Computers & Graphics (2022)

Contents lists available at ScienceDirect

Computers & Graphics

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

Vector graphics on surfaces using straightedge and compass constructions

Claudio Mancinelli[a,*], Enrico Puppo[a]

[a]*DIBRIS - University of Genoa, Italy*

ARTICLE INFO

Article history:
Received May 3, 2022

Keywords: Geometry Processing, Non-Euclidean Geometry, Geometric Primitives on Surfaces, Graphics systems and Interfaces

ABSTRACT

We port classical straightedge and compass constructions to manifold surfaces under the geodesic metric. We propose two complementary approaches: one working on the tangent plane; and another working directly on the surface. In both cases, many constructions lack some of the geometric properties they have in the Euclidean case. We devise alternative constructions that guarantee at least a subset of such properties. We integrate our constructions in the context of a prototype system supporting the interactive drawing of primitives of vector graphics.

1. Introduction

The ancient Greek mathematicians developed a set of geometric techniques, which go under the name of *straightedge and compass constructions*, to draw planar geometric figures. Such constructions do not require taking any explicit measure, they are granted by Euclid's first three postulates, and are based on two idealized tools: the straightedge, which can extend indefinitely the straight-line through any pair of points; and the compass, which can trace circles with its needle and pencil points at any two points in the plane. Besides, all intersections between straight lines and circles drawn with such tools can be found.

In the Euclidean setting, the straightedge and compass constructions can be substituted with simpler closed form solutions, though, as it is customary in 2D drawing systems. However, when addressing similar operations on a surface, one must rely on the computation of distance fields and geodesic lines. Such building blocks are indeed similar in nature to those available in the straightedge and compass framework. In the context of our effort to bring vector graphics to surface domains [1, 2], we thus investigate how to port such constructions to the manifold setting. This paper extends a previous preliminary study [3].

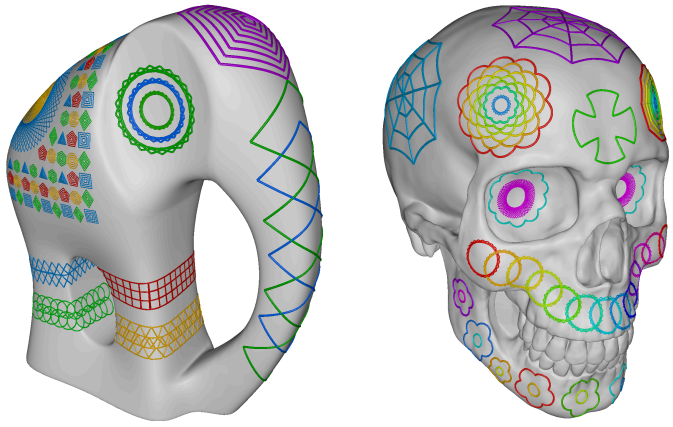


Fig. 1. Examples of drawings obtained interactively with our prototype system on two meshes, each consisting of 1M triangles.

We address the problem with two complementary approaches. The first approach performs constructions in a tangent plane and then maps the result to the surface. The second approach extends the concepts of straightedge and compass to the geodesic metric and works directly on the surface.

Euclidean constructions rely on properties that no longer hold under the geodesic metric, due to the intrinsic curvature of the surface. Because of that, both approaches fail in producing results that preserve *all* properties of their Euclidean counter-

*Corresponding author:
e-mail: claudio.mancinelli@dibris.unige.it (Claudio Mancinelli)

part. In fact, even the topological properties of straight lines and circles do not hold on a surface without additional conditions: geodesic lines may self-intersect or mutually intersect multiple times; and a generic isoline of the distance field is not even guaranteed to be homeomorphic to a circle.

In order to address the topological limitations, we constrain our constructions to occur on sufficiently local subsets of the surface domain. Concerning the metric aspects, the first approach suffers from *geodesic distortion*, which is caused by the curvature when mapping Euclidean geometries from the tangent plane to the surface domain. On the contrary, geodesic lines and circles are well-behaved while working directly on the surface, as long as they the are "small enough". This fact increases our leeway in imposing some local properties.

We integrate all our constructions in a prototype system that supports interactive drawing with geodesic polygons and circles. We also support affine transformations of primitives and all the usual editing operations, such as copy, paste, and delete. We achieve real time interaction on meshes consisting of up to a few million triangles. This is made possible thanks to efficient algorithms to compute geodesic distances and shortest paths.

2. Related work

Intrinsic Geometry of Surfaces. The straightedge and compass constructions rely on basic theorems of the Euclidean geometry that relate lengths and angles. When trying to define similar relations on a surface, curvature must be taken into account. This subject was thoroughly investigated in the classical theory of intrinsic geometry of surfaces. See the books by Cheeger and Ebin [4] and by Chavel [5] for comprehensive accounts. Referring just to the cases addressed in this paper, a corollary of the Gauss-Bonnet theorem relates the internal angles of a geodesic polygon to the curvature of the region it encloses. Such result explains the challenge in addressing constructions that require geodesic lines of given lengths *and* forming given angles. See, e.g., the isosceles triangle in Sec. 5.4.

Alexandrov investigated thoroughly the relations between quantities measured on a surface with their counterpart on surfaces with constant curvature (a.k.a. CAT – Cartan-Alexandrov-Topogonov – spaces) [6]. In a nutshell, geodesic lines, which are cast from a common source along different directions, tend to converge if the curvature of the space is positive, and to diverge if it is negative. Based upon these facts, many comparison theorems involving Alexandrov and CAT spaces have been proposed in the literature. See Alexander et al. [7] for a recent account on this subject; interestingly enough, the title of the chapter addressing geodesic triangles is *The ghost of Euclid*.

Vector Graphics. Vector graphics in 2D is a consolidated subject, supported in many systems and tools at industrial level [8, 9, 10, 11, 12, 13]. Until recently, vector graphics on surfaces under the geodesic metric was considered too computationally expensive to be supported. Traditional methods to decorate a surface resort to parametrization and mapping, but this approach is prone to seams and distortion, as discussed by Nazzaro et al. [2] and Yuksel et al. [14]. The literature concerning tools for geodesic computations is vast, though, and has been recently surveyed by Crane et al. [15]. Some recent contributions demonstrated that such technology is mature enough to support interactive editing directly on surfaces [1, 2, 16]. Most methods we rely on for our constructions are taken from such works and summarized in Sec. 7.

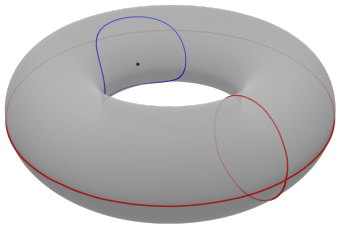


Fig. 2. The cut locus (red) of a point p (black dot) on a torus. The convex ball centered at p (blue) cannot extend further, since otherwise it would contain pairs of points connected with a shortest path crossing the outer equator, hence not entirely contained in such ball. Details about the implementation of the cut locus and convex balls are given in Section 7.

3. Preliminaries

The Geodesic Metric on Surfaces. Here, we provide just some basic definitions, referring to any book of differential geometry, e.g. Chavel [5], for further details.

Let S be a complete smooth surface embedded in $\mathbb{R}^3$. The embedding induces a Riemannian metric, defining the length $L(\gamma)$ of any curve γ on S. The *geodesic distance* $d(p, q)$ between two points $p, q \in S$ is the infimum of length L over all curves γ having their endpoints at p and q; one such curve γ_{pq} satisfying $L(\gamma_{pq}) = d(p, q)$ is called a *shortest geodesic path* between p and q.

A shortest geodesic path may not be unique. The *cut locus* of a point p is defined as the closure of the set of points that can be connected to p with more than one minimal geodesic. A *normal ball* centered at p is a ball that does not intersect the cut locus of p. A set $U \subset S$ is said to be *strongly convex* if for each pair of points p and q in U there exists a unique shortest geodesic path γ_{pq} connecting p to q in S, which is entirely contained in U. A convex set U cannot extend beyond a maximal normal ball centered at any of its points. Figure 2 shows the cut locus (red curve) of a point (blue bullet) on a torus, and a maximal strongly convex ball (blue circle) centered at that point.

Geodesic curves can be also characterized by their *straightness*. Intuitively, from an extrinsic point of view, a geodesic curve does not make any further turn except the strictly necessary to follow the curvature of S: it turns *with* S, but it does not turn *on* S. Thus, geodesics play the role of straight lines on S.

Similarly to straight lines in Euclidean space, a geodesic curve may extend indefinitely and is completely defined by a point p and its tangent vector in the tangent plane T_pS at p. The *exponential map* $\exp_p : T_pS \longrightarrow S$ maps vectors of the tangent plane to points on the surface, where point $\exp_p(t)$ is defined as the other endpoint of the geodesic traced from p in the direction of t for length $|t|$. The *injectivity radius* of p is the maximum radius r_p such that $\exp_p(t)$ is invertible at all t such that $|t| < r_p$, and a geodesic ball B centered at p with radius $r < r_p$ is said to be a *normal ball.* Note that r_p is indeed the

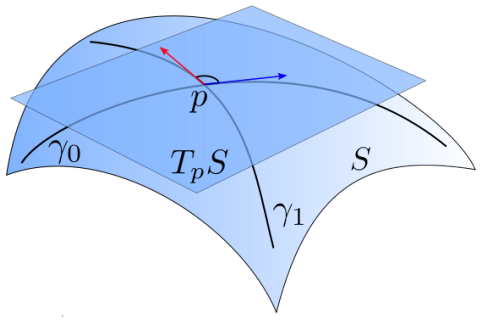


Fig. 3. Two geodesic lines γ and γ' intersecting at point $p \in S$ form an angle defined by their tangents at p on the tangent plane T_pS (red and blue arrows)

distance of p from its cut locus. The *logarithmic map*, $\log_p(q)$ is the inverse of the exp map, which associates to every $q \in B$ the vector $v_q \in T_pS$ such that $\exp_p(v_q) = q$. To fix ideas, the exp map and the log map can be seen as the counterparts of the sum of a point and a vector $P + v$ and the difference of two points $Q - P$ in an affine space, respectively.

Let γ_0 and γ_1 be two geodesics intersecting at p; the angle between them at p is defined from their tangent directions in the tangent plane T_pS. See Fig. 3 for an example. Given two points $p, q \in S$ and a curve $\gamma(t)$ such that $\gamma(0) = p$ and $\gamma(1) = q$, we can *parallel transport* a vector $v \in T_pS$ to T_qS. Essentially, parallel transporting a vector consists in defining a vector field $V(t)$ along $\gamma(t)$ such that $V(0) = v$ and $V(t)$ has no variations in the direction $\gamma'(t)$ for every t, i.e v somehow "preserves its direction" w.r.t. the tangent vector of γ when transported. A formal definition of parallel transport would require the definition of affine connection, which we omit for the sake of brevity.

With abuse of notation, in the following we will consider interchangeably the elements of T_pS as either vectors in a vector space, or points on a plane. Thus, we can consider the log map also as mapping points of S onto points of the Euclidean plane. Throughout this paper, we will denote points of S with small letters, like a, and points of T_pS as well as of the standard Euclidean plane with small letters with a bar, like $\bar{a}$, implicitly assuming a mutual mapping between a and $\bar{a}$ through the exp and log maps. We will also denote with $\bar{a}\bar{b}$ a line segment in the plane and with ab the corresponding shortest geodesic path.

Basic Straightedge and Compass Constructions in the Plane. Straightedge and compass constructions involve just points, (segments of) straight lines, and (arcs of) circles in the Euclidean plane. They consist of iteratively applying the following five basic constructions:

1. line through two existing points;
2. circle through one point with center another point;
3. intersection point of two non-parallel lines;
4. intersection points of a line and a circle;
5. intersection points of two circles.

Typical constructions usually start from few objects in the plane. In the most complex constructions – e.g., for polygons with many sides – the five operations above may be iterated many times, producing a number of intermediate objects, possibly much larger than the number of objects in the final result.

Fig. 4 illustrates the manifold counterparts of the five basic constructions on a sphere: straight-line segments are substituted with shortest geodesic paths; and circles are substituted with isolines of the distance field from a point.

The Geodesic Arsenal. Throughout the paper, we will rely on the following primitive operations to be performed on the surface S under the geodesic metric. The implementation of such operations will be addressed in Sec. 7.

- *Geodesic-tracing:* given point $p \in S$ and a tangent direction $t \in T_pS$, trace a geodesic through p with tangent vector t at p; this is equivalent to a point-wise evaluation of the exp map at p.
- *Tangent:* given a curve γ on S and one of its points p, return the direction $t \in T_pS$ tangent to γ at p; if γ is a geodesic line, this is indeed equivalent to a point-wise evaluation of the log map at p.
- *Shortest-path:* given points $p, q \in S$, return the shortest geodesic path γ_{pq} connecting them;
- *Distance-field:* given $p \in S$, compute the distance field $d_p : S \longrightarrow \mathbb{R}$ where $d_p(q) := d(p, q)$;
- *Isoline:* given the distance field d_p and a point $q \in S$ return the isoline of d_p that goes through q;
- *Intersect:* given any two lines on S, not necessarily geodesic, return their intersection points.

In the context of Sec. 4, we will rely just on the first three primitives, namely the point-wise evaluation of the exp and log map and the shortest path between two points. While in Sec. 5 we will also make use of the other primitives, to reproduce the straightedge and compass tools directly on the surface S.

4. Constructions in Tangent Space

The constructions described in this section are based on the following idea: given an initial configuration of points of S, we use the log map centered at a suited point $c \in S$ to map such points onto the tangent space T_cS. We then apply the Euclidean construction in T_cS, and finally map the result onto S through $\exp_c$. In order to preserve topological consistency, we assume all the points involved in a construction to be contained in a convex ball centered at c. Some constructions may work on a large neighborhood as well, though.

Since T_cS is a 2-dimensional vector space, we do not need any extension of the straightedge and compass tools. However, most of the properties of a given construction will be lost after applying the exponential map. Remarkably, this approach and the one described in Section 5 are somehow complementary: in many cases, the properties that one loses by using the former, can be preserved by using the latter, and vice-versa, Table 1 summarizes the results obtained with both approaches.

4.1. Operations with segments

Fig. 5 shows some basic constructions in the Euclidean case, which are extended to the manifold setting in a straightforward way. These constructions are the only ones addressed in this paper, which are also insensitive to the method used to implement them. Everything works fine because they are based just

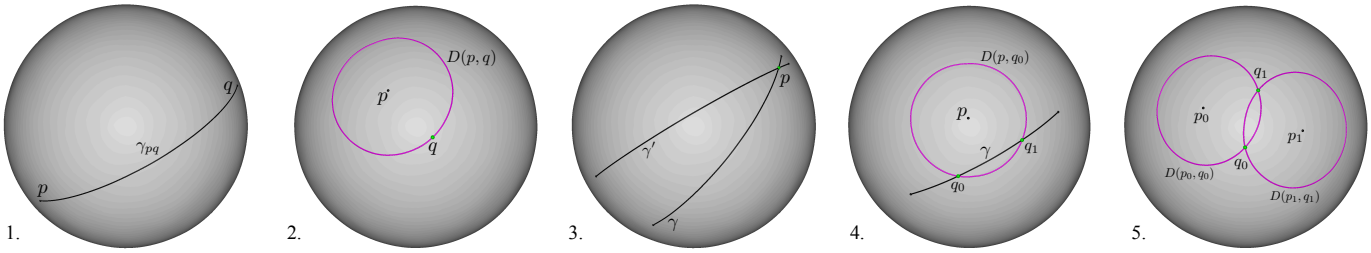


Fig. 4. The five basic constructions on a sphere. The black curves are geodesic lines, while the curves in magenta are geodesic circles. We denote with $D(p,q)$ the geodesic circle centered at point p and passing through point q.

on distance and collinearity, whose properties are preserved in the manifold setting.

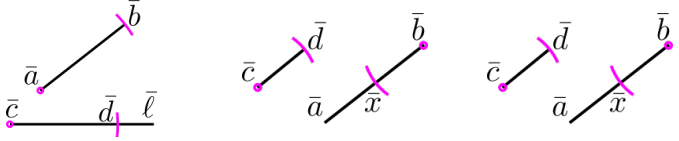


Fig. 5. Transferring the length of a segment onto another one (*left*); adding two segments (*center*); and subtracting two segments (*right*).

Given a line segment $\bar{a}\bar{b}$ in the plane and a line $\bar{\ell}$ through another point $\bar{c}$, find a point $\bar{d}$ on $\bar{\ell}$ such that $\bar{a}\bar{b}$ and $\bar{c}\bar{d}$ have the same length. In the plane, the aperture of the compass is taken at $\bar{a}\bar{b}$, then the needle point is placed at $\bar{c}$ and a circle is traced; point $\bar{d}$ is taken at an intersection of the circle with line $\bar{\ell}$.[1] In the manifold case, we start with geodesic line segment ab and a geodesic line ℓ on S. We first lift b to a point $\bar{b}$ on the tangent plane T_aS through the log map; likewise, we lift ℓ to a radial line $\bar{\ell}$ on T_cS. Then we use a standard compass to find the length of segment $\bar{a}\bar{b}$ on T_aS; and we use the same compass to draw a circle centered at c on T_cS. We find the intersection between this circle and line $\bar{\ell}$; and finally we map the intersection point to S with the exp map.

Given two line segments $\bar{a}\bar{b}$ and $\bar{c}\bar{d}$ in the plane, extend $\bar{a}\bar{b}$ at $\bar{b}$ for a length equal to $\bar{c}\bar{d}$. In the plane, segment $\bar{a}\bar{b}$ is extended to a line with the straightedge; the aperture of the compass is taken at $\bar{c}\bar{d}$ and a circle is traced by placing the needle point at $\bar{b}$; the intersection $\bar{x}$ of this circle with the line is taken, which lies on the opposite side of $\bar{a}$ wrt $\bar{b}$; line segment $\bar{a}\bar{x}$ is the result. The construction in the manifold case is analogous to the previous and is omitted for brevity.

Given two line segments $\bar{a}\bar{b}$ and $\bar{c}\bar{d}$ in the plane, shorten $\bar{a}\bar{b}$ at $\bar{b}$ by the length of $\bar{c}\bar{d}$. In the plane, the aperture of the compass is taken at $\bar{c}\bar{d}$ and a circle is traced by placing the needle point at $\bar{b}$; the intersection $\bar{x}$ of this circle with $\bar{a}\bar{b}$ is taken; line segment $\bar{a}\bar{x}$ is the result. The construction in the manifold case is also analogous to the previous ones and is omitted for brevity.

4.2. Operations with angles

In the plane, an angle is defined by two half-lines $\bar{\ell}_a$ and $\bar{\ell}_b$ incident at a point $\bar{c}$, which can be built with the straightedge, given two points $\bar{a}$ and $\bar{b}$ lying on them, respectively. Such angle can be bisected as follows. Place the needle point of the compass at $\bar{c}$, trace any circle and let $\bar{p}$ and $\bar{q}$ be its intersections with $\bar{\ell}_a$ and $\bar{\ell}_b$. Place the needle point at $\bar{p}$, and next at $\bar{q}$, with aperture $\bar{p}\bar{q}$ trace another two circles; let y be any of their two intersection points. The line $\bar{\ell}_y$ through $\bar{c}$ and $\bar{y}$ bisects the angle at $\bar{c}$. An additional property of the bisector is that all its points are equidistant from $\bar{\ell}_a$ and $\bar{\ell}_b$.

Analogously, given two geodesics γ_{ca} and γ_{cb} intersecting at c, we extend their tangent vectors at c to lines $\bar{\ell}_a$ and $\bar{\ell}_b$ in T_cS, and use the Euclidean construction to find line $\bar{\ell}_y$ as above; then we map $\bar{\ell}_y$ to a geodesic γ_{cy} emanating from c and through $y = \exp_c(\bar{y})$. Line γ_{cy} bisects the angle, in the sense that the angles formed by its tangent at c and the tangents of the two input lines γ_{ca} and γ_{cb} at c are equal, by construction. Fig. 6 (left) illustrates this construction.

However, the points of γ_{cy} in general will *not* be equidistant from the input lines. In fact, the locus of equidistant points from the two lines is not a geodesic line in general, and finding it is beyond the scope of this paper, as it requires using the distance fields from γ_{ca} and γ_{cb}, while we limit our distance fields to have their sources at single points (see Section 3).

A number of other constructions deal with operations on angles, such as copying an angle, adding or subtracting angles, or creating angles of a few specified amplitudes. These problems are somehow local to the point c at the tip of the angle, and can be addressed by finding the tangents of the geodesic lines that define the angles at play, resolving the Euclidean construction in the tangent plane, and using the resulting directions to map the geodesics to the surface S. For this reason, we do not analyze such constructions in detail.

4.3. Perpendicular to a line and the Square-set operator

Perpendicular bisector and midpoint. In the plane, the bisector is constructed as follows. Given points $\bar{a}, \bar{b} \in \mathbb{R}^2$, first use the straightedge to trace the straight-line segment joining them. Then place the needle point of the compass at $\bar{a}$ and the pencil point at $\bar{b}$ and trace a circle; repeat the same operation with needle at $\bar{b}$ and pencil at $\bar{a}$. Let $\bar{c}, \bar{d}$ be the intersection points of the two circles; use the straightedge to trace segment $\bar{c}\bar{d}$. The *straight line* line through $\bar{c}, \bar{d}$ intersects segment $\bar{a}\bar{b}$ *orthogonally* and *at its midpoint* $\bar{c}$; this is also the *locus of points that have equal distance* from $\bar{a}$ and $\bar{b}$. The Euclidean construction is depicted at the top of Fig. 6(middle).

[1] We are assuming a non collapsible compass here; the same result can be also achieved with a collapsible compass, through a more involved procedure though.

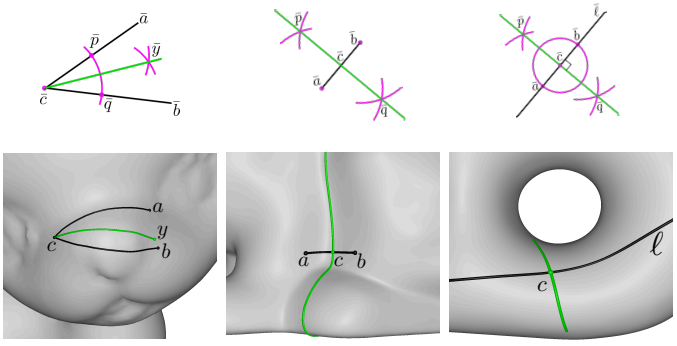


Fig. 6. Euclidean constructions of the angle bisector(*top left*), segment bisector (*top center*) and perpendicular to a line at a point (*top right*). The corresponding constructions in the manifold case (*bottom*) are obtained by mapping the lines, which are obtained with the Euclidean constructions in the tangent plane at c, to S through the exp map.

Let now a, b be two points in S and let γ be the shortest geodesic connecting them. We cannot apply the Euclidean construction in either tangent plane T_aS or T_bS and then map the result to S, as it would not have any of the above properties in general, and we would also obtain different results in the two cases. We rather apply the Euclidean construction in tangent plane T_cS, where c is the midpoint of γ. In order to find c, we first find point $\bar{c}$ in the Euclidean construction on T_aS and we map it to S through the exp map; c is the midpoint of γ by construction. Now we consider the tangent t_c of γ at c, and we proceed as before to find the vector $t_c^\perp$ orthogonal to t_c. The result $\gamma^\perp$ is the geodesic line tangent to $t_c^\perp$ at c. The result is shown at the bottom of Fig. 6 (middle).

Note that this construction satisfies just two of the three propertie of its Euclidean counterpart, since our result it is not the locus of points equidistant from a and b. In Section 5.2 we propose a method that constructs a curve whose points satisfy this last property, without being a geodesic though.

Perpendicular to a line at a point. In the plane, let $\bar{\ell}$ be a line and $\bar{c}$ a point on it, we want to find a line through $\bar{c}$ and orthogonal to $\bar{\ell}$. To this aim, it is sufficient to trace any circle centered at $\bar{c}$, finding its intersections $\bar{a}, \bar{b}$ with $\bar{\ell}$, and then finding the bisector of line segment $\bar{a}\bar{b}$.

Such construction can be ported to the manifold setting as above. The advantage in this case is that we already know the position of c on the geodesic γ, so we just work in T_cS. Fig. 6 (right) shows both constructions. The same method can be used to find the tangent at a point a to a circle centered at x and through a. This is in fact the perpendicular to geodesic segment ax and passing through a. Fig. 7 shows such construction.

The Square-set as derived operator. Given a curve γ on S and a point c on it, the above construction can be used to compute the vector $t^\perp \in T_cS$ orthogonal to the tangent t_c of γ at c. This procedure implements an operation that we call *Square-set*, which will be used as an atomic operation in the following.

4.4. Regular Polygons

In the Euclidean plane, it is known since Gauss that t is possible to construct regular n-gons with the straightedge and com-

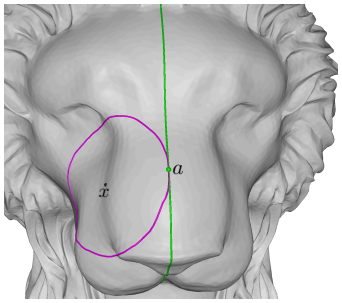


Fig. 7. Line tangent to a circle centered at x and through a: the square set is placed at a and oriented according to the tangent of γ_{xa} at a.

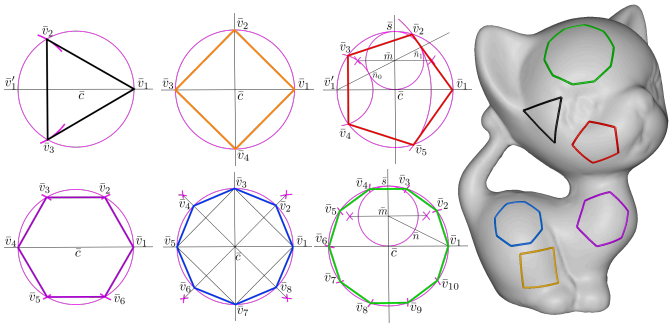


Fig. 8. Euclidean construction of an inscribed regular n-gon for $n = 3, 4, 5, 6, 8, 10$ (*left*) and the results obtained by mapping such constructions on a mesh (*right*).

pass for all integers of the type

$$n = 2^j \prod_{i=1}^{m} p_i$$

where the p_i are m Fermat numbers. However, practical constructions are known only for a relatively small subset of polygons. The constructions for $n = 3, 4, 5, 6, 8, 10$ are depicted in Fig. 8. For brevity, we will not review other constructions.

In the manifold case, given two points $c, v_1 \in S$, we are interested in constructing a "regular" geodesic polygon centered at c and having one of its vertices at v_1. To this aim we lift point v_1 to point $\bar{v}_1$ in the tangent plane T_cS. Next, we apply the proper construction in tangent space, as in Fig. 8; then we map the vertices of the resulting polygon to S with the exp map; and finally we connect them with geodesic lines.

Note that in general the result will neither have sides of equal length, nor equal angles at its corners. The only guarantee is that the radial geodesic lines from its center to the vertices have all equal lengths and form angles of $2\pi/n$.

4.5. Parallelogram, rhombus and rectangle

With similar constructions one can build a parallelogram, a rhombus or a rectangle centered at a given point. See Fig. 9.

For a parallelogram, given three points c, v_1, v_2 on S, we lift them to the tangent plane T_cS, perform the 2D construction, map the other two vertices $\bar{v}_3$ and $\bar{v}_4$ of the result to points v_3 and v_4 of S with the exp map, and finally connect v_1, v_2, v_3, v_4 with shortest geodesic paths.

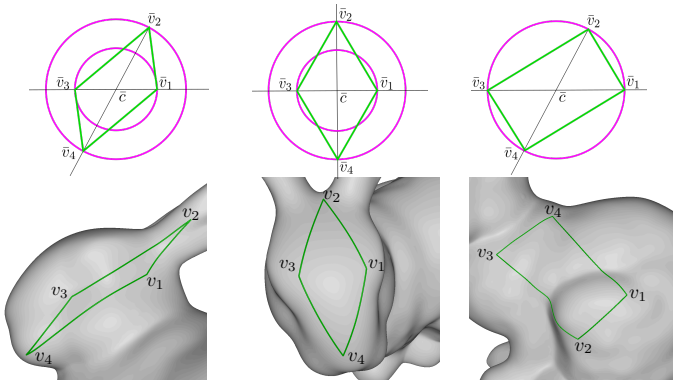


Fig. 9. Euclidean constructions of a parallelogram, a rhombus and a rectangle (*top*) and the results obtained by mapping such constructions to a mesh (*bottom*).

For a rhombus, the construction is similar. Three points c, v_1 and w are taken on S and lifted to T_cS. Point $\bar{w}$ only sets the radius of the outer circle, while the vertices $\bar{v}_2, \bar{v}_3, \bar{v}_4$ are taken at the intersections between the two circles and the two perpendicular axes obtained from c and $\bar{v}_1$ as described in Sec. 4.3.

For a rectangle, three points c, v_1 and w are taken on S and lifted to T_cS. Vertex $\bar{v}_3$ is taken at the other intersection between the line through c and v_1 and the circle through v_1; and vertices $\bar{v}_2$ and $\bar{v}_4$ are taken at the intersections between the same circle and the line through c and w.

Also in this case, we have no guarantees of equal length of opposite sides or equal angles at opposite corners; let alone the notion of "parallel sides", which is ill defined on a manifold. The only guarantee is that opposite semi-diagonals lie on a geodesic through c and have equal lengths, and, consequently, opposite angles at the center are equal. The rhombus has the additional property that the diagonals are orthogonal. And the rectangle has all four semi-diagonals with the same length.

In summary, all constructions above can guarantee only properties related to lengths and angles that depend just on the radial geodesics emanating from the center c, at which the tangent plane is placed. Note that the sides of the geodesic polygons are traced only *after* their corners have been mapped to S through the exp map. The length of such lines, as well as the angles they form at the corners, are influenced from the Gaussian curvature of S in the region covered by the polygon: the more the Gaussian curvature around c varies, the more the shape of the geodesic polygon will differ from its Euclidean counterpart.

5. Direct Constructions on the Surface

We now change approach, by defining the equivalent tools for the straightedge and compass directly on S. Referring to the geodesic arsenal defined in Sec. 3, the operator *Shortest-path* allows us to trace geodesic segments between any two endpoints; and the joint use of *Tangent* and *Geodesic-tracing* allows us to extend such a segment indefinitely from both sides. We thus define the derived operation *Geodesic-line* that traces an arbitrarily long line through a pair of points, generalizing the straightedge to the manifold setting.

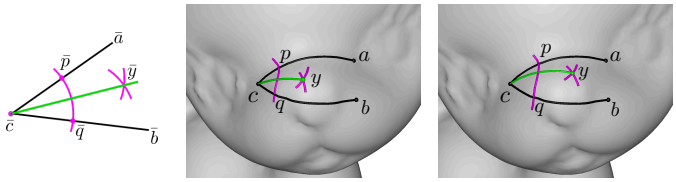


Fig. 10. The Euclidean construction to bisect an angle (*left*) fails when ported to a surface: the resulting lines (*center* and *right*) do not bisect the angle at c and they are different depending on the choice of points p and q.

Likewise, the *Geodesic-compass* is a derived operation defined as the *Isoline* through a given point of the *Distance-field* from another center point. Note that, the *Distance-field* alone does not belong to the straightedge and compass framework, because it implicitly takes measures. On the other hand, since this operator is anyhow necessary to implement the *Geodesic-compass*, we will use it also directly to address constructions where the basic tools fail.

We address the five basic constructions listed in Sec. 3 by means of *Geodesic-line* (1); *Geodesic compass* (2); and *Intersect* (3, 4, 5), as depicted in Fig. 4. Besides, we will make use of the *Square-set* operator, as defined in Sec. 4.3.

As in the previous case, we will assume that our constructions are contained inside convex sets. Thus, any geodesic line segment will be the unique shortest geodesic path between its endpoints; and any geodesic circle will be homeomorphic to the standard circle. Moreover, the result of *Intersect* will always be topologically consistent with what happens in the plane: a geodesic line do not self-intersect, and it intersects another (non-overlapping) geodesic line at most at a single point; and a line and a circle, or two circles, either do not intersect, or are tangent at one point, or intersect at exactly two points.

In the following, we review some straightedge and compass constructions, showing their extension to the manifold setting with this approach, as an alternative to the constructions in tangent plane presented in the previous section. Again, we will see that these constructions can preserve only some of the properties that are guaranteed in the Euclidean case.

5.1. Angle bisection

The Euclidean construction depicted in Fig. 6 (top-left) fails when ported to a surface with our geodesic tools. See Fig. 10. The geodesic line through c and y neither bisects the angle at c, nor its points are equidistant from the input lines. Moreover, the result depends on the radius chosen to find p and q.

5.2. Line segment bisector and midpoint

The Euclidean construction described in Sec. 4.3 also fails on a surface. If we use *Geodesic-line* and *Geodesic-compass* to obtain points p, q, the two geodesic paths γ_{ab} and γ_{pq} in general will not intersect at the midpoint of γ_{ab}, nor they will be orthogonal at c. Concerning distances, we only know that p and q are equidistant from a and b, but distances can be different at all other points of γ_{pq}. See Fig. 11 (left).

We thus resort to our additional tools. Let d_a, d_b be the two distance fields with sources at a and b, respectively. Compute the difference field $d_{ab} = d_a - d_b$; the point p computed before

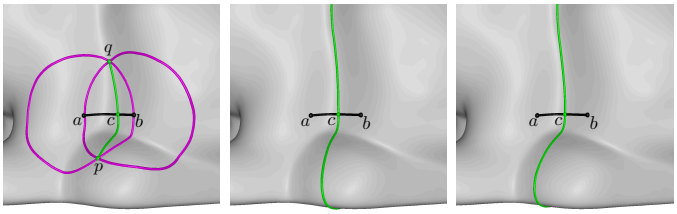


Fig. 11. The bisector of a geodesic segment computed by reproducing the Euclidean construction (*left*); by the zero isoline of the difference of distance fields from a and b (*center*); and by tracing a geodesic from the midpoint of the segment along the orthogonal direction computed with the *Square-set* (*right*). The last construction is equivalent to the one in Sec. 4.3.

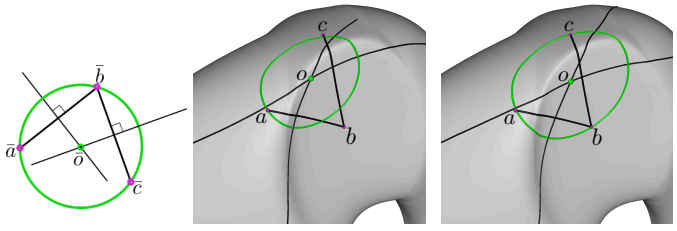


Fig. 12. Euclidean constructions of a circle through three points a, b, c (*left*). A straightforward reproduction of the Euclidean construction fails because the intersection o of the two thin black lines is not equidistant from a, b, c (*center*). The intersection of curves obtained as isolines of the difference distance fields from pairs of points gives the correct center of the geodesic circle (*right*).

belong to the zero isoline of this field. If we extract the *Isoline* of d_{ab} through p, the resulting line $\gamma^{\perp}_{ab}$ will intersect orthogonally γ_{ab} at its midpoint. See Fig. 11 (center). This construction has the further property, which we did not have with the construction in Sec. 4.3, that all points of $\gamma^{\perp}_{ab}$ are equidistant from a and b. However, $\gamma^{\perp}_{ab}$ is not a geodesic line, hence not *straight* in the manifold sense. Finally, the construction presented in Sec. 4.3 can be replicated by first finding the midpoint c of geodesic γ_{ab}, as above, and then applying the *Square-set* operator at c to find the perpendicular line. See Fig. 11 (right).

5.3. Circle through three non-collinear points

In the plane, given three non-collinear points $\bar{a}, \bar{b}, \bar{c}$, this construction can be done by first computing the perpendicular bisectors of segment $\bar{a}\bar{b}$ and $\bar{b}\bar{c}$; then intersecting such two bisectors at point $\bar{o}$; and finally tracing the circle centered at $\bar{o}$ and through $\bar{a}$ (and, consequently, through $\bar{b}$ and $\bar{c}$). See Fig. 12(left). The same procedure trivially gives the circle circumscribed to a triangle $\bar{a}\bar{b}\bar{c}$.

This construction relies on the fact that all points on a bisector are equidistant from the endpoints of the input segment, a property which is not fulfilled in the manifold case when the bisector is a straight line, as in Fig. 12(middle). However, if the two bisectors are obtained as isolines of the difference distance field, as described above, then their intersection will indeed be equidistant from the three points, hence we can use it as the center for a geodesic circle through them. See Fig. 12(right).

Note that this construction cannot be replicated while working in tangent space, because one would need to know the center o of the circle in advance.

5.4. Polygons

We already observed that the constructions in Sec. 4.4 and Sec. 4.5 do not ensure any property concerning the length of the sides and/or the amplitude of the internal angles of the resulting polygons. We now address some of those properties with direct constructions on S. To this aim, we rely on different Euclidean constructions, which do not work inside a circle.

Triangles. A triangle can be copied to another place with the same construction, both in the planar and in the manifold setting. Let $\bar{a}\bar{b}\bar{c}$ be a triangle, $\bar{\ell}$ a line and $\bar{a}'$ a point on $\bar{\ell}$. We want to copy the triangle in such a way that $\bar{a}$ goes to $\bar{a}'$, $\bar{b}$ goes to a point $\bar{b}'$ on $\bar{\ell}$, and $\bar{c}$ is placed at a point $\bar{c}'$ accordingly. We first draw a circle with amplitude $\bar{a}\bar{b}$ centered at $\bar{a}'$ and we select a point $\bar{b}'$ as one of the two intersections of the circle with line $\bar{\ell}$. Next we trace two more circles, one with amplitude $\bar{a}\bar{c}$ centered at $\bar{a}'$ and another with amplitude $\bar{b}\bar{c}$ centered at $\bar{b}'$; we select point $\bar{c}'$ as one of the intersections of such two circles. In the manifold setting, the result is a triangle with edges of the same length of abc, but nothing can be said about its angles. Moving a triangle while preserving the amplitude of its angles is inherently impossible in general, for consequences of the Gauss-Bonnet theorem.

Creating an equilateral triangle is among the simplest constructions: given an edge $\bar{a}\bar{b}$, intersect the two circles with radius $\bar{a}\bar{b}$ and centered at $\bar{a}$ and $\bar{b}$, respectively. Any of their two intersections can be chosen as the third vertex $\bar{c}$ of the triangle. The same procedure works in the manifold setting too, if we aim at obtaining a triangle with three edges of the same length. This does not guarantee any other of the properties of the equilateral triangles, e.g., having three equal angles, having three equal heights that bisect the angles and bisect the edges, etc. Constructions fulfilling even one of such requirements seem not easy to obtain in the manifold setting.

Likewise, it is easy to build an isosceles triangle on a basis $\bar{a}\bar{b}$ with the diagonal edges of a given length (transferred with the compass from some given segment). Alternatively, one can build an isosceles triangle of a given height, by first constructing the perpendicular bisector of $\bar{a}\bar{b}$ and then transferring the height on it with the compass. Both such constructions work to some extent in the manifold setting, too. However, the first construction will not warrant anything about either equality of the angles at the basis, or the height from c to bisect ab. While the second construction will just warrant the latter property, but neither that the diagonal edges, nor that the angles at the basis are equal. In our system, we implemented a more practical, yet equivalent, variant of the first construction: we consider the *Isoline* of points equidistant from a and b, as in Sec. 5.2, and we let the user choose the length of the sides by dragging point c along such bisector. Fig. 13 (bottom) shows Euclidean constructions for equilateral and isosceles triangles, together with their counterparts on a surface.

Squares and rectangles. A square can be built from one of its edges $\bar{a}\bar{b}$ as follows. A line perpendicular to $\bar{a}\bar{b}$ and through $\bar{a}$ is built first. Then the the length of $\bar{a}\bar{b}$ is transferred to segment ad on such a line by placing the needle point of the compass at $\bar{a}$. Finally, the needle point of the compass is placed at $\bar{b}$ and at

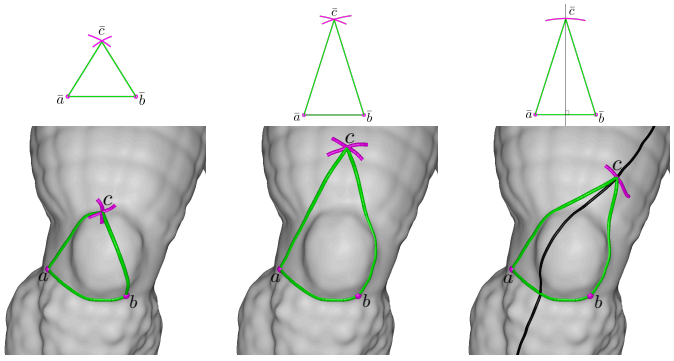


Fig. 13. Straightedge and compass constructions of an equilateral (*left*) and isosceles triangle given the length of the sides (*center*) and the height (*right*) in the Euclidean (*top*) and manifold setting (*bottom*).

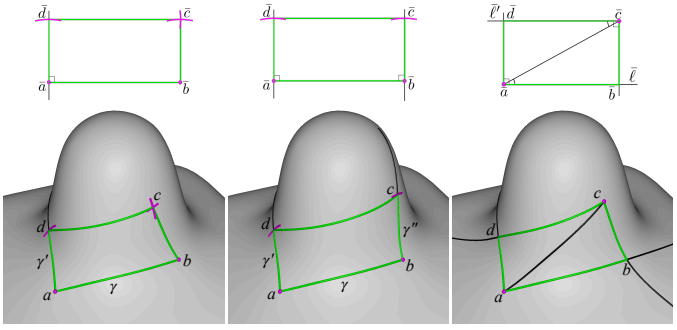


Fig. 14. Rectangles obtained with different constructions: by tracing two perpendicular lines γ and γ' intersecting at a and tracing opposite sides of the same length (*left*); by tracing two lines γ' and γ'' perpendicular to γ at a and b and setting points d and c on γ' and γ'' at equal distance from a and b, respectively (*center*); by tracing the diagonal ac, transferring angle bac to acd and tracing two lines perpendicular to ab and cd at a and c, respectively (*right*). The constructions are shown both in the Euclidean (*top*) and in the manifold (*bottom*) setting.

$\bar{d}$ with the same aperture $\bar{a}\bar{b}$, and the intersection $\bar{c}$ of the two circles gives the last vertex of square $\bar{a}\bar{b}\bar{c}\bar{d}$.

This same construction works in the manifold setting too. However, the resulting polygon will have four edges of equal length, but only angle $d\hat{a}b$ is guaranteed to be a square angle. An alternative construction consists of tracing perpendicular lines at both a and b, by means of the *Square-set*, transferring the length of ab on both of them, and connecting the points c and d obtained in this way. In this case, in the manifold setting we obtain a quadrilateral with three edges of the same length, namely ab, ad and bc, and two right angles $d\hat{a}b$ and $a\hat{b}c$; but nothing can be said about the length of edge cd and the amplitude of angles at c and d.

The same constructions apply to draw a rectangle, except that the aperture of the compass to obtain the vertical edges can be different than the length of ab. The outcome in the manifold setting has the analogous (lack of) properties.

We describe a third construction, which is more appropriate to the GUI of drawing systems. Given a basis line $\bar{\ell}$ and a point $\bar{a}$ lying on it, a diagonal segment $\bar{a}\bar{c}$ is traced first. Then the angle between such segment and line $\bar{\ell}$ is transferred at $\bar{c}$, to obtain a line $\bar{\ell}'$ parallel to $\bar{\ell}$. Finally, two lines are traced through $\bar{a}$ and $\bar{c}$, which are perpendicular to $\bar{\ell}$ and $\bar{\ell}'$, respectively. The

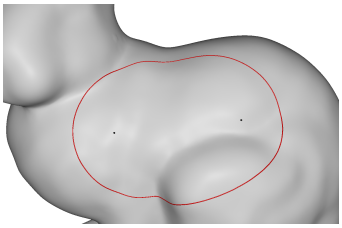


Fig. 15. Ellipse (red curve) computed as an isoline of the sum of the distance fields from its foci (black dots).

intersections of such lines with the first two lines give the other two vertices $\bar{b}$ and $\bar{d}$ of the rectangle. This construction applies to the manifold setting, too, by copying the angle in the tangent planes, as described in Sec. 5.1, and using the square set to trace perpendicular lines. However, the resulting quadrilateral has two square angles at a and c, but nothing can be said on the amplitude of the other two angles, and opposite edges are not congruent in general. A number of other constructions can be devised, which are all equivalent in the Euclidean setting, while none of them can warrant congruent opposite edges and four right angles. Each such construction privileges some of the properties of rectangles, at the expense of others.

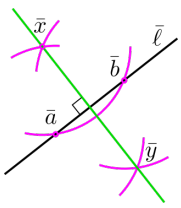
Fig. 14 shows examples of rectangles obtained with the three constructions described above.

5.5. Ellipse

An ellipse cannot be constructed using the straightedge and compass. The best one can do is to compute the position of a point on the ellipse, using the so called *de La Hire*'s construction. However, since the ellipse can be defined as an isoline of the sum of the distance fields from its foci a and b. In details, once the foci have been chosen, we use the *Isoline* operator to compute the isoline of the field $f = d_a + d_b$ equal to $\alpha\ell$, where ℓ is the distance between the two foci and α is a scaling factor. We therefore added this primitive to our drawing system for practical reasons. An example is shown in Fig.15.

6. Unresolved Contructions

We could not find a straightforward way to port further constructions to a surface by relying just on the basic tools available in our arsenal. For completeness, we briefly discuss some such constructions, which may be relevant in the applications, leaving their investigation to future work.



Perpendicular to a line through a point not on the line. This is a basic construction, which is also useful in the context of more complex constructions in the Euclidean plane. Given a line $\bar{\ell}$ and a point $\bar{x}$ not on the line, find a line through $\bar{x}$ and perpendicular to $\bar{\ell}$. In the plane, we trace a circle centered at $\bar{x}$, with an aperture larger than its distance from $\bar{\ell}$; we find the intersection points $\bar{a}, \bar{b}$ of this circle with $\bar{\ell}$; and we trace another two circles centered at $\bar{a}$ and $\bar{b}$ with the same aperture. The result is the line through $\bar{x}$ and $\bar{y}$.

In the geodesic setting, the orthogonal projection of a point x onto a geodesic γ in general will not be the midpoint of the segment intercepted on γ with a circle centered at x. We rather have to define the problem in terms of distances: if point z on γ minimizes the distance from x, then the geodesic path γ_{xz} meets γ orthogonally at z, because it is a radial path of the circle centered at x and tangent to γ. This problem could be tackled by computing the distance field from γ (which is not part of our arsenal, though) and evaluating it at x: the geodesic circle centered at x with radius $d_\gamma(x)$ is tangent to γ at z. Alternatively, one could restrict the distance field d_x to γ and find its minimum along it. Notice that both such solutions take measures, thus violating the rules of the straightedge and compass framework. A possible workaround consists of growing a geodesic circle centered at x until it becomes tangent to γ. The radius of the circle can be halved or doubled by relying on the basic constructions and a bisection technique can be followed. A similar problem is mirroring a point x about a line γ not containing it. Once we have found the projection z of x on γ, it is sufficient to trace a circle centered at z and through x and then find the intersection between such circle and the geodesic line through x and z.

Parallel lines. A number of constructions in the plane deal with parallel lines. In the manifold setting, the concept itself of parallel lines is ill-defined. Given a geodesic line γ_x and point on $x \in \gamma_x$, the tangent t_x of γ_x in x belongs to the tangent plane T_xS, and it is well defined its *parallel transport* to the tangent plane T_yS of another point $y \in S$. The parallel transport is a fairly complex operation that we have not considered in our preliminaries. Once the parallel transported direction t_y is given, we could trace the geodesic through y tangent to t_y and consider it "parallel" to γ_x. The trouble here is, that the direction t_y will be different depending both on the starting point x on γ_x, and on the trajectory that we choose to transport t_x to T_yS. Therefore, the result is not unique and it is somehow arbitrary. Another straightforward possibility it to take a reference line γ and define a bundle of "parallel" lines as all those lines that intersect γ with a given angle. Given points $x_0, \ldots, x_n$ along γ and the reference angle, it is possible to use the construction in Sec. 5.1 to trace such parallel lines through the points x_i. Note that, even if we were allowed to take distances, the locus of points that have a given distance from a geodesic line γ consists of two curves that in general are *not* geodesic lines. Addressing this problem thus requires first a robust notion of parallelism on a manifold.

More constructions. Several other constructions exploit relations between angles and distances, which do not hold in the manifold case. For this reason, we could not find a straightforward way to reproduce such constructions in terms of our geodesic arsenal:

- Tangents to a circle through an external point: the construction in the plane is based on the fact that an angle at the circumference in a half circle measures $\pi/2$. This is no longer true in the manifold case.
- Circle inscribed in a triangle: the construction in the plane is based on the fact that all points in the bisectors of angles are equidistant from the edges. This is no longer true in the manifold case. It is not clear how the locus of points that are equidistant from two edges can be constructed, unless the distance fields from the edges can be computed (see also the discussion in Sec. 5.1).

Note also that constructions like the trisection of an angle or doubling the volume of a cube are not possible using the straightedge and compass while they can be achieved by using a marked ruler. This suggests that in the manifold case, too, more constructions could be supported be allowing the explicit computation of distances from curves.

7. Implementation

All the constructions described in the previous sections, which apply to the manifold setting, have been implemented by means of the primitives defined in Sec. 3 and included as an extension of an existing library [17]. We have developed a prototype system that supports their interactive usage on meshes up to the size of millions of triangles.

Data structures. The surface S is represented with a piecewise flat triangular mesh M, which is represented with an indexed data structure – i.e., encoding a list of vertices V and a list of triangles F – augmented with triangle-to-triangle adjacencies to support mesh navigation.

We need to deal with generic points lying on the mesh, not just its vertices. A *mesh point* p is encoded as a triple (t, α, β) where t is the index of the triangle containing p, and α, β are two barycentric coordinates of p in t (while the third barycentric coordinate is computed by difference to the unit).

The tangent space T_pM at point p is defined in a standard way, as follows. If p lies inside a triangle t, or on an edge e, then its neighborhood is mapped isometrically to a plane T_pM, which is identified with the plane containing t, or with a plane where we flatten the two triangles sharing e, respectively. If p is a vertex, then the mapping cannot be isometric in general. In this case, we map the triangles incident at p to T_pM by rescaling the total angle they form about p to 2π; and we associate polar coordinates to each point in the one-ring of p accordingly. For every vertex v, we store the total angle about it and the coordinates of its neighbors in tangent space.

A curve γ on S is discretized as a polyline having vertices at all intersections with edges of M. A curve connecting points p and q is encoded with a strip of triangles $(t_0, \ldots, t_h)$ of M, where t_0 and t_h contain p and q, respectively, and an array of scalars $(l_0, \ldots, l_{h-1})$, where l_i encodes the intercept of the polyline with the edge common to t_i, t_{i+1} parametrized along such edge.

Shortest-path. We use the algorithm described in Mancinelli et al. [1], which is a variation of the one by Xin and Wang [18]. The algorithm first extracts a strip of triangles connecting the endpoints, by navigating the adjacency graph of the mesh. The strip is unfolded to the 2D plane and the shortest path within it is computed in linear time with the funnel algorithm [19]. Finally, turns in the path, which occur at reflex vertices on the boundary of the strip, are iteratively analyzed, and the strip is possibly modified, in case a shortcut exists from the opposite side of the turn, similarly to Xin and Wang [18]. All the details about the

	Tangent Space	**Geodesic Tools**	
Angle Bisection	is a geodesic ✓ bisects the angle ✓ is equidistant from sides ✗	is a geodesic ✓ bisects the angle ✗ is equidistant from sides ✗	
Segment Bisector	is a geodesic ✓ bisects the segment✓ is orthogonal ✓ is equidistant from endpoints ✗	is a geodesic ✗ bisects the segment ✓ is orthogonal ✓ is equidistant from endpoints ✓	
Circle through 3 points	✗	✓	
Isoscele Triangle	✗	2 equal sides ✓ apex belonging to the perpendicular bisector of the base ✗	2 equal sides ✗ apex belonging to the perpendicular bisector of the base ✓
Equilateral Triangle	equal sides ✗ equal angles ✗ radial geodesics of the same length ✓ angles at the center of 120° ✓	equal sides ✓ equal angles ✗ radial geodesics of the same length ✗ angles at the center of 120° ✗	
Square	equal sides ✗ equal angles ✗ radial geodesics of the same length ✓ angles at the center of 90° ✓ diagonals intersect at their midpoints ✓	4 equal sides ✓ 1 right angle ✓ radial geodesics of the same length ✗ angles at the center of 90° ✗ diagonals intersect at their midpoints ✗	2 equal sides ✓ 2 right angles ✓ radial geodesics of the same length ✗ angles at the center of 90° ✗ diagonals intersect at their midpoints ✗
Rectangles	equal sides ✗ equal angles ✗ radial geodesics of the same length ✓ diagonals intersect at their midpoints ✓	4 equal sides ✓ 1 rigth angle ✓ radial geodesics of the same length ✗ diagonals intersect at their midpoints ✗	2 equal sides ✓ 2 right angles ✓ radial geodesics of the same length ✗ diagonals intersect at their midpoints ✗
Polygons	equal sides ✗ equal angles ✗ radial geodesics of the same length ✓ equal angles at center (only regular ones) ✓	✗	

Table 1. Summary of the supported constructions, with the properties preserved by the algorithms described in Sec. 4 (Tangent Space) and in Sec. 5 (Geodesic Tools). Where more than one construction is available, we report the main differences between them splitting the corresponding columns. In most cases, the properties that fail with one approach are preserved by the other.

implementation and comparisons with state-of-the-art methods can be found in Mancinelli et al. [1].

Tangent. Let γ be the shortest path connecting two points q_0, q_1, represented as described above. Given the representation of polylines described above, the tangent vector w_p at a point p on γ is computed as follows. If p lies in a triangle t_i then w_p belongs to the plane containing t_i, and it is computed as $w_p = p_i - p$, with $p_i := (1 - l_i)v_0 + l_i v_1$ where v_0, v_1 are the endpoints of the edge shared by t_i and t_{i+1}. If p belongs to an edge, we proceed in a similar way: the only difference is that w_p will be of the form $p_{i+1} - p_i$, with obvious meaning of the notations. If p is a vertex, we first compute w_p as before, obtaining a vector defined in the plane containing a triangle t_i in the one-ring of p. Then we map w_p to the tangent space of p in the same way we have mapped its neighbors.

Parallel Tranport. Let p, q be two points on M, and let w_p a vector in the tangent space of p. In order to parallel transport w_p to a vector w_q in the tangent space of q, we operate as follows. We first evaluate the shortest path connecting p to q, as explained before. Then we transport the vector through the vertices of the polyline representing such path. In order to do this, we iteratively apply primitive operations to parallel transport across the tangent spaces of two adjacent triangles, or a triangle and one of its vertices. Such primitives have been described in Knöppel et al. [20] and are omitted here for brevity.

Geodesic-tracing. We adopt the *straightest geodesics* method proposed by Polthier and Schmies [21], which is based on parallel transport. Given an initial point p and an initial direction w in its tangent space, we trace a straight line segment in the direction w within the triangle t containing p, until intersecting the boundary of t. Then we parallel transport w to the tangent space either of a neighbor of t, or of one of its vertices, depending on the point of intersection. Each time we cross an edge, we repeat the operation on the adjacent triangle. Whenever we cross a vertex v, we extend the line along the same direction in the tangent space of v, entering another triangle of its one-ring, and we proceed likewise. We iterate this process until the traced path reaches the desired length.

Distance-field. We rely on a very efficient graph-based solver, similar to the techniques described in Nazzaro et al. [2] and Wang et al. [22], on a different graph though. When loading a mesh, we pre-compute a graph having all the nodes of V as vertices, and arcs from each vertex v to all vertices in its k-ring. Each arc is weighted with the length of the shortest path between two vertices, computed as described before. During the online phase, given a generic point p on M, we first compute the exact distances from p to the vertices of its containing triangle t and to the vertices opposite to the edges of t, and we initialize the distances of the corresponding nodes of the graph accordingly. Then, we propagate the distance field to all other vertices, by using a heuristic algorithm, which results more efficient than a classical Dijkstra search. See Nazzaro et al. [2] for further details. The parameter k used in building the graph provides a trade-off between the accuracy and the cost of the solver. A low value of k may compromise the smoothness of the distance field, hence resulting in wiggly isolines; while a too high value may slow down computations considerably, thus hindering interaction. We used $k = 3, 4$ in all our experiments.

Isoline. We linearly interpolate a field inside each triangle of M. For each triangle t, which crosses a given isovalue, the segment of isoline crossing t is computed independently. While linear interpolation is good enough on high resolution meshes, it might be too rough on coarse meshes. Better results can be

achieved by supersampling the polyline while using a more accurate estimate of the distance field inside t. For instance, isolines can be approximated as arcs of circle in the plane containing t, where the center of the circle is estimated on the basis of the values of the distance field at the vertices of t.

Intersect. Lines on M are encoded as paths, as described before. Intersections between a pair of lines are found in linear time in the total number of triangles in the corresponding paths. Each triangle intersecting one of the paths is assigned a unique tag; next the triangles forming the other path are scanned, and intersections are computed just at tagged triangles.

Convexity balls. As already remarked, we assume that our constructions occur within a convex set. We thus provide an algorithm to test the radius of convexity about a given point p. This is computed by considering the largest ball B centered at p within which the Hessian of $d_p^2(x)$ is definite-positive for every $x \in B$. By estimating the metric tensor g at every point, as described by Mancinelli et al. [23], the Hessian matrix can be expressed in terms of g, then testing its positive-definiteness reduces to checking whether or not its eigenvalues are positive. Note that, while a test of convexity guarantees correctness, in practice many constructions may work well also on larger neighborhoods. Since all constructions are interactive, we leave freedom to the user to apply them over arbitrarily large regions.

Rotation, translation, scaling. In order to support interaction, we allow the user to edit a drawing by translating, rotating and scaling geometric objects over the surface. Geometric transformations are applied to the control points that define our constructions, while the objects are generated each time from the updated points. Given an anchor point p, we use the log map to represent all control points of the object at hand in the tangent plane of p. This is implemented point-wise by evaluating the shortest paths between p and each such point, and finding the tangent of each path at p. Rotation and scaling are implemented trivially, by changing one of the polar coordinates of the points in tangent space: the angle for rotation and the distance for scaling. Then we map the updated points to the surface with the exp map, which is implemented point-wise by tracing geodesic lines from p either in the updated directions, or with updated lengths. Translation consists of dragging the anchor point while parallel transporting the reference frame of its tangent space along the trajectory. Upon dragging, the control points are regenerated likewise from the transported frame.

For most constructions described in Sec. 4, a natural choice for the anchor point is the center c where we locate the tangent space for the construction. For the remaining constructions, we select as anchor point one of the control points participating in the construction.

Macros. We provide some macro-operations, which combine different primitive constructions to obtain complex decorations at once. Some examples of macros are shown in the decorations in Fig. 1. For instance: a *wreath* is obtained by multiple instances of a polygon rotated about the same center; similarly, we allow the user to draw *nested shapes* like circles or polygons; a *flower* is built by drawing arcs of circles centered at the vertices of a polygon and trimming them at their intersections. The *cross* and the *spider net* on the skull are also generated by macros that intersect circles. Macros are controlled interactively with simple parameters that tune, e.g., the number of polygons forming a wreath, the number of nested shapes, the number of petals in a flower, etc.

8. Results

We have integrated our constructions in the context of a drawing system, endowed with a GUI that allows its interactive usage to decorate surfaces with geometric shapes. All primitive operations and constructions are supported via intuitive click-and-drag, which mimic the behavior of standard 2D drawing systems. We have experimented our system during interactive sessions on high-resolution meshes that range from about 100k to about 7M triangles. All images shown in this paper have been generated interactively with our system. A live session is shown in the accompanying video.

A performance and robustness evaluation of the basic algorithms listed in Sec. 7 is out of the scope of this paper, as at least the most critical of them have been evaluated thoroughly in previous works [1, 2, 23]. Our constructions are based on simple combinations of such algorithms and hardly require more than 0.1 seconds each on a laptop computer, even on the largest meshes we have tested, always remaining compatible with interaction. The only exception is given by complicated macros, such a braids, which may involve hundreds of primitives, thus requiring a few seconds to complete. For such shapes, we support a stylized version during interaction, while the final construction is performed just upon mouse button release.

Concerning off-line operations, the only time-intensive one is the construction of the graph for the solver of the distance field, which may require a few minutes to complete on the largest meshes for $k \geq 4$. Note that such graph is computed just once when loading a mesh for the first time, it can be saved on disk and retrieved quickly for later usage. As an alternative, the simpler graph adopted by Nazzaro et al. [2] can be used, which is built in negligible time, but provides a less accurate evaluation of the distance field, possibly leading to wiggly isolines on meshes with a relatively low resolution.

9. Concluding remarks

We have presented two approaches – namely, constructions in tangent space and direct constructions on the surface – to port straightedge and compass constructions to the manifold setting. It follows from our analysis that not all constructions can be ported successfully, and also those that can be ported may guarantee only *some* of the properties they have in the Euclidean case. We extended the scope of basic constructions by exploiting our *Distance-field* operator beyond the limitations of the straightedge and compass framework, yet remaining compatible with it, since we neither take explicit measures nor do arithmetic computations. The constructions we propose already support several operations in the context of interactive vector graphics on surfaces.

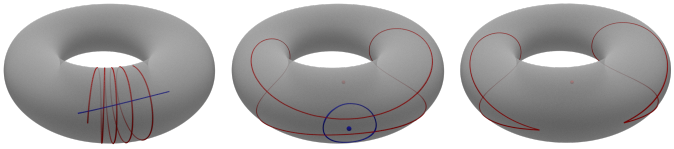


Fig. 16. The intersection of a convex set (blue) with a non-convex one (red) could be not topologically consistent with what happens in the Euclidean plane: geodesics may intersect in more than one point (*left*) and circles may intersect in more than two points (*middle*); a circle with radius greater than the injectivity radius of its center may not be even homeomorphic to a Euclidean circle, and may be not smooth at the cut locus of the center (*right*).

A few relevant constructions are still not supported. Such operations may require explicit measures, which are forbidden in the straightedge and compass framework and may require further tools beyond our geodesic arsenal, such as computing the distance field from a curve.

A further challenge is extending our primitives to work over larger regions. However, even basic properties of lines and circles can be lost outside strongly convex regions. See Fig. 16 for some examples. Some operations have been addressed already in the literature, including primitives that can be computed with distance fields [2], and Bézier splines [1].

A relevant limitation, stemming from the intrinsic curvature of surfaces, is the impossibility to warrant the congruence of both lengths and angles together. Regular tilings, which are hard to apply because of this limitation, can be addressed by relaxing some conditions on angles and/or lengths, but they remain challenging to extend over large regions. This problem is tightly related to the design of N-RoSy fields [24], in particular to the presence of field singularities, which cannot be avoided, as a consequence of the Gauss-Bonnet theorem.

A possible avenue is to relax the constraint of lines to be straight, in geodesic terms, trading some straightness for other properties. This leads to the concept of *as-straight-as-possible* lines under given constraints, e.g., joining their endpoints with a prescribed length or with given tangent directions. This approach entails investigating Jacobi fields [25, 26] and related optimization problems.

We plan to address the above challenges in our future work.

References

[1] Mancinelli, C, Nazzaro, G, Pellacini, F, Puppo, E. b/surf: Interactive bézier splines on surfaces. 2021. arXiv:2102.05921v2.

[2] Nazzaro, G, Puppo, E, Pellacini, F. geoTangle: interactive design of geodesic tangle patterns on surfaces. ACM Trans Graph 2022;41(2):12:1–12:17.

[3] Mancinelli, C, Puppo, E. Straightedge and Compass Constructions on Surfaces. In: Frosini, P, Giorgi, D, Melzi, S, Rodolà, E, editors. Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference. The Eurographics Association. ISBN 978-3-03868-165-6; 2021,doi:10.2312/stag.20211469.

[4] Cheeger, J, Ebin, D. Comparison theorems in Riemannian geometry. North-Holland mathematical library; North-Holland; 1975.

[5] Chavel, I. Riemannian Geometry: A Modern Introduction. Cambridge Studies in Advanced Mathematics; Cambridge University Press; 2006.

[6] Alexandrov, AD. Intrinsic geometry of convex surfaces. OGIZ, Moscow-Leningrad 1948;.

[7] Alexander, S, Kapovitch, V, Petrunin, A. Alexandrov geometry: preliminary version no. 1. 2019. arXiv:1903.08539.

[8] W3C. Scalable vector graphics. 2010. URL: https://www.w3.org/Graphics/SVG/.

[9] Adobe. Adobe illustrator. 2021. URL: https://adobe.com/products/illustrator.

[10] Inkscape's Contributors, . Inkscape. 2021. URL: https://inkscape.org/.

[11] Autodesk, . Mudbox. 2021. URL: https://autodesk.com/mudbox.

[12] Pilgway, . 3d-coat. 2021. URL: https://3dcoat.com.

[13] Pixologic, . Zbrush. 2022. URL: https://pixologic.com/zbrush/features/overview/.

[14] Yuksel, C, Lefebvre, S, Tarini, M. Rethinking Texture Mapping. Comp Graph Forum 2019;38(2):535–551.

[15] Crane, K, Livesu, M, Puppo, E, Qin, Y. A survey of algorithms for geodesic paths and distances. 2020. arXiv:2007.10430.

[16] Sharp, N, Crane, K. You can find geodesic paths in triangle meshes by just flipping edges. ACM Trans Graph 2020;39(6):249:1–15.

[17] Pellacini, F, Nazzaro, G, Carra, E. Yocto/GL: A Data-Oriented Library For Physically-Based Graphics. In: Agus, M, Corsini, M, Pintus, R, editors. Smart Tools and App. for Graphics - Eurographics Ital. Chap. Conf. 2019,URL: https://github.com/xelatihy/yocto-gl.

[18] Xin, SQ, Wang, GJ. Efficiently determining a locally exact shortest path on polyhedral surfaces. Computer Aided Design 2007;39(12):1081–1090.

[19] Lee, D, Preparata, F. Euclidean shortest paths in the presence of rectilinear barriers. Networks 1984;14(3):393–410.

[20] Knöppel, F, Crane, K, Pinkall, U, Schröder, P. Globally optimal direction fields. ACM Trans Graph 2013;32(4):1–14.

[21] Polthier, K, Schmies, M. Straightest geodesics on polyhedral surfaces. In: Mathematical Visualization. New York: Springer-Verlag; 1998, p. 135–150.

[22] Wang, X, Fang, Z, Wu, J, Xin, SQ, He, Y. Discrete geodesic graph (dgg) for computing geodesic distances on polyhedral surfaces. Computer Aided Geometric Design 2017;52(C):262–284.

[23] Mancinelli, C, Livesu, M, Puppo, E. Practical computation of the cut locus on discrete surfaces. Computer Graphics Forum 2021;40(5):261–273.

[24] Vaxman, A, Campen, M, Diamanti, O, Bommes, D, Hildebrandt, K, Ben-Chen, M, et al. Directional field synthesis, design, and processing. In: ACM SIGGRAPH 2017 Courses. ACM; 2017,.

[25] Pottmann, H, Huang, Q, Deng, B, Schiftner, A, Kilian, M, Guibas, L, et al. Geodesic patterns. ACM Trans Graph 2010;29(4):43:1–43:10.

[26] Le Brigant, A. A Discrete Framework to Find the Optimal Matching Between Manifold-Valued Curves. Jou of Mathematical Imaging and Vision 2019;61(1):40–70.