 has  $c = c_m$  and resembles Möbius, but has log-spiral isocurves. COTS 2 with  $c$  at a different point on circle  $C$ . COTS 3 with corner  $c$  no longer on circle  $C$ .

The authors of [Lipman et al. 2012] compare several CO, but non-TS, maps: Four-Point Interpolant (FPI) [Lipman et al. 2012], Projective warping (PROJ) [Perny et al. 1982], Mean Value Coordinates (MVC) [Floater 2003; Ju et al. 2005], Moving Least-Squares with SIMilarity transformations (MLS-SIM) [Schaefer et al. 2006], Least-Squares Conformal Maps (LSCM) [Igarashi et al. 2005; Lévy et al. 2002], Cauchy-Green coordinates with point to point (CG-P2P) [Weber et al. 2009], and As-Rigid-As-Possible deformation (ARAP) [Igarashi et al. 2005]. For each, the authors of [Lipman et al. 2012] visualize a measure of the distortion and show that “FPI has a constant conformal distortion, lower than the maximal conformal distortion of the other maps”. They highlight the distortion and fold-overs of some of the other maps.

### 3.4 Restricted FPI is not TS

The **Four-Point Interpolant (FPI)** [Lipman et al. 2012] is particularly relevant to the COTS map because its restriction to a map from the unit square  $D$  to a range  $R$  produces tilings that, at least for typical configurations, are close to those of the COTS map with the same corners (Fig. 7-left) and because it “minimizes conformal distortion and distributes it evenly”, even though FPI is not TS (Fig. 7-right).

The full FPI is the composition of the inverse of a Möbius map with an affinity and with another Möbius map. Each of the Möbius maps is computed so as to take the corners of some unspecified parallelogram to a quadruplet of control corners. The affinity relates these two parallelograms. Hence FPI may be used as a map from one curved quad to another, each controlled by a different quadruplet. FPI is comparable to the Copy-Warp-And-Paste variant (Sec. 7.6).

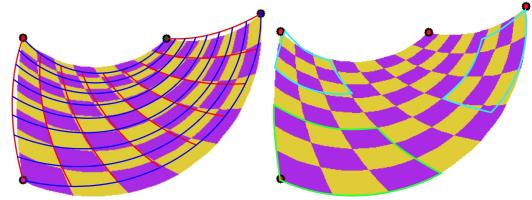


Fig. 7. The isocurves (red and blue) of a COTS map drawn over the tiling of a restricted FPI map (from [Lipman et al. 2012]) with the same control quadruplet (left). To show that FPI is not TS, we draw (right) the border (cyan) of the lower-left  $3 \times 3$  grid of its tiles and two of its copies transformed by similarities to best align them with other  $3 \times 3$  grids of tiles.

of COTS. In Fig. 7, we use an image from [Lipman et al. 2012] for which the first quadruplet are the vertices of a unit square (with an arbitrary unit measure). This **restricted FPI map** is controlled by the four corners of its range, and hence is CO. It is the composition,  $\mathcal{A} \cdot \mathcal{M}$ , of an affinity with a Möbius map. In contrast,

**THEOREM 3.2.** *A COTS map is the composition  $\mathcal{S}_{f,a} \cdot \mathcal{P} \cdot \mathcal{L}$  of a linear transformation,  $\mathcal{L}$ , with a log-polar map,  $\mathcal{P}$ , and with a similarity,  $\mathcal{S}_{f,a}$  (Sec. 2.2). (Proof in Appendix A.5.)*

How can the restricted FPI map, which is non-TS, have “constant (equally distributed) conformal distortion everywhere (that is the differential of the map has constant ratio of maximal to minimal singular values)” [Lipman et al. 2012]? To understand how, observe that the measure of conformal distortion only captures one particular aspect of the distortion and that the directions associated with these “singular values” in the local frame of each shape instance are not identical for all instances. The fact that a map with equally distributed conformal distortion produces a matrix of dissimilar instances is illustrated both for the Möbius map and for the restricted FPI map in Fig. 8.

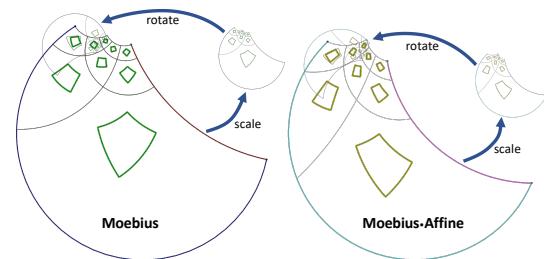


Fig. 8. Distortions of a regular  $3 \times 3$  matrix of squares by a Möbius map (left) and by its composition with an affine map (right), which corresponds to the restricted FPI map. We overlay a scaled and rotated copy of each to show that these maps are not TS.

### 3.5 COTS may provide a variant of the Michell truss

We conjecture that the simple COTS lattice can provide a useful variant for some configurations of the Michell truss [Michell 1904], of which the elastic bars follow an orthogonal systems of log-spirals and which minimizes the total volume, while providing the desired

resistance to a given load. The Symmetric Right-Angle COTS lattice (Sec. 6) shown in Fig. 9a appears identical to the pattern of lines of principal action shown in [Hemp 1973] as the analytic solution to the Michell truss. This observation suggests that Tran-Similarity may play a role in lattice optimization and in generalizations of the Michell truss [Bouchitté et al. 2008]. For example, in Fig. 9b, we show a portion of a Symmetric Right-Angle COTS lattice trimmed to remove bars that lie inside the disk of center  $f$  and radius  $|fd|$ . Hence, we assume that the lattice nodes on that circular arc (black curve from  $b$  to  $d$ ) are fixed and that the load is applied at  $a$ . In Fig. 9c we show more general, non-symmetric configurations, where the curve of fixed nodes is no longer circular, but a portion of a log-spiral. The COTS lattice formulation provides simple and efficient algorithms for generating such lattices, for adjusting the shape of that log-spiral and the repetition count (Fig. 9d), and for computing the total mass of the lattice.

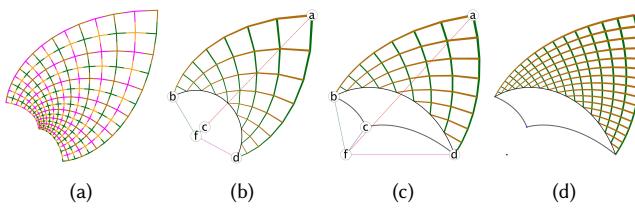


Fig. 9. The Symmetric Right-Angle COTS lattice (a) appears identical to a pattern of lines of principal action shown in [Hemp 1973] as the analytical solution to the Michell truss. Another Symmetric Right-Angle COTS lattice (b), from which we removed all bars inside the disk of center  $f$  and radius  $|fd|$ . A non-symmetric version of it (c) and its refinement (d).

### 3.6 COTS Lattice-Slabs

The planar COTS lattice, proposed here, may be “fleshed out” (replacing disks by balls and flat bars by cone sections) into a 3D **Lattice-Slab** (Fig. 10-left). A COTS Lattice-Slab may be used as template for a **Lattice-Brick** (Fig. 10-right) by forming a steady row of slabs and connecting them with inter-slab beams.

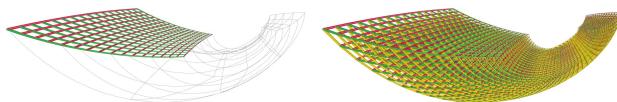


Fig. 10. A COTS Lattice-Slab (left) is used to make a Lattice-Brick (right).

Processing the image of a regular  $n \times n$  Lattice-Slab by a non-TS map requires  $O(n^2)$  algorithms for computing its integral properties (surface-area, mass, centroid) and requires rectifying the balls and beams. The use of Planar Steady Lattices (Sec. 8.5), which are controlled by 5 points, eliminates rectification and reduces the cost of integral properties to  $O(n)$  [Gupta et al. 2019]. The PMC (Point-Membership Classification) cost of establishing whether a candidate point is inside a given PSL is also  $O(n)$  [Kurzeja and Rossignac 2019]. The COTS lattices introduced here resemble constrained Steady Lattices (Sec. 8.2), but produce more regular matrices of beams and

reduce to  $O(1)$  the cost of PMC and of the integral property calculations mentioned above. These benefits are essential for iterative optimization of lattices.

The computational cost of performing PMC or of computing the volume or other integral properties on a slab of  $1000 \times 1000$  hubs increases a thousand times, when, rather than using a COTS Lattice-Slab, we use a Steady Lattice Slab [Gupta et al. 2019; Kurzeja and Rossignac 2019] and by a million times when we use an unsteady lattice.

## 4 MATHEMATICAL FOUNDATIONS OF THE COTS MAP

In this section, we use complex numbers to define key concepts and to prove the key results for the proposed COTS map.

### 4.1 Complex form of a proper similarity

Let  $\mathcal{U} = \langle f, \lambda, \alpha \rangle$  be a proper similarity that combines a rotation by  $\alpha$  and a dilation by  $\lambda$  about the same fixed point  $f$ . We represent it in complex form by the tuple  $\langle f, \mathbf{u} \rangle$ , where  $\mathbf{u} = \lambda e^{i\alpha}$ . The complex form of  $\mathcal{U} \cdot \mathbf{p}$  is

$$\mathcal{U} \cdot \mathbf{p} = f + \mathbf{u}(\mathbf{p} - f). \quad (3)$$

Let  $SIM(a, b, d, c)$  be a proper similarity that maps  $ab$  to  $dc$ .

**THEOREM 4.1.** *The complex form of  $SIM(a, b, d, c)$  is*

$$\langle \frac{ac - bd}{a - b + c - d}, \frac{c - d}{b - a} \rangle. \quad (4)$$

The proof in Appendix A.2.

### 4.2 Definitions of CO, TS, and COTS maps

A planar map,  $\mathcal{M}$ , takes point  $\mathbf{p} = (x, y)$  to a point, which we write as  $\mathcal{M}(x, y)$ ,  $\mathcal{M}(\mathbf{p})$ , or  $\mathcal{M} \cdot \mathbf{p}$ , whichever most clearly conveys the concept being presented. Note that, for readability, we denote the input parameters using  $(x, y)$ , rather than the more traditional  $(u, v)$ .

**Definition 4.2.** A map,  $\mathcal{M}$ , is **Corner-Operated (CO)** if it satisfies the following four constraints:

$$\mathcal{M}(0, 0) = a, \mathcal{M}(0, 1) = b, \mathcal{M}(1, 1) = c, \mathcal{M}(1, 0) = d, \quad (5)$$

which guarantee that  $\mathcal{M}$  is point-interpolating, i.e., that it maps the four corners of the unit-square, parametric domain,  $D$ , onto given **quadruplet**  $\{a, b, c, d\}$  of **control-corners**.

Recall that  $\mathcal{T}_{\vec{v}}$  denotes the translation by vector  $\vec{v}$ .

**Definition 4.3.** A map,  $\mathcal{M}$ , is **Tran-Similar (TS)**, iff, for every vector  $\vec{v}$ , there exists a similarity  $\mathcal{S}$ , such that, for any point  $\mathbf{p}$  and for any scalar  $t$ :

$$\mathcal{M}(\mathcal{T}_{\vec{v}}^t \cdot \mathbf{p}) = \mathcal{S}^t \cdot \mathcal{M}(\mathbf{p}). \quad (6)$$

We propose below a definition of a particular CO map, which we call **COTS**, and show that it is TS.

**Definition 4.4.** The CO map,  $\mathcal{M}$ , defined by the ordered quadruplet of control-corners,  $\{a, b, c, d\}$  is a **COTS map** if

$$\mathcal{M}(x, y) = \mathcal{V}^y \cdot \mathcal{U}^x \cdot a, \quad (7)$$

where  $\mathcal{U} = SIM(a, b, d, c)$  and  $\mathcal{V} = SIM(a, d, b, c)$ .

Note that  $\mathcal{U}$  maps  $ab$  to  $dc$  and  $\mathcal{V}$  maps  $ad$  to  $bc$ , and that COTS moves corner  $a$  by  $\mathcal{U}^x$  and then by  $\mathcal{V}^y$ .

### 4.3 Four-Point Similarity-Center (FPSC)

The key property of the COTS map and its benefits for the efficient processing of COTS lattices stem from the following:

**THEOREM 4.5.** *SIM(a, b, d, c) and SIM(a, d, b, c) have the same fixed point (Fig. 11a and b).*

**PROOF.** Swapping  $b$  and  $d$  in expression 4 does not change  $f$ .  $\square$

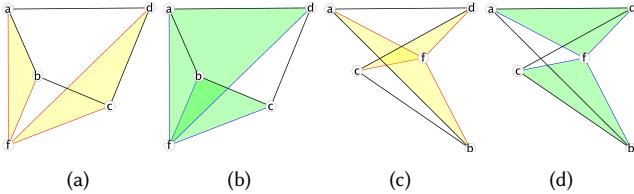


Fig. 11. Given the four corners  $\{a, b, c, d\}$  of a non-parallelogram quad. Its Four-Point Similarity-Center,  $f$ , satisfies  $fab \sim fdc$  (a) and  $fad \sim fbc$  (b). This property holds even when the quad is self-crossing (c and d).

**Definition 4.6.** The **Four-Point Similarity-Center (FPSC)** of the quadruplet  $\{a, b, c, d\}$  is the common fixed point of  $SIM(a, b, d, c)$  and of  $SIM(a, d, b, c)$ .

Let  $F(a, b, c, d)$  denote the FPSC of this quadruplet. Th. 4.5 holds even when the quad is self-crossing (Fig. 11c and d). Using Expression 4, one can verify that  $f$  is preserved by a cyclic permutation ( $F(a, b, c, d) = F(b, c, d, a)$ ), and by swapping diagonally-opposite corners ( $F(a, b, c, d) = F(a, d, c, b)$ ). But, swapping two consecutive corners will, in general, not preserve  $f$ .

Consider a quadruplet  $\{a, b, c, d\}$  for which  $\vec{ab} \neq \vec{dc}$ . Let  $\mathcal{U}$  be the similarity  $SIM(a, b, d, c)$  that takes oriented edge  $ab$  to oriented edge  $dc$ . We can compute its canonical representation,  $\langle f, \lambda, \alpha \rangle$  (Sec. 2.2), using Expression 4 or:

$$\lambda = \frac{|dc|}{|ab|}, \quad \alpha = \vec{ab} \vee \vec{dc}, \quad f = a + \frac{\langle \vec{w} \cdot \vec{ba}, \vec{w} \cdot \vec{ba} \rangle}{d}, \quad (8)$$

with  $\vec{w} = (\lambda \cos \alpha - 1, \lambda \sin \alpha)$  and  $d = \vec{w}^2$ .

The determinant  $d$  of this system is  $\vec{w}^2 = (\lambda \cos \alpha - 1)^2 + (\lambda \sin \alpha)^2$ . It is null when both  $\alpha = 0$  and  $\lambda = 1$ , hence, when  $\vec{ab} = \vec{dc}$ .

### 4.4 A COTS map is TS

It is easy to show that, if  $\mathcal{U} = \langle f, u \rangle$  and  $\mathcal{V} = \langle f, v \rangle$ , then the complex form of  $\mathcal{M}$  can be written:

$$\mathcal{M}(x, y) = f + u^x v^y (a - f), \quad (9)$$

**LEMMA 4.7.** *If  $\mathcal{U} = SIM(a, b, d, c)$  and  $\mathcal{V} = SIM(a, d, b, c)$  then  $\mathcal{U}$  and  $\mathcal{V}$  commute, i.e.,  $\mathcal{U} \cdot \mathcal{V} = \mathcal{V} \cdot \mathcal{U}$*

**THEOREM 4.8.** *A COTS map,  $\mathcal{M}$ , is Tran-Similar. (Proof in A.3.)*

### 4.5 Evaluating the COTS map and its inverse

To compute our representation of a COTS map, given quadruplet  $\{a, b, c, d\}$ , we compute its fixed point  $f$  (Sec. 4.3) and its two pairs of ratios and angles, as follows:

$$\lambda_v = \frac{|dc|}{|ab|} \text{ and } \lambda_u = \frac{|bc|}{|ad|}, \quad (10)$$

$$\alpha_v = \vec{ab} \vee \vec{dc} + 2\pi k_v \text{ and } \alpha_u = \vec{ad} \vee \vec{bc} + 2\pi k_u \quad (11)$$

where  $k_u$  and  $k_v$  are updated at each frame during interactive editing or animation to select the **branch** that minimizes changes in the COTS map in  $\alpha_u$  and  $\alpha_v$ . A COTS map is represented by points  $a$  and  $f$ , and by the four parameters:  $\lambda_u$ ,  $\lambda_v$ ,  $\alpha_u$ , and  $\alpha_v$ .

To compute the image  $\mathcal{M}(p)$  of point  $p = (x, y)$ , we use

$$\mathcal{M}(x, y) = f + \lambda_u^x \lambda_v^y \vec{fa}^\circ (x\alpha_u + y\alpha_v). \quad (12)$$

The inverse of a COTS map also has a closed-form expression.

**THEOREM 4.9.** *Given  $q = \mathcal{M}(x, y)$ , parameters  $x$  and  $y$  are the solutions of the following linear system (proof in A.4):*

$$x \ln \lambda_u + y \ln \lambda_v = \ln \lambda \quad (13)$$

$$x\alpha_u + y\alpha_v = \alpha \quad (14)$$

However, in configurations where  $k_u \neq 0$  or  $k_v \neq 0$  we must consider all branching options and hence, solve the above system with  $\lambda_u$  and  $\lambda_v$  defined by Eq. 10,  $\alpha_u$  and  $\alpha_v$  defined by Eq. 11,

$$\lambda = \frac{|\mathbf{f}\mathbf{q}|}{|\mathbf{f}\mathbf{a}|}, \quad \text{and } \alpha = \vec{fa} \vee \vec{fq} + 2\pi k, \quad (15)$$

where integer  $k$  identifies a **branching option**. We explain below how we generate and test a sufficient set of these options.

The determinant of this system is  $d = \alpha_v \log \lambda_u - \alpha_u \log \lambda_v$ .

When  $d = 0$ , there is no solution. This may happen when the quad is a parallelogram, or a point, in which case, the inverse is not defined.

**Otherwise**, we find the limits  $\alpha_{min} = \min(0, \alpha_u, \alpha_v, \alpha_u + \alpha_v)$  and  $\alpha_{max} = \max(0, \alpha_u, \alpha_v, \alpha_u + \alpha_v)$  and, for each candidate value of  $k$  for which  $\alpha_{min} \leq \alpha \leq \alpha_{max}$ , we compute coordinates

$$x = (\alpha_v \log \lambda - \alpha \log \lambda_v)/d \quad (16)$$

$$y = (\alpha - x\alpha_u)/\alpha_v. \quad (17)$$

The **valid candidates** are those for which  $x$  and  $y$  are both in  $[0, 1]$ .

### 4.6 Isocurves of a COTS map

A **u-curve** is an isocurve of points  $\mathcal{M}(*, y)$ , for which  $x$  varies, but not  $y$ . Similarly, the term **v-curve** refers to  $\mathcal{M}(x, *)$ .

**PROPERTY 1.** *A Tran-Similar map takes lines to log-spirals and constant velocity motions to Steadily-Animated Similarities.*

Consequently, isocurves, and hence borders of R are log-spirals (Fig. 12). The borders never cross in simple branching configurations (for which  $k_u \neq 0$  and  $k_v \neq 0$ ).

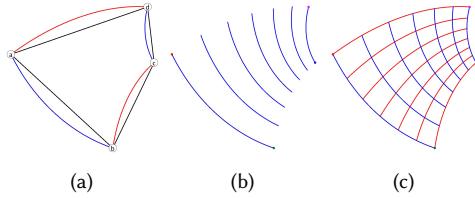


Fig. 12. The COTS map border (blue and red) drawn (a) over the control polygon (black). The steady one-pattern (b) of seven v-curves (blue). The overlay of the v- and u-curves (red) decomposes  $R$  into a grid of  $6 \times 6$  tiles (c), which form a COTS matrix (Sec. 5.1).

## 5 QUERIES ON COTS MATRICES

### 5.1 Tiles and patterns

**Definition 5.1.** A **COTS matrix** is a Corner-Operated TS matrix, i.e., a Tran-Similar matrix (Def. 2.8) of which the layout is controlled by the four corners of the range  $R$  of the associated COTS map.

To visualize the uniformly distributed distortion of the COTS map, we show it on square tiles of  $D$ . Hence, we set  $\underline{m} = \underline{n}$  in our examples and further discussion.

Consider the set of  $(\underline{n}+1)$  v-curves, (blue in Fig. 12b) for  $x=i/\underline{n}$ , where integer  $i$  varies from 0 to  $\underline{n}$ . Similarly, consider the set of  $(\underline{n}+1)$  u-curves (red in Fig. 12c) for  $y=j/\underline{n}$ , where integer  $j$  varies from 0 to  $\underline{n}$ . These two families of isocurves decompose  $R$  into a COTS matrix (Def. 5.1) of  $\underline{n} \times \underline{n}$  tiles.  $T_{i,j}$  denotes the tile bounded by segments of v-curves for  $x = i/\underline{n}$  and  $x = (i + 1)/\underline{n}$  and of u-curves for  $y = j/\underline{n}$  and  $y = (j + 1)/\underline{n}$ .

In Fig. 13, we show an  $\underline{m} \times \underline{n} = 4 \times 4$  COTS matrix of tiles controlled by corners  $\{a, b, c, d\}$ . We can use it to create an  $\underline{m} \times \underline{n}$  **COTS matrix of shapes** in which each instance is associated with a tile. For clarity, we placed the template so that each instance fits inside its tile (Fig. 13a). We can also use the same tiles to create an  $(\underline{m}+1) \times (\underline{n}+1)$  COTS matrix of shapes in which each instance is associated with a grid-point of that tiling. For clarity, we placed the template so that each instance is centered around its grid-point (Fig. 13b).

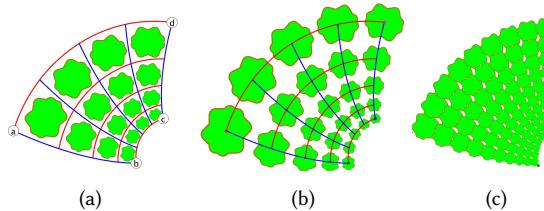


Fig. 13. A  $4 \times 4$  COTS matrix of tiles can be used to define a  $4 \times 4$  (a) or a  $5 \times 5$  (b) COTS matrix of shapes. Adjacent instances may overlap (c).

### 5.2 Finding the tiles containing a query point

Given the coordinates  $(x, y)$  of the preimage (Sec. 4.5) of query point  $q$ , we use floor operators to compute, in constant time, the indices of tile that contains  $q$ :  $i = \lfloor x\underline{n} \rfloor$  and  $j = \lfloor y\underline{n} \rfloor$ . This is an important improvement over the naïve,  $O(\underline{n}^2)$ , approach of testing

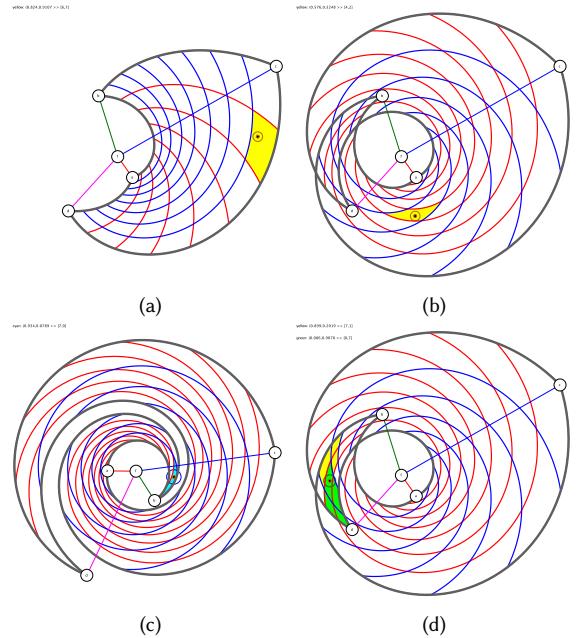


Fig. 14. Inversion: We fill the tile that contains the query point  $q$  indicated by a dot in a red circle. We paint it yellow (a, b, and d), when  $k=0$ . We paint it cyan (c), when  $k=1$ . We paint it green (d), when  $k=-1$ . Note that, when  $R$  self-overlaps (b and d), point  $q$  may be contained in more than one tile (d).

$q$  against each tile, and over the improved,  $O(\underline{n})$ , **Range-Finder** solution proposed in [Kurzeja and Rossignac 2019].

In Fig. 14, we color-code the tiles containing  $q$  to indicate the corresponding branch number  $k$  (see Sec. 4.5). Note that, when  $R$  self-overlaps and  $q$  lies inside more than one tile, several candidate inverses are valid and must be considered by the application.

### 5.3 Point-Inclusion-Testing (PIT)

To test whether a given query point,  $q$ , is contained inside the union of the shapes of a COTS matrix of shapes  $\{X_{i,j}\}$ , in a pre-processing step, we consider each one of the nine **sets of tiles**: 4 **corner-tiles**, 4 rows of **border-tiles**, and the remaining set of **inner-tiles**. For each set of tiles, we pick one tile to be the **reference-tile**. For each one of these nine reference-tiles, we identify and record the list of shape instances that intersect it. In our representation, we identify each instance by its two index-offsets. So, for example, if the reference tile of a given set is  $\{T_{i,j}\}$ , and if it intersects instance  $\{X_{i,j}\}$ , we store the index-offset pair  $(0,0)$ . If it also intersects instance  $\{X_{i+1,j}\}$ , we also store index-offset pair  $(1,0)$ .

Then, given query point,  $q$ , we establish the list of tiles that contain it (possibly more than one, as discussed in Sec. 5.2). For each containing tile,  $T_{i,j}$ , of this list, we identify which set (corner-, border-, inner-) the containing tile  $T_{i,j}$  belongs to. Then, we identify the representative tile  $T_{i',j'}$  of that set. Then, we compute the point  $q'$  that is to  $T_{i',j'}$  what  $q$  is to  $T_{i,j}$ . (We obtain  $q'$  from  $q$  by (1) computing the preimage of  $q$ , (2) translating it by  $((i' - i)/\underline{n}, (j' - j)/\underline{n})$ , and (3) applying the COTS map to the result.) Finally, we classify  $q'$  against the list of shape-instances associated with  $T_{i',j'}$ .

## 5.4 Total Area Calculation (TAC) of a COTS matrix

The total surface area of a COTS matrix of disjoint shape instances  $X_{i,j}$ , such as shown in Fig. 13a and 13b, may be computed as the sum of the areas  $a_{i,j}$  of the instances. Given the area  $a_{0,0}$  of the template shape  $X_{0,0}$ , the area  $a_{i,j}$  of instance  $X_{i,j}$  is  $\lambda_v^{2j/n} \lambda_u^{2i/n} a_{0,0}$ . The formula for the total area,  $a_{total}$ , is the cascade of two sums of geometric progressions. Building on [Gupta et al. 2019], we compute it directly, without iterations, using the closed-form expression:

$$a_{total} = \sum_{j=0}^{n-1} \lambda_v^{2j/n} \left( \sum_{i=0}^{n-1} \lambda_u^{2i/n} a_{0,0} \right) = a_{0,0} \left( \frac{\lambda_v^{2n} - 1}{\lambda_v^{2/n} - 1} \right) \left( \frac{\lambda_u^{2n} - 1}{\lambda_u^{2/n} - 1} \right).$$

We propose another, constant-cost approach for computing the total area of a COTS matrix of disjoint shapes. Let  $s$  denote the area of the template shape,  $X_{0,0}$ ,  $t$  denote the area of the template tile,  $T_{0,0}$ , and  $r$  denote the total area of  $R$ . The total area of the COTS matrix is  $a = sr/t$ . Hence, it may be computed in constant time, independently of the value of  $n$ . This simple closed-form, constant-cost expression for the area of a COTS pattern can be extended to configurations illustrated in Fig. 13b by suitably enlarging the domain  $D$  and by splitting it into  $(n+1) \times (n+1)$  tiles such that each  $X_{i,j}$  is fully contained in tile  $T_{0,0}$ . It may be further extended to configurations where  $X_{i,j}$  is not fully contained in tile  $T_{0,0}$ , provided that the shapes of  $\{X_{i,j}\}$  are pairwise disjoint. It may even be extended to configurations where adjacent instances overlap (as illustrated in Fig. 13c), but such an extension may require the separate processing of the four corner-tiles and of the four rows of border-tiles (see Sec. 5.3).

## 6 PROPERTIES OF THE AUGMENTED COTS MAPS

**Definition 6.1.** The **Augmented COTS map** (abbreviated **ACOTS**) uses a bilinear interpolation (Sec. 3.3) for cases where the control quadruplet is a parallelogram.

When the quadruplet approaches a parallelogram configuration, COTS approaches ACOTS, hence producing a continuous behavior during interactive editing or animation. As demonstrated in Fig. 15, that function is continuous, even when the motion of the quadruplet produces a self-crossing control polygon (Fig. 15b), yields a degenerate configuration, where  $R$  is collapsed to a log-spiral curve (Fig. 15c), and restores a non-degenerate configuration (Fig. 15d).

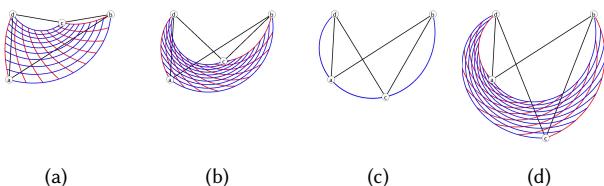


Fig. 15. Starting from an initial deformation (a), we drag control corner  $c$  downward. The COTS map evolves in a continuous manner, even when the control-polygon becomes self-crossing (b), yields a degenerate configuration, where  $R$  is collapsed to a log-spiral curve (c), and restores a non-degenerate configuration, but with reversed orientation (d).

For simplicity, through the rest of the paper, we assume that the term COTS refers to ACOTS.

Tran-Similarity (Th. 4.8 ) implies the following.

**PROPERTY 2.** If two shapes are related by a translation, their images by a COTS map are related by a similarity.

**PROPERTY 3.** COTS images of rectangular blocks of same repetition-count in each direction in a regular pattern of tiles or of shapes are similar to each other (Fig. 16-left).

The control polygon may be easily arranged so that two opposite border edges of  $R$  lie on the same log-spiral and so that the tile-borders abutting on that spiral from both sides align perfectly (Fig. 16-right).

These properties hold for tiles and shape matrices produced by extending the COTS map outside of  $R$ .

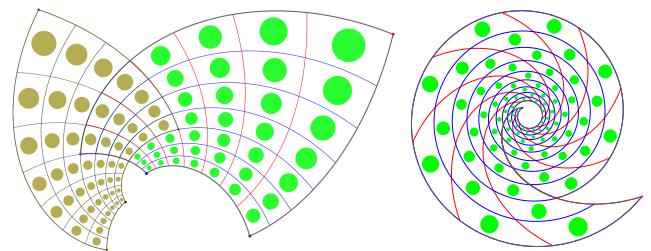


Fig. 16. Tran-Similarity makes it possible to overlap two similar copies of a COTS map so that the tiles and shapes (disks here) in the overlap region (here a  $3 \times 4$  block) match perfectly (left). It also makes it possible to ensure that the borders of tiles along opposite edges of  $R$  align perfectly (right).

**PROPERTY 4.** The opposite angles of each tile of a COTS map are identical.

By foregoing one degree of freedom, we can constrain the arrangements of control corners so as to ensure that the angles at all four corners of each tile are right angles. This happens when the following constraint is satisfied:  $\tan^{-1}(\alpha_u/\ln \lambda_u) - \tan^{-1}(\alpha_v/\ln \lambda_v) = \pi/2$ . We use the term **Right-Angle COTS map** to identify these special cases. In Fig. 17, we show (as a thick black stroke), for a configuration of the three control corners  $b$ ,  $c$ , and  $d$ , the curve where control corner  $a$  should lie so as to produce a Right-Angle COTS map.

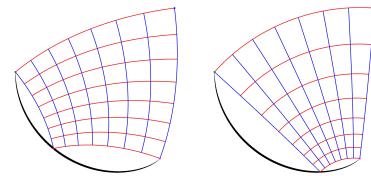


Fig. 17. We show two different Right-Angle COTS maps, for which isocurves cross orthogonally. We also show (thick black stroke) the curve along which we can slide corner  $a$  while maintaining this property.

In general, a Right-Angle COTS map is not conformal. For example, in Fig. 18b, we trace the images (green and brown) of two lines orthogonal in parameter space (Fig. 18a). We observe (Fig. 18b) that the two curves do not cross at right angle. They do (Fig. 18c), when the Right-Angle COTS map is symmetric (when  $b$  and  $d$  are symmetric with respect to the line through  $a$  and  $c$ ).

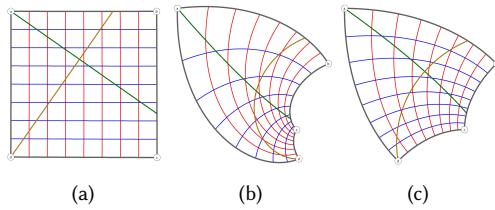


Fig. 18. Given two orthogonal lines (green and brown) in the parameter-space (a), their images under a Right-Angle COTS map (b) need not cross at right angle. But, when the Right-Angle COTS map is symmetric, they do (c).

## 7 NON-TS VARIANTS AND APPLICATIONS

### 7.1 PM variant

To morph from  $\vec{v}_0$  to  $\vec{v}_1$ , COTS uses the **Log-Polar Morph (LPM)**, which yields  $\vec{v}(t) = \lambda^t \vec{v}_0^\circ(\alpha t)$ , with  $\lambda = |\vec{v}_1| / |\vec{v}_0|$  and  $\alpha = \vec{v}_0^\vee \vec{v}_1$ . This motion is **steady** [Rossignac and Vinacua 2011], meaning that the similarity between  $\vec{v}(t)$  and  $\vec{v}(t+u)$  is independent of  $t$ . Indeed,  $\vec{v}(t+u) = \lambda^{t+u} \vec{v}_0^\circ(\alpha t + u\alpha) = \lambda^u (\vec{v}(t))^\circ(u\alpha)$ . The **Polar Morph (PM)** uses a linear, rather than exponential, interpolation of the magnitudes. We compare PM and LPM in Fig. 19 to the linear interpolation (**LERP**) and to **SLERP** [Shoemake 1985], which uses  $\vec{w}(t) = \langle \sin((1-t)\alpha), \sin(t\alpha) \rangle / \sin \alpha$ . In general, LERP, SLERP, and PM are not steady – LPM always is.

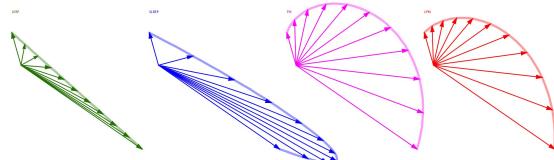


Fig. 19. Interpolations between two vectors: LERP (green), SLERP (blue), PM (magenta), and LPM (red), which is steady and traces a log-spiral,

Replacing LPM by PM for vector interpolation produces the non-TS, yet more “voluptuous” **PM variant** of COTS (Fig. 20).

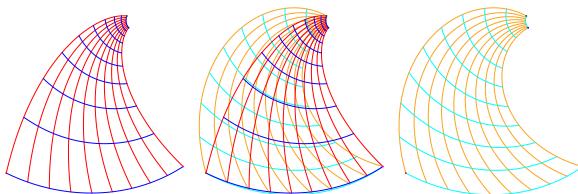


Fig. 20. A COTS map (left), which is based on LPM, and its PM variant (right), which is based on PM and is not TS. Their overlay (center).

### 7.2 Coons Patch from COTS borders

We use a Coons patch [Coons 1967] to produce a map that interpolates the four log-spiral border-edges of the range  $R$  of a COTS map. The Coons construction adds the linear interpolations between pairs of opposite borders and subtract the bilinear interpolation of the corners. Although this variant resembles the COTS map, it is not TS (Fig. 21).

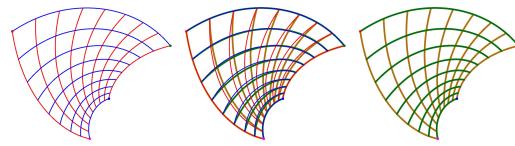


Fig. 21. COTS map (left), Coons patch (right), their overlay (center).

### 7.3 Neville-SAS Interpolation of edge sequences

$\mathcal{U} = \text{SIM}(a, b, d, c)$  defines a SAS (Def. 2.2) that morphs  $ab$  to  $dc$ .  $\mathcal{V} = \text{SIM}(a, d, b, c)$  defines a SAS that morphs  $ad$  to  $bc$  (Fig. 22). A strip of connected SAS maps (Fig. 22-c) that interpolates a sequence of control edges (black) can be smoothed (Fig. 22-d) by replacing LERP by SAS in the Neville construction [Barry and Goldman 1988]. The result may be used to define a non-TS map controlled by a quad-strip. Because the transversal iso-curves are straight, we expect that such a map will be useful for bending long shapes or images and varying their thickness.

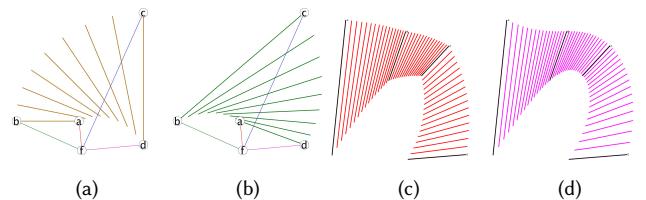


Fig. 22. Two steady rows of  $(n+1)$  edges created using the SAS (Steadily-Animated Similarities) defined by quadruplet  $\{a, b, c, d\}$ : Edges  $\mathcal{U}^{j/n} \cdot ab$  (a) and Edges  $\mathcal{V}^{j/n} \cdot ad$  (b). A strip of SAS (c) that interpolates 4 control edges (black). An interpolating strip that smoothly blends these (d).

### 7.4 Symmetric Log-Spiral (SLS) map

The **Symmetric Log-Spiral (SLS)** map is controlled by 8 points,  $\{a, a', b, b', c, c', d, d'\}$ , which define a quadruplet of labeled edges,  $\{aa', bb', cc', dd'\}$  (Fig. 23). To define it, we start with the bilinear interpolation (Sec. 3.3),  $I(I(a, x, b), y, I(d, x, c))$ , but replace each LERP by the SAS (Sec. 2.3) between the corresponding edges. For example, we replace  $I(a, x, b) = (1-x)a + xb$  by SAS  $\mathcal{U}^x \cdot aa'$  where  $\mathcal{U} = \text{SIM}(a, a', b, b')$ . To make the solution symmetric, we perform the evaluation twice (swapping the  $u$ -first and  $v$ -first options) and average the results, using the mid-course edge of a SAS. Although, in general, SLS is not TS, it provides control continuity, and may be suitable for tweaking a COTS map, because quadruplet  $\{a', b', c', d'\}$  may be set automatically so that the SLS is identical to the COTS map for quadruplet  $\{a, b, c, d\}$  and then adjusted as desired, compromising between Tran-Similarity and additional control of the distortion.

### 7.5 Bézier Patch and Curve of COTS

In Fig. 24, we compare a standard bi-quadratic Bézier patch, which we evaluate using five calls to a bilinear interpolation, to a **bi-quadratic Bézier Patch of COTS**, which we evaluate by replacing

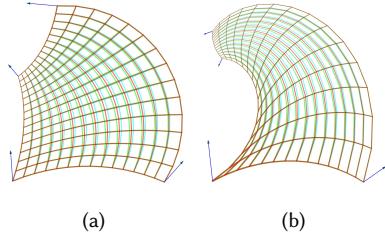


Fig. 23. Examples of a Symmetric Log-Spiral (SLS) map controlled by four corner frames, each defined by an arrow (blue). The two primary ( $u$ -first and  $y$ -first) constructions are shown in green and cyan. Their results are nearly identical. Their average is shown in brown.

the bilinear interpolation, mentioned above, by a COTS map. Although the COTS version affords more expressive power, it may exhibit unexpected undulations and may, for extreme deformations, is more difficult to control.

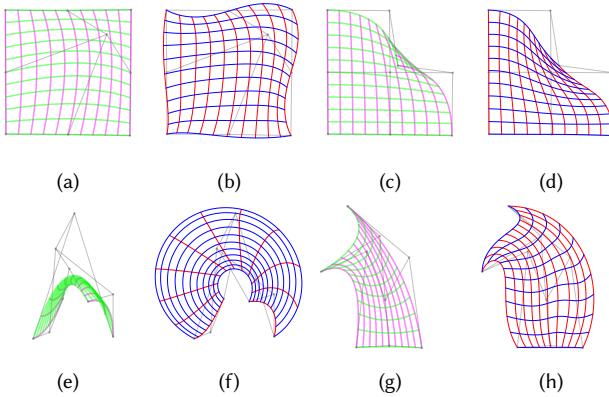


Fig. 24. Quadratic Bézier patch: We show (a,c,e,g) the iso-curves of standard bi-quadratic Bézier patch formulated as a composition of bilinear interpolations and (b,d,f,h), for each of these, the iso-curves of the bi-quadratic Bézier patch of COTS for the same  $3 \times 3$  control-grid (black), which we compute by simply replacing the bilinear interpolation by a COTS map. Notice that the COTS version is more expressive (b,f) and less prompt to fold (d), but may exhibit undesired undulations (h).

In Fig. 25, we compare a mixed-degree Bézier patch with a  $4 \times 2$  control-grid to what we call a **Cubic Bézier Curve of COTS (CBCC)**, which we evaluate using a cascade of 3, then 2, then 1 COTS maps. Again, the COTS version affords more expressive power and does not exhibit unexpected undulations. It is easier to control than the bi-quadratic Bézier patch of COTS. It may offer a valuable alternative to the variant proposed in Sec. 7.3.

### 7.6 Copy-Warp-And-Paste (CWAP) map

We can use two COTS maps to define a **Copy-Warp-And-Paste (CWAP)** map. When set over a source image (see Fig. 26-top), the range  $R_1$  of the first COTS map,  $M_1$ , defines the template,  $T$  (portion of the image), that we want to copy. The range  $R_2$  of the second COTS map,  $M_2$ , (see Fig. 26) defines where  $T$  should be placed and

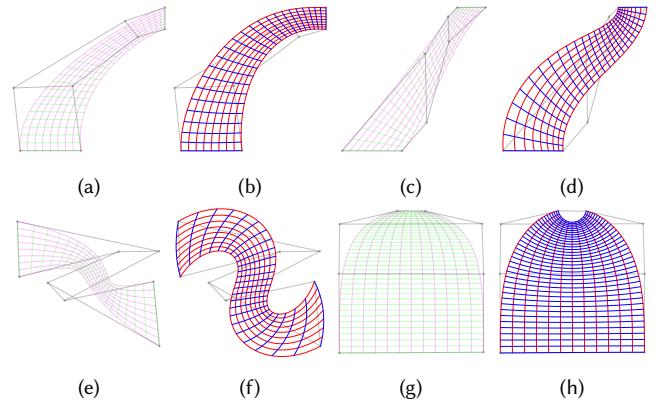


Fig. 25. We show (a,c,e,g) the iso-curves of mixed-degree ( $4 \times 2$ ) Bézier patch and (b,d,f,h), for each of these, the iso-curves of the Cubic Bézier Curve of COTS for the same control-grid (black).

how it should be distorted. The result is  $M_2 \cdot M_1^{-1} \cdot T$ . In general, the composition  $M_2 \cdot M_1^{-1}$  is not TS. Yet, it provides continuity of control (Sec. 6) and a map that is free from distortion artifacts.

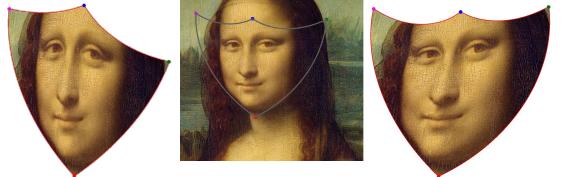


Fig. 26. In a CWAP application, the range  $R_1$  of COTS map  $M_1$  is used (center) to select a portion,  $T$ , of an image. COTS map  $M_2$  controls where  $T$  is pasted and how it is distorted (left and right).

We mention CWAP because it resembles the **Four-Point Interpolant (FPI)** [Lipman et al. 2012] discussed in Sec. 3.4. However, CWAP and FPI have only 16 DoFs, and hence, by themselves, may not offer sufficient flexibility for most image warping applications.

## 8 APPLICATION TO LATTICES

In this section, we explain how using the COTS map for designing planar lattices improves their steadiness and reduces the computational cost of fundamental queries [Hoffmann et al. 2014], such as **PIT** (Point-Inclusion-Testing) and **TAC** (Total Area Computation).

### 8.1 Goal and motivation

To appreciate the strategic importance of modeling steady lattices and of reducing the associated query-costs, consider that advances in additive manufacturing [Gao et al. 2015] allow us to make objects with structural complexity exceeding significantly what can be fabricated using traditional manufacturing. The fine scale of geometric features may be exploited to make objects with unprecedented strength-to-weight ratio or energy absorption [Schaeffler and Carter 2016].

Although numerous efforts are focused on the automatic generation of non-periodic structures, using for example Topology Optimization [Sanders et al. 2017], scaling these solutions to micro- or nano-structures that comprise billions of elements falls far beyond the capacity of current CAD systems. Hence, recent efforts focus on providing efficient tools that support the design, visualization, analysis, and fabrication of procedural models of lattices [Chu et al. 2008; Gupta et al. 2019; Regli et al. 2016], which can be defined concisely and evaluated lazily, instantiating, when needed, only the portion of the lattice in the current Region-of-Interest (RoI).

The challenge of processing extremely complex lattices defined by procedural models is the query-cost. We need solutions for identifying, in constant time (independent of  $\underline{n}$ ), the tiles and the lattice elements that intersect a given RoI and for computing the area, centroid, or mass of a lattice in constant time (without having to instantiate a large number of its elements).

## 8.2 Steady Lattices in 3D

The **Steady lattice**, introduced in [Gupta et al. 2019] is the union of solid beams defined in terms of a three-dimensional **steady tensor** (row of matrices) of **ball-groups**. A **beam** is either a cone-sphere [Max 1990] (the convex hull of two spheres) or a sculptured-beam with non-linearly varying radius (such as a quadric of revolution [Gupta et al. 2018]), and connects two balls of the same or of adjacent ball-groups.

The 3D **Steady Lattice** proposed in [Gupta et al. 2019] and exploited in [Kurzeja and Rossignac 2019] were invented to address the conflicting challenges of supporting warped lattices with perfect balls and straight beams and of reducing the complexity of **PIT** (Point-Inclusion-Testing) and of **IPC** (Integral Property Calculations), which include Total-Volume-Computation, Center-of-Mass-Computation, and **TAC** (Total-Area-Computation).

Their tensor of ball-groups is defined in terms of 4 user-controlled similarity frames,  $\mathcal{A}, \mathcal{B}, \mathcal{C}$ , and  $\mathcal{D}$ , (each defined by an orthonormal coordinate system and a uniform scaling), from which one defines three similarities:  $\mathcal{U}$  that takes  $\mathcal{A}$  to  $\mathcal{B}$ ,  $\mathcal{V}$  that takes  $\mathcal{A}$  to  $\mathcal{C}$ , and  $\mathcal{W}$  that takes  $\mathcal{A}$  to  $\mathcal{D}$ . These define a steady tensor of local frames,  $\mathcal{F}_{i,j,k} = \mathcal{W}^{k/\underline{n}} \cdot \mathcal{V}^{j/\underline{n}} \cdot \mathcal{U}^{i/\underline{n}} \cdot \mathcal{A}$ , which defines a steady tensor of ball-groups,  $\mathcal{G}_{i,j,k} = \mathcal{F}_{i,j,k} \cdot \mathcal{G}_{0,0,0}$ , where  $\mathcal{G}_{0,0,0}$  is a user-controlled **template** ball-group. Each ball-group belongs to three rows:  $\mathcal{G}_{*,j,k}$ ,  $\mathcal{G}_{i,*,k}$ , and  $\mathcal{G}_{i,j,*}$ . Each row is steady. Hence, PIT and TAC against any one of these rows of ball-groups takes constant time. But performing PIT or TAC against a whole lattice that also includes the beams, which each connects a pair of these balls, have each complexity of  $O(\underline{n}^2)$  [Gupta et al. 2019; Kurzeja and Rossignac 2019]. Below, we present the planar version of such Steady Lattices in terms of a steady matrix (Def. 2.5) of **disk-groups** and of planar **bars**. We analyze and explain the associated PIT and TAC costs. We compare them to corresponding costs on regular lattices and on various warps of these, including by a COTS map.

## 8.3 Connectivity of a planar lattice

A planar **lattice** is defined by an array,  $\mathbb{D}[d,m,n]$ , of disks (Fig. 27a) and is the union of an array,  $\mathbb{B}[b,m,n]$ , of bars (Fig. 27b). The **disks** form a matrix of  $m \times n$  **disk-groups**, each containing  $d$  disks. The

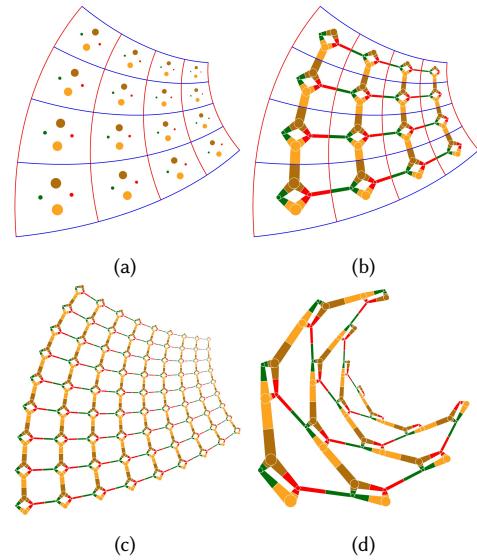


Fig. 27. Clean COTS lattice with a  $4 \times 4$  matrix of disk-groups (a), 6 matrices of bars, 4 matrices of hubs (b). A  $9 \times 9$  version (c) and a distorted version (d).

**bars** form  $b$  matrices of  $m \times n$  bars each. Bar  $\mathbb{B}[b,m,n]$  is the convex hull of the union of disks  $\mathbb{D}[s[d],m,n]$  and  $\mathbb{D}[e[d],m+i[d],n+j[d]]$ . The **offsets**,  $i[d]$  and  $j[d]$ , are index increments defining whether the bar that starts at disk  $\mathbb{D}[s[d],m,n]$  connects to disk  $e[d]$  in the same group (when  $i[d]=j[d]=0$ ) or in a neighbor-groups (when  $i[d]$  or  $j[d]$  is not 0). We split each bar into two halves (by a line orthogonal to its axis and equidistant from its two disks). To each disk,  $\mathbb{D}[d,m,n]$ , corresponds **hub**,  $\mathbb{H}[d,m,n]$ , which is the union of those halves of all the bars incident on  $\mathbb{D}[d,m,n]$  that contain it. We assume that the lattice is **clean**, i.e., that the interiors of the hubs are disjoint. For a given  $d$ , we split the matrix of hubs,  $\{\mathbb{H}[d, m, n]\}$ , into 9 **hub-types**: 4 types comprise a single corner-hub (such as  $\mathbb{H}[d, 0, 0]$ ), 4 types comprise a border-row of hubs (such as  $\{\mathbb{H}[d, *, 0]\}$ ) (without the corner-hubs), and the last type comprises the remaining  $(m-2) \times (n-2)$  **interior-hubs**. When discussing asymptotic complexity, we assume that  $n = m$  and that  $d$  and  $b$  are small constants. Hence, we express time-complexity in terms of  $n$ . In Fig. 27a, we show a clean lattice with a tile-count  $\underline{n}$  of 4. It has 4 disks (green, red, orange, brown) in each group. It has 6 matrices of bars (Fig. 27b): 4 short intra-group bars and 2 longer inter-group bars (red-to-green one for  $(i[d], j[d])=(1,0)$  and brown-to-orange one for  $(i[d], j[d])=(0,1)$ ). The lattice has 4 matrices of hubs (green, red, brown, and orange), each with 4 interior-hubs.

## 8.4 Warps of regular lattices

An irregular lattice may be defined as the warp of a regular one [Elber 2016]. A **translation lattice** is a steady lattice of which the disk-groups, each, form a translation matrix (Def. 2.8). PIT and TAC queries may be performed in constant time on translation lattices [Pasko et al. 2011]. A **regular lattice** is a translation lattice, for which the disk-groups form a regular matrix (Def. 2.9). **Free-form Deformations** (FFD) [Llamas et al. 2003; Sederberg and Parry 1986] of regular lattices provide a large design space. However, they distort

the bars and disks (which, for most applications, must be rectified to perfect disks and to bars with straight edges), they may require iterative root-finders to compute map-inversions [Machchhar and Elber 2018; Patrikalakis and Maekawa 2009] and they yield non-steady matrices of inner-hubs, which leads to quadratic-cost,  $O(\underline{n}^2)$ , algorithms for TAC and other IPC queries. One could also use one of the non-TS maps discussed in Sec. 3. Several of these have closed-form expressions for computing the inverse of the map, which could be used to simplify the support of PIT queries at constant cost. But, due to their lack of Tran-Similarity, the hubs of any row will, in general, not be similar to each other, which again, leads to  $O(\underline{n}^2)$  algorithms for TAC.

### 8.5 Five-point-operated Planar Steady Lattice (PSL)

To define a **Planar Steady Lattice (PSL)**, i.e., a planar version of the 3D Steady Lattice (Sec. 8.2), we need only 3 similarity frames ( $\mathcal{A}$ ,  $\mathcal{B}$ , and  $\mathcal{D}$ ). Each one may be defined by a planar similarity and represented by two control points (Sec. 2.2). Hence, a matrix of steady frames may be defined by six control points:  $a$  and  $a'$  for  $\mathcal{A}$ ,  $b$  and  $b'$  for  $\mathcal{B}$ , and  $d$  and  $d'$  for  $\mathcal{D}$ . From these, we define similarity  $\mathcal{U}$  that takes  $\mathcal{A}$  to  $\mathcal{D}$  and similarity  $\mathcal{V}$  that takes  $\mathcal{A}$  to  $\mathcal{B}$ . From these, we define a steady matrix of local frames:  $\mathcal{F}_{i,j} = \mathcal{V}^{j/\underline{n}} \cdot \mathcal{U}^{i/\underline{n}} \cdot \mathcal{A}$ .

In fact, we need only 5 control-points to define a PSL. In Fig. 28, we compare a COTS lattice (left) defined (Sec. 8.6) by a COTS map,  $\mathcal{M}$ , to a PSL (right), for which one may use the additional control point,  $e$ , which is the image of  $(0.5, 0)$ , to modify  $\mathcal{U}$ . The hub matrices of that lattice are not TS, which leads to  $O(n)$  cost for TAC.

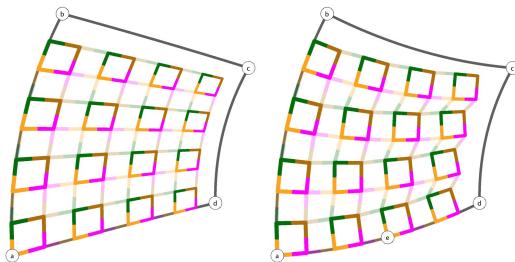


Fig. 28. COTS lattice (left) with  $\mathcal{U} = \text{SIM}(a, b, d, c)$  and  $\mathcal{V} = \text{SIM}(a, d, b, c)$ , for which all intra-group quads (made of a cycle of bars in solid-colors) and all inter-group quads (made of a cycle of semi-transparent bars) are similar to each other. A PSL (right) with  $\mathcal{U} = \text{SIM}(a, e, e, d)^2$  and  $\mathcal{V} = \text{SIM}(a, d, b, c)$ , for which all intra-group quads are similar to each other, but the inter-group quads are not and the m-rows are not steady.

### 8.6 Lattice from a COTS map of disks

The COTS Lattices may be viewed as a constrained subset of PSLs (Sec. 8.5), for which  $e = \mathcal{M}(0.5, 0)$ . Setting  $e = \mathcal{M}(0.5, 0)$  restores the original  $\mathcal{U}$  and yields a COTS lattice (Fig 28-left) for which TAC queries have  $O(1)$  cost. The lattice shown in Fig. 27b is a COTS lattice. For clarity, we placed each disk-group inside a tile of the COTS map and drew (red and blue) the tile borders (Fig. 27a). In Fig. 27c, we show a higher density version of this lattice obtained by simply increasing the repetition counts ( $\underline{m} = \underline{n} = 9$ ). Note that, when the user adjusts  $\underline{n}$ , we automatically adjust the disk radii and

positions. In Fig. 27d we show a more warped, although still clean, version obtained by moving the control corners of the COTS map.

For a COTS lattice, the PIT and TAC queries may each be performed in constant time. For TAC, we compute the area of  $\mathbb{H}[d, 1, 1]$  and cascade two summations of geometric series (Sec. 5.4). For PIT, we could precompute the preimage  $p$  of the query point  $q$  using the closed-form inversion of the COTS map (Sec. 5.2). However, the preimage of a COTS lattice is the union of Translation-Steady matrices of distorted bars. Hence, we do not classify the preimage of query point  $q$  against these distorted bars. Instead, we use inversion to identify the tile that contains  $q$  (Sec. 5.2) and then classify  $q$  against the original bars of the COTS lattice that interfere with the identified tile. This novel solution extends the idea proposed in [Kurzeja and Rossignac 2019] from a 3D steady row to a planar COTS Lattice-Slab.

## 9 CONCLUDING SUMMARY

The concept of Tran-Similarity (TS), introduced here, is novel and important. If a map,  $\mathcal{M}$ , is TS, then, for each  $\vec{v}$ , there exists a similarity  $\mathcal{S}$ , such that  $\mathcal{M}(X + t\vec{v}) = \mathcal{S}^t \cdot \mathcal{M}(X)$  for any  $t$ . A TS map deforms a regular grid of tiles (or shapes) into a TS matrix in which all tiles (or all shapes) are related to their neighbors by constant similarities. This property has useful benefits. For example, the computation of the total surface area of a  $1000 \times 1000$ , non-TS matrix of disjoint shapes typically requires performing a million different shape-area computations. When the matrix is the image of a regular matrix by a TS map, this cost is reduced to the computation of a single shape-area. This nearly 1,000,000-to-1 cost reduction is especially important when the warped matrix is used to design and optimize the gradation of the micro-structure of a mechanical part of which the total mass is constrained.

Much of prior art on planar maps was focused on conformal or quasi-conformal maps or concerned with the minimization of a measure of conformal distortion. We posit that striving to minimize distortion may be the wrong objective for artistic, engineering, or architectural applications, in which a planar warp is used precisely to distort an image or a regular pattern. We propose that, for such applications, the goodness of a map be measured by the evenness of distortion. According to this new measure, a TS map is optimal, since it deforms all tiles or all shapes of a regular matrix identically.

When a map has  $d$  degrees of freedom, we say that it has  $d-4$  degrees of distortion (DoD), because 4 degrees may be allocated to control the overall position, orientation, and size. A Möbius map has 2 DoD. It is conformal, but not TS. An affine map is TS, but also has only 2 DoD, which is not sufficient to model warps and gradations of structures. We are not aware of any previously proposed TS map that has more than 2 DoD.

Of particular interest here is the class of Corner-Operated (CO) maps, which are fully controlled (in a symmetric and affine invariant way) by the images of the four corners of a unit square. These have 4 DoD. For example, the restricted version of the recently proposed FPI map, which is the composition of an affine and a Möbius map, is CO and minimizes conformal distortion, but is not TS.

The proposed COTS map is CO, resembles the restricted FPI, and is TS. It has an elegant formulation:  $\mathcal{M}(x, y) = \mathcal{V}^y \cdot \mathcal{U}^x \cdot a$ , where

$\mathcal{U}$  and  $\mathcal{V}$  are proper similarities and  $a$  is a control corners. We also show that it can be expressed as the product of a similarity, a log-polar map, and a linear transform. We provide simple, closed-form expressions for computing  $\mathcal{U}$  and  $\mathcal{V}$  and for evaluating  $M(x, y)$  and its inverse.  $M$  is TS because  $\mathcal{U}$  and  $\mathcal{V}$  commute, which means that they have the same fixed point,  $f$ , which we call the Four-Point Similarity-Center. We provide a simple, closed-form expression for  $f$  in terms of the four control corners.

We discuss the practical application of the COTS map to the design, analysis, and optimization of warped and graded, planar lattices. We explain its benefits over the recently proposed Steady Lattice formulations. These include steady alignment of inter- and intra-group beams and a reduction, from  $O(n)$  to  $O(1)$ , of the computational costs of two key queries.

The main contributions of this paper are the formulation and computation of the COTS map, the exploitation of its TS property to accelerate computations, and the suggestion that distortion evenness may be an important measure of the quality of a planar warp. We also believe that the COTS map may have broader benefits for defining non-TS maps with higher DoD and for designing aesthetic animations of simple warps. To inspire further research in this direction, we show several warp constructions that combine COTS maps, as, for example, a Bézier patch combines bilinear interpolations.

## 10 ACKNOWLEDGEMENTS

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. The author thanks the reviewers for their suggestions, Ashish Gupta and Kelsey Kurzeja for their comments, and Dr. Alex Diaz for suggesting possible links between COTS lattices and Michell trusses.

## REFERENCES

- Marc Alexa, Daniel Cohen-Or, and David Levin. 2000. As-rigid-as-possible Shape Interpolation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 157–164. <https://doi.org/10.1145/344779.344859>
- Phillip J. Barry and Ronald N. Goldman. 1988. A Recursive Evaluation Algorithm for a Class of Catmull-Rom Splines. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 199–204. <https://doi.org/10.1145/378456.378511>
- Guy Bouchitté, Wilfrid Gangbo, and Pierre Seppecher. 2008. Michell Trusses and Lines of Principal Action. *Mathematical Models and Methods in Applied Sciences* 18, 9 (Sept. 2008), 1571–1603. <https://hal.archives-ouvertes.fr/hal-00528973>
- Renjie Chen and Craig Gotsman. 2016. Complex Transfinite Barycentric Mappings with Similarity Kernels. In *Proceedings of the Symposium on Geometry Processing (SGP '16)*. Eurographics Association, Goslar Germany, Germany, 41–53. <https://doi.org/10.1111/cgf.12962>
- Renjie Chen, Ofir Weber, Daniel Keren, and Mirela Ben-Chen. 2013. Planar Shape Interpolation with Bounded Distortion. *ACM Trans. Graph.* 32, 4, Article 108 (July 2013), 12 pages. <https://doi.org/10.1145/2461912.2461983>
- Albert Chern, Ulrich Pinkall, and Peter Schröder. 2015. Close-to-conformal Deformations of Volumes. *ACM Trans. Graph.* 34, 4, Article 56 (July 2015), 13 pages. <https://doi.org/10.1145/2766916>
- Edward Chien, Renjie Chen, and Ofir Weber. 2016. Bounded Distortion Harmonic Shape Interpolation. *ACM Trans. Graph.* 35, 4, Article 105 (July 2016), 15 pages. <https://doi.org/10.1145/2897824.2925926>
- Chen Chu, Greg Graff, and David W. Rosen. 2008. Design for additive manufacturing of cellular structures. *Computer-Aided Design and Applications* 5, 5 (2008), 686–696.
- S. A. Coons. 1967. *Surfaces for computer-aided design of space forms*. Technical Report. Cambridge, MA, USA.
- Gershon Elber. 2016. Precise construction of micro-structures and porous geometry via functional composition. In *International Conference on Mathematical Methods for Curves and Surfaces*. Springer, 108–125.
- Michael S. Floater. 2003. Mean Value Coordinates. *Comput. Aided Geom. Des.* 20, 1 (March 2003), 19–27. [https://doi.org/10.1016/S0167-8396\(02\)00002-5](https://doi.org/10.1016/S0167-8396(02)00002-5)
- Wei Guo, Yunbo Zhang, Devarajan Ramanujan, Karthik Ramani, Yong Chen, Christopher B. Williams, Charlie C.L. Wang, Yung C. Shin, Song Zhang, and Pablo D. Zavattieri. 2015. The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design* 69 (2015), 65 – 89. <https://doi.org/10.1016/j.cad.2015.04.001>
- Ashish Gupta, George Allen, and Jarek Rossignac. 2018. QUADOR: QUADric-Of-Revolution beams for lattices. *Computer-Aided Design* 102 (2018), 160–170.
- Ashish Gupta, Kelsey Kurzeja, Jarek Rossignac, George Allen, Pranav Srinivas Kumar, and Suraj Musuvathy. 2019. Programmed-Lattice Editor and accelerated processing of parametric program-representations of steady lattices. *Computer-Aided Design* 113 (2019), 35 – 47. <https://doi.org/10.1016/j.cad.2019.04.001>
- W.S. Hemp. 1973. *Optimum Structures*. Oxford University Press, Oxford.
- Christoph Hoffmann, Vadim Shapiro, and Vijay Srinivasan. 2014. Geometric interoperability via queries. *Computer-Aided Design* 46 (2014), 148–159.
- Kai Hormann and Michael S. Floater. 2006. Mean Value Coordinates for Arbitrary Planar Polygons. *ACM Trans. Graph.* 25, 4 (Oct. 2006), 1424–1441. <https://doi.org/10.1145/1183287.1183295>
- Takeo Igarashi, Tomer Moscovich, and John F. Hughes. 2005. As-rigid-as-possible Shape Manipulation. In *ACM SIGGRAPH 2005 Papers (SIGGRAPH '05)*. ACM, New York, NY, USA, 1134–1141. <https://doi.org/10.1145/1186822.1073323>
- Miao Jin, Wei Zeng, Feng Luo, and Xianfeng Gu. 2009. Computing Teichmüller Shape Space. *IEEE Transactions on Visualization and Computer Graphics* 15, 3 (May 2009), 504–517. <https://doi.org/10.1109/TVCG.2008.103>
- Tao Ju, Scott Schaefer, and Joe Warren. 2005. Mean Value Coordinates for Closed Triangular Meshes. *ACM Trans. Graph.* 24, 3 (July 2005), 561–566. <https://doi.org/10.1145/1073204.1073229>
- Kelsey Kurzeja and Jarek Rossignac. 2019. RangeFinder: Accelerating ball-interference queries against steady lattices. *Computer-Aided Design* 112 (2019), 14 – 22. <https://doi.org/10.1016/j.cad.2019.03.002>
- F. C. Langbein, C. H. Gao, B. I. Mills, A. D. Marshall, and R. R. Martin. 2004. Topological and Geometric Beautification of Reverse Engineered Geometric Models. In *Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications (SM '04)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 255–260. <http://dl.acm.org.prx.library.gatech.edu/citation.cfm?id=1217875.1217915>
- Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Trans. Graph.* 21, 3 (July 2002), 362–371. <https://doi.org/10.1145/566654.566590>
- Yaron Lipman, Vladimir G. Kim, and Thomas A. Funkhouser. 2012. Simple formulas for quasiconformal plane deformations. *ACM Trans. Graph.* 31, 5 (2012), 124:1–124:13. <https://doi.org/10.1145/2231816.2231822>
- Xingchen Liu and Vadim Shapiro. 2018. Multiscale shape-material modeling by composition. *Computer-Aided Design* 102 (2018), 194–203.
- Yanxi Liu, Robert T. Collins, and Yanghai Tsin. 2004. A Computational Model for Periodic Pattern Perception Based on Frieze and Wallpaper Groups. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 3 (March 2004), 354–371. <https://doi.org/10.1109/TPAMI.2004.1262332>
- Ignacio Llamas, Byungmoon Kim, Joshua Gargus, Jarek Rossignac, and Chris D. Shaw. 2003. Twister: A Space-warp Operator for the Two-handed Editing of 3D Shapes. *ACM Trans. Graph.* 22, 3 (July 2003), 663–668. <https://doi.org/10.1145/882262.882323>
- Jinesh Machchhar and Gershon Elber. 2018. A note on zeros of univariate scalar Bernstein polynomials. *Computer Aided Geometric Design* 66 (2018), 75 – 79. <https://doi.org/10.1016/j.cagd.2018.09.001>
- Nelson Max. 1990. Cone-spheres. *SIGGRAPH Comput. Graph.* 24, 4 (Sept. 1990), 59–62. <https://doi.org/10.1145/97880.97885>
- Anthony G. M. Michell. 1904. The limits of economy of material in frame-structures. *Phil. Mag., Ser* 8, 47 (1904), 589–597.
- Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. 2013. Symmetry in 3D Geometry: Extraction and Applications. *Comput. Graph. Forum* 32, 6 (Sept. 2013), 1–23. <https://doi.org/10.1111/cgf.12010>
- Xianshu Nian and Falai Chen. 2016. Planar Shape Interpolation Based On Teichmüller Mapping. *Comput. Graph. Forum* 35, 7 (Oct. 2016), 43–56. <https://doi.org/10.1111/cgf.13002>
- Alexander A. Pasko, Oleg Fryazinov, Turlif Vilbrandt, Pierre-Alain Fayolle, and Valery Adzhiev. 2011. Procedural function-based modelling of volumetric microstructures. *Graphical Models* 73 (2011), 165–181.
- Nicholas M Patrakakis and Takashi Maekawa. 2009. *Shape interrogation for computer aided design and manufacturing*. Springer Science & Business Media.
- D. Perny, M. Gangnet, and Ph. Coueignoux. 1982. Perspective Mapping of Planar Textures. *SIGGRAPH Comput. Graph.* 16, 1 (May 1982), 70–100. <https://doi.org/10.1145/988468.988470>

- Roi Poranne and Yaron Lipman. 2014. Provably Good Planar Mappings. *ACM Trans. Graph.* 33, 4, Article 76 (July 2014), 11 pages. <https://doi.org/10.1145/2601097.2601123>
- William Regli, Jarek Rossignac, Vadim Shapiro, and Vijay Srinivasan. 2016. The new frontiers in computational modeling of material structures. *Computer-Aided Design* 77 (2016), 73–85.
- Jarek Rossignac and Álvar Vinacua. 2011. Steady affine motions and morphs. *ACM Transactions on Graphics (TOG)* 30, 5 (2011), 116.
- Emily D Sanders, Adeildo S Ramos, and Glauco H Paulino. 2017. A maximum filter for the ground structure method: An optimization tool to harness multiple structural designs. *Engineering Structures* 151 (2017), 235–252.
- Tobias A Schaedler and William B Carter. 2016. Architected cellular materials. *Annual Review of Materials Research* 46 (2016), 187–210.
- Scott Schaefer, Travis McPhail, and Joe Warren. 2006. Image Deformation Using Moving Least Squares. *ACM Trans. Graph.* 25, 3 (July 2006), 533–540. <https://doi.org/10.1145/1141911.1141920>
- Teseo Schneider and Kai Hormann. 2015. Smooth bijective maps between arbitrary planar polygons. *Computer Aided Geometric Design* 35–36 (2015), 243 – 254. <https://doi.org/10.1016/j.cagd.2015.03.010> Geometric Modeling and Processing 2015.
- Thomas W. Sederberg and Scott R. Parry. 1986. Free-form Deformation of Solid Geometric Models. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 151–160. <https://doi.org/10.1145/15886.15903>
- Aviv Segal and Mirela Ben-Chen. 2016. Iterative Closest Conformal Maps Between Planar Domains. In *Proceedings of the Symposium on Geometry Processing (SGP '16)*. Eurographics Association, Goslar Germany, Germany, 33–40. <https://doi.org/10.1111/cgf.12961>
- Ken Shoemake. 1985. Animating Rotation with Quaternion Curves. *SIGGRAPH Comput. Graph.* 19, 3 (July 1985), 245–254. <https://doi.org/10.1145/325165.325242>
- Robert E. Skelton and Mauricio C. de Oliveira. 2010. Optimal tensegrity structures in bending: The discrete Michell truss. *Journal of the Franklin Institute* 347, 1 (2010), 257 – 283. <https://doi.org/10.1016/j.jfranklin.2009.10.009> Dynamics and Control.
- Robert Tilove. 1980. Set Membership Classification: A Unified Approach to Geometric Intersection Problems. *IEEE Trans. Comput. C-29*, 10 (Oct 1980), 874–883. <https://doi.org/10.1109/TC.1980.1675470>
- Amir Vaxman, Christian Müller, and Ofir Weber. 2015. Conformal Mesh Deformations with Möbius Transformations. *ACM Trans. Graph.* 34, 4, Article 55 (July 2015), 11 pages. <https://doi.org/10.1145/2766915>
- Hongqing Wang and David W Rosen. 2002. Parametric modeling method for truss structures. In *ASME Computers and Information in Engineering Conference*.
- Ofir Weber, Mirela Ben-Chen, and Craig Gotsman. 2009. Complex Barycentric Coordinates with Applications to Planar Shape Deformation. *Computer Graphics Forum* (2009). <https://doi.org/10.1111/j.1467-8659.2009.01399.x>
- Ofir Weber and Craig Gotsman. 2010. Controllable Conformal Maps for Shape Deformation and Interpolation. *ACM Trans. Graph.* 29, 4, Article 78 (July 2010), 11 pages. <https://doi.org/10.1145/1778765.1778815>
- Ofir Weber, Ashish Myles, and Denis Zorin. 2012. Computing Extremal Quasiconformal Maps. *Comput. Graph. Forum* 31, 5 (Aug. 2012), 1679–1689. <https://doi.org/10.1111/j.1467-8659.2012.03173.x>
- Yufeng Zhu, Jovan Popović, Robert Bridson, and Danny M. Kaufman. 2017. Planar Interpolation with Extreme Deformation, Topology Change and Dynamics. *ACM Trans. Graph.* 36, 6, Article 213 (Nov. 2017), 15 pages. <https://doi.org/10.1145/3130820>

## A PROOFS OF THEOREMS

### A.1 Theorem 2.6

**PROOF.**  $X_{r,n} = \mathcal{J}^n \cdot (\mathcal{I}^r \cdot X)$  implies that row  $X_{r,*}$  is steady and has  $\mathcal{I}^r \cdot X$  for template.  $X_{m,r} = (\mathcal{J}^r \cdot \mathcal{I}^m \cdot \mathcal{J}^{-r}) \cdot (\mathcal{J}^r \cdot X)$  can be written as  $X_{m,r} = \mathcal{S}^m \cdot (\mathcal{J}^r \cdot X)$  with  $\mathcal{S}^m = \mathcal{J}^r \cdot \mathcal{I}^m \cdot \mathcal{J}^{-r}$ , and thus  $\mathcal{S}^m = (\mathcal{J}^r \cdot \mathcal{I} \cdot \mathcal{J}^{-r})^m$  with  $\mathcal{S} = (\mathcal{J}^r \cdot \mathcal{I} \cdot \mathcal{J}^{-r})$ . Hence, row  $X_{*,r}$  is also steady.  $\square$

### A.2 Theorem 4.1

**PROOF.** Use Equation 3 to produce the complex forms of the two constraints that define  $\mathcal{U}$ :

(1) Constraint  $\mathcal{U} \cdot a = d$  implies

$$d - f = u(a - f). \quad (18)$$

(2) Constraint  $\mathcal{U} \cdot b = c$  implies

$$c - f = u(b - f). \quad (19)$$

Dividing Eq. 18 by Eq. 19 eliminates  $u$ :

$$\frac{d - f}{c - f} = \frac{a - f}{b - f}, \quad (20)$$

which, after simplification, yields

$$f = \frac{ac - bd}{a - b + c - d}, \quad (21)$$

Substituting Eq. 21 into Eq. 18 and simplifying yields

$$u = \frac{c - d}{b - a}. \quad (22)$$

$\square$

### A.3 Theorem 4.8

**PROOF.** Let us show that  $\mathcal{M}$  satisfies Eq. 6 in Def. 4.3. Let  $\langle x', y' \rangle$  denote the Cartesian components of  $\vec{v}$ . From Eq. 9, the complex form of  $\mathcal{M}(x + tx', y + ty')$  is

$$f + u^{x+tx'} v^{y+ty'} (a - f), \quad (23)$$

which yields

$$f + (u^{x'} v^{y'})^t (u^x v^y (a - f)), \quad (24)$$

and therefore

$$f + (u^{x'} v^{y'})^t ((f + u^x v^y (a - f)) - f), \quad (25)$$

which is

$$f + (u^{x'} v^{y'})^t (\mathcal{M}(x, y) - f). \quad (26)$$

$\square$

### A.4 Theorem 4.9

The complex form of  $q = \mathcal{M}(x, y)$  is

$$q = f + u^x v^y (a - f), \quad (27)$$

which can be written as

$$q - f = u^x v^y (a - f), \quad (28)$$

and, by taking the log, as

$$x \ln u + y \ln v = \ln z, \quad (29)$$

where

$$z = \frac{q - f}{a - f}. \quad (30)$$

Rewriting Eq. 29 as

$$x(\ln \lambda_u + i\alpha_u) + y(\ln \lambda_v + i\alpha_v) = (\ln \lambda_z + i\alpha_z) \quad (31)$$

and equating the real and imaginary parts, produces a system of two linear equations.

### A.5 Theorem 3.2

**PROOF.**  $\mathcal{L}$  maps  $(x, y)$  onto  $(\lambda, \alpha)$  with  $\lambda = x \ln \lambda_u + y \ln \lambda_v$  and  $\alpha = x\alpha_u + y\alpha_v$ .  $\mathcal{P}$  maps  $(\lambda, \alpha)$  onto complex number  $z = e^{\lambda+i\alpha}$ . Finally,  $\mathcal{S}_{f,a}$  maps  $(0, 0)$  to  $f$  and  $(1, 0)$  onto  $a$ . Hence, their composition,  $\mathcal{S}_{f,a} \cdot \mathcal{P} \cdot \mathcal{L}$ , maps  $(x, y)$  onto  $\mathcal{M}(x, y) = f + e^{\lambda} \vec{f} \cdot \vec{a}^\circ(\alpha) = f + \lambda_u^x \lambda_v^y \vec{f} \cdot \vec{a}^\circ(x\alpha_u + y\alpha_v)$  (matching Eq. 12).  $\square$

Received July 2018