Feature-aligned T-meshes

Ashish Myles* New York University
Nico Pietroni† ISTI, Italian National Research Council
Denis Kovacs‡ New York University
Denis Zorin§ New York University

Abstract

High-order and regularly sampled surface representations are more efficient and compact than general meshes and considerably simplify many geometric modeling and processing algorithms. A number of recent algorithms for conversion of arbitrary meshes to regularly sampled form (typically quadrangulation) aim to align the resulting mesh with feature lines of the geometry. While resulting in a substantial improvement in mesh quality, feature alignment makes it difficult to obtain coarse regular patch partitions of the mesh.

In this paper, we propose an approach to constructing patch layouts consisting of small numbers of quadrilateral patches while maintaining good feature alignment. To achieve this, we use quadrilateral T-meshes, for which the intersection of two faces may not be the whole edge or vertex, but a part of an edge. T-meshes offer more flexibility for reduction of the number of patches and vertices in a base domain while maintaining alignment with geometric features. At the same time, T-meshes retain many desirable features of quadrangulations, allowing construction of high-order representations, easy packing of regularly sampled geometric data into textures, as well as supporting different types of discretizations for physical simulation.

CR Categories: I.3.5 [Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems

Keywords: patch layout, quadrangulation, parametrization, T-splines

1 Introduction

Subdivision surfaces, surface splines, and related multiresolution and regularly sampled surface representations are far more compact and efficient than general meshes and simplify many geometric modeling and processing algorithms. Converting arbitrary meshes to this type of representations is difficult because of many conflicting requirements for such conversions.

Most regularly-sampled surface representations consist of patches forming a *domain mesh*, with a regular pattern of samples for each patch. We focus on quadrilateral patches, as these are most commonly used.

Important requirements include (cf. [Bommes et al. 2009]):

1. **Patch quality:** Patches should be well-shaped, with minimal skew and bounded aspect ratio.
2. **Approximation:** Each patch should approximate the original mesh well.
3. **Mesh complexity:** The domain mesh should have as few vertices as possible, while satisfying other constraints.
4. **Orientation and Alignment:** In areas with well-pronounced consistent curvature directions, patch parametric lines should follow the curvature; patch boundaries should be aligned with sharp features and smooth surface boundaries.

Existing techniques offer a tradeoff between *alignment* with features and isometry and the *number* of patches in the domain mesh. Techniques allowing to keep the number of patches small have only restricted forms of alignment control, while many recent algorithms with good alignment control often yield a larger number of patches in the domain mesh.

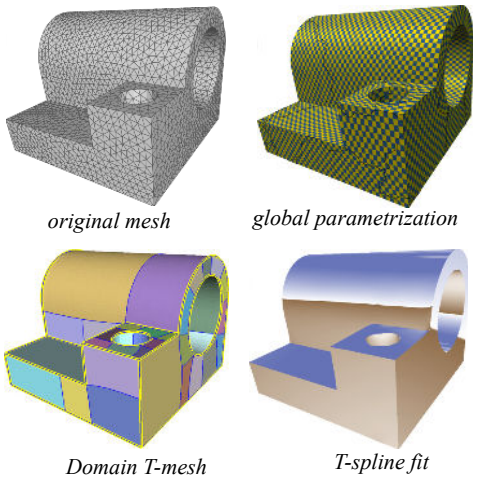


Figure 1: *Transforming the* `joint` *mesh into a T-spline surface.*

Quite often the tradeoff between the number of patches in the coarse mesh and alignment is fundamental, and not a feature of any specific algorithm: the maximal patch size in a local area is determined by the distance between nearby feature lines, which can be quite small. This local size restriction propagates globally (Figure 2) if the patch boundaries are aligned with feature lines, resulting in domain meshes with numbers of patches growing far larger than the complexity of the object suggests.

In this paper, we propose an approach to constructing domain meshes consisting of small numbers of patches while maintaining good feature alignment. Our approach is based on using domain *T-meshes*, in which the intersection of two faces may be not the whole edge or vertex, but a part of an edge. T-meshes dramatically change the relation between the total number of patches needed and the local feature size making it possible to align patches with the field without restricting their size. Thus we generate one-to-two orders of magnitude fewer patches than the coarsest quadrangulations aligned to the same features.

We show that feature-aligned coarse T-meshes are naturally obtained using recently developed global parametrization techniques for quadrangulation [Kälberer et al. 2007; Bommes et al. 2009].

While T-meshes offer more flexibility, they also retain many desirable features of domain meshes with no T-joints (*conforming*

*e-mail: amyles@cs.nyu.edu
†e-mail: nico.pietroni@isti.cnr.it
‡e-mail: kovacs@cs.nyu.edu
§e-mail: dzorin@cs.nyu.edu

ACM Reference Format
Myles, A., Pietroni, N., Kovacs, D., Zorin, D. 2010. Feature-aligned T-meshes.
ACM Trans. Graph. 29, 4, Article 117 (July 2010), 11 pages. DOI = 10.1145/1778765.1778854
http://doi.acm.org/10.1145/1778765.1778854.

meshes). Several high-order constructions (T-splines/NURCCs [Sederberg et al. 2003] and PT-splines [Li et al. 2009; Deng et al. 2008]) are available, with natural refinement structure allowing for multiresolution [Sederberg et al. 2004]. Adaptive structured meshes, a subset of T-meshes, are widely used in simulation; many local constructions developed for adaptive meshes, such as finite-volume and finite-element discretizations, can be transferred to general T-meshes ([Bazilevs et al. 2009]).

Overview. Our approach consists of the following main components: (1) global parametrization construction; (2) constrained parametrization optimization, aiming to improve T-mesh structure while maintaining field alignment; (3) construction of an initial patch layout and its optimization, and optionally, reconstruction of a T-spline approximation to the original mesh.

We use a global parametrization method closely following [Bommes et al. 2009], with some important changes discussed in Section 4, aiming to improve the quality of the feature cross-field guiding the parametrization.

The parametrization optimization step aims to reduce the number of T-joints in the domain mesh, by changing the global parametrization so that more singularities are on the same parametric lines. (Section 6).

We construct an initial patch layout with a number of patches within a constant factor from the minimal possible for a given number singularities. This initial layout may contain patches with bad approximation quality, unnecessary T-joints, and with unbalanced areas. We use a greedy constrained optimization strategy to move the patch boundaries while maintaining alignment to obtain the final layout (Section 5).

Finally, we fit a T-spline approximation to the surface, using the optimized domain T-mesh as the T-spline domain. The resulting surface can be either used directly (if the original mesh is well approximated by a piecewise-smooth surface), or used as the base for a displaced surface.

2 Related work

The literature on parameterization, quadrangulation and conversion to high-order surfaces is quite extensive, and we survey only the most closely related work. Broader reviews can be found in [Hormann et al. 2007; Sheffer et al. 2006].

A number of methods [Eck et al. 1995; Lee et al. 1998; Khodakovsky et al. 2003; Marinov and Kobbelt 2005; Daniels et al. 2009a; Daniels et al. 2009b; Pietroni et al. 2009; Tarini et al. 2010] use simplification techniques for constructing a conforming domain mesh. These techniques make it possible to obtain very coarse domain meshes, with good user control over the domain mesh size. While some degree of feature alignment is possible (cf. [Lee et al. 1998], [Marinov and Kobbelt 2005]), it is limited by the difficulty of preserving features in simplification. Other methods use global harmonic or conformal parametrizations with singularities [Gu and Yau 2003; Dong et al. 2006; Tong et al. 2006; Ben-Chen et al. 2008; Springborn et al. 2008; Kovacs et al. 2009]. While some of these methods offer a degree of control over the size and structure of the domain mesh (e.g., [Dong et al. 2006]), feature alignment is limited to determining positions of parametrization singularities. [Huang et al. 2008] describes an algorithm for adding alignment and orientation control to the parametrization, but the domain mesh is still constructed independently of geometry features.

Field-alignment techniques [Ray et al. 2006; Kälberer et al. 2007; Bommes et al. 2009] adapt the parameterization to the shape by fitting the parametrization gradient to smoothed principal curvature directions, or more generally, to a smooth *cross-field* capturing surface features. The topological structure of the field (singularities and separating lines) indirectly determines how fine the domain mesh can be. Reducing the number of singularities is often difficult without significant smoothing of the field. Furthermore, as the examples in Figure 2 show, even sparsely placed singularities do not guarantee that they are connected together in a way that allows constructing a coarse domain mesh. The method that we propose is based on field alignment. However, similar to simplification-based methods, we aim to produce coarse domain meshes, even for geometry with relatively complex features, while maintaining alignment.

In geometric modeling, *T-meshes* were considered primarily in the context of T-splines, T-NURCCs [Sederberg et al. 2003; Sederberg et al. 2004], and PT-splines [Li et al. 2009; Li et al. 2007; Deng et al. 2008]. [Li et al. 2006] demonstrated how to use periodic global parametrization (PGP) of [Ray et al. 2006] to fit T-spline surfaces to meshes. An important feature of PGP is its ability to introduce T-joints during the parametrization process. However, the complexity of the resulting domain mesh is still determined by the topological structure of the field, with significant smoothing required to make it simpler. [Eppstein and Erickson 1999] demonstrate how to use motorcycle graphs to partition a quad mesh into rectangular patches allowing T-joints, and prove bounds on the possible number of patches, but the quality of the patch layout cannot be controlled. [Carr et al. 2006] constructs rectangular geometry images (effectively, a T-mesh) by partitioning a mesh into approximately rectangular patches and parametrizing each on a rectangle, but the patches are not adapted to the geometry. [He et al. 2006] constructs a T-spline from an arbitrary mesh using global conformal parametrization. In this extreme case it is possible to have effectively a single-patch domain mesh for an arbitrary surface. As is the case with other harmonic methods, feature alignment control is limited to parametrization singularity placement.

The quality of the feature-aligned quadrangulation depends on the quality of feature detection, a difficult problem for many classes of meshes. A number of techniques for defining and detecting feature lines were proposed: [Ohtake et al. 2004; Hildebrandt et al. 2005; Weinkauf and Günther 2009]. We use ridges and valleys computed from smoothed curvature values obtained using the robust estimation of [Kalogerakis et al. 2007] to determine which curvature directions should be considered salient (Section 4).

3 Field-aligned quadrangulations

To motivate our approach, we consider constraints imposed on a conforming quadrangulation by field alignment. These considerations are not specific to any particular quadrangulation method. Recall that a cross-field (4-rosy field or 4-symmetry field) [Hertzmann and Zorin 2000; Palacios and Zhang 2007; Ray et al. 2008] is a quadruple of tangent vectors assigned to each surface point. A quadrangulation algorithm aligns the edges of the quad with the vectors of this field, so that no quad has singularities in the interior. As a consequence, a field singularity has to be a quad vertex, and there are quad edges following field integral lines starting at singularities (*separating lines*) (Figure 2, right). Chains of quadrangulation edges starting at singularities have to end at singularities, as we can always extend a chain past a regular vertex.

The most fundamental restriction on the size of mesh patches is imposed by the *distance between field singularities*, as no quad can contain singularities inside. In many cases, it is essential to place singularities close to each other for the field to follow features (Figure 2, left). If the quad size changes smoothly and is close to con-

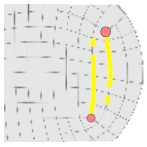


Figure 2: *Left: close singularities result in a strip of small quads. Right: a singularity close to separating line.*

stant over the mesh, a local size restriction becomes global.

More generally, it is not essential for two singularities to be close to each other for the quad size to be constrained: *it is sufficient for two separating lines starting at these singularities to be close* (Figure 2, right). As there have to be quad edges along each separating line, at best, we can produce long and thin quads bounded by these lines.

Both cases can be either due to the structure of surface features (like singularities at two close corners) or be an artifact of constructing a smooth field from the salient feature lines. In the first case, a coarse mesh may be fundamentally incompatible with being aligned with features. In the second case, the feature field can be changed to improve the parametrization without changing the quality of the alignment.

By allowing T-joints in the domain mesh, we make it possible to switch to larger-size quads away from closely spaced singularities, and terminate chains of quad edges following separating lines early, removing both restrictions. Section 5 describes our algorithm for T-mesh construction. By detecting closely spaced separating lines and adding constraints to parametrization to snap them together, we reduce the number of nonessential T-joints in the resulting domain mesh (Section 6).

The need for rounding. For a general cross-field it is possible that integral lines starting at singularities may pass arbitrarily close to each other, or any integral line may pass arbitrarily close to itself. For example, if we tilt the natural parametric lines on the torus so that the slope is irrational, any integral line of this field will be an infinite spiral around the torus.

To be able to obtain a valid quadrangulation, we need to ensure that the integral lines are closed or end at singularities. In [Kälberer et al. 2007; Bommes et al. 2009] this is ensured by a rounding procedure we discuss in greater detail in Section 4, which requires deviation of the quadrangulation lines from the original field. Creating larger quads for the domain mesh requires moving singularities further, resulting in non-aligned quadrangulations and higher distortion. T-meshes avoid the need for extreme rounding while still allowing to obtain large patches.

4 Feature-aligned parametrization

The starting point for our T-mesh construction is the global parametrization of [Bommes et al. 2009]. We briefly summarize the algorithm and the main differences in our version, as the structure of the algorithm is essential for introducing singularity constraints described in Section 6.

The algorithm computes a *global parametrization* of a mesh M, i.e., an assignment of planar (u,v) coordinates to each *triangle corner* (A triangle corner is a pair (f,w) where f is a triangle of the mesh and w is one of f's vertices). The mapping to the plane defined by these coordinates is one-to-one and orientation-preserving on each triangle. In addition, we assume that the whole mesh is mapped to a topological disk. More precisely, each vertex gets the same (u,v) coordinates in all incident triangles, excluding vertices along a *cut*, a connected graph C of mesh edges, such that $M \setminus C$ is topologically equivalent to a disk. The algorithm proceeds in several steps:

1. The shape operator is estimated on all triangles, and *salient* triangles are detected. For a salient triangle, principal curvature directions are likely to correspond to a feature; we discuss how salient triangles are detected below.
2. A cross-field, represented on each triangle T by the angle θ_T between one of four field directions and a reference edge of T, as well as integer *matchings* (Figure 3) on each edge, is optimized to minimize a measure of field smoothness. A matching determines corresponding directions on two triangles. Matching -1 means that direction 4 in triangle 2 corresponds to 1 in triangle 1. Matchings can be arbitrary (i.e., not necessarily mapping closest directions to each other). The directions are fixed on salient triangles, and the matchings are restricted to be integers. [Bommes et al. 2009] describes an efficient greedy mixed-integer solver that we use to solve the optimization problem.
3. The cut C passing through all singularities of the field is computed.
4. The cross-field is made consistent: the angles representing the field are changed so that the matchings across all non-cut edges are zero. It is possible to achieve this if the cut passes through all singularities (we refer to [Ray et al. 2008; Bommes et al. 2009] for details).
5. As the matchings are all zero at non-cut edges, if we arbitrarily label one of the directions of the field on a triangle T_o $\mathbf{u}_{T_0}$ (the target vector for ∇u), the label can be consistently propagated to all other facets. The 90-degree rotated vectors of the cross-field are labeled $\mathbf{v}_T$. The vectors $\mathbf{u}_T$ and $\mathbf{v}_T$ are the target values for the gradients of the parametric coordinates on the triangle T.
6. The parametrization is computed as a solution to the constrained minimization problem

$$\sum_{\text{triangles } T} \text{area}(T) \left(\|\nabla u - h\mathbf{u}_T\|^2 + \|\nabla v - h\mathbf{v}_T\|^2 \right) \to \min \quad (1)$$

where the scale factor h sets the correspondence between the length scale of the parametric domain and the surface.

The constraints imposed on (u,v) values correspond to transitions across seams: we want the match across seams to be the same as for the guiding cross-field: if the $\mathbf{u}_T$ direction across a seam is transformed to a $\mathbf{v}_T$ direction, then the parametric directions are transformed in the same way. More precisely, if two triangles T and T' share a cut edge e, with parametric positions of endpoint corners $p_1 = (u_1, v_1)$ and $p_2 = (u_2, v_2)$ on one side of the cut, and p_1' and p_2' on the other side, these are related by

$$p_1' = R_e p_1 + t_e, \qquad p_2' = R_e p_2 + t_e$$

where R_e is a $k_e \pi/2$ rotation defined by the matching k_e of the cross-field on the edge, and t_e is an unknown translation.

For models with sharp features, apart from transition constraints, constraints are imposed on parametric coordinates of vertices on sharp edges: for example, if the direction of a sharp edge is close to $\mathbf{u}_T$, its vertices are constrained to have the same v coordinates.

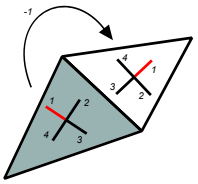
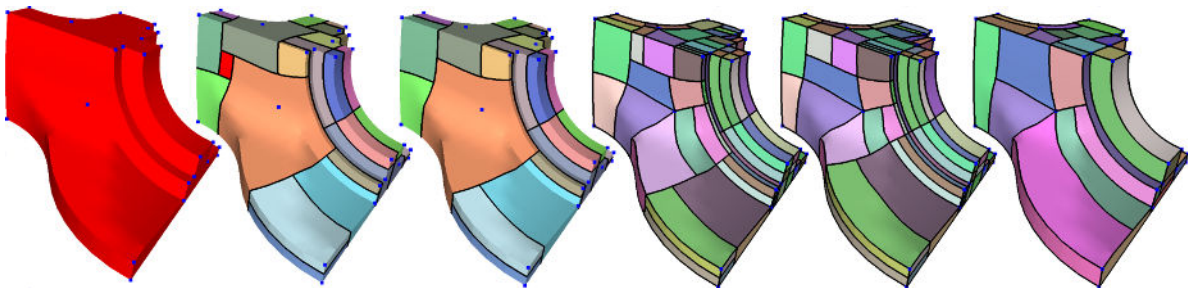


Figure 3: *Matchings between cross-fields in adjacent triangles.*

To make a conforming quadrangulation possible, [Bommes et al. 2009] require that the translational parts of transition maps t_e to be integers. In addition, all parametrization singularities are required to be at integer locations. These constraints ensures that the cross field formed by ∇u and ∇v does not have infinite separating lines of the type discussed in Section 3. When the quadrangulation is generated by tracing the integer parametric lines on the surface, rounding ensures that the field singularities are at quad corners and that quad edges are continued seamlessly across cut edges of the mesh. Note that compared to "unrounded" global parametrization that minimizes (1) with no constraints, for large values of h rounding forces the parametric line directions further away from the cross-field directions.

Quite often, the algorithm described above yields parametrizations with inverted triangles; as a result the parametrization has more



(a) (b) (c) (d) (e) (f)

Figure 5: *Main steps of the T-mesh patch layout construction. (a) An initial set of vertices are placed at singularities. (b) Cells with field-aligned edges are uniformly expanded from singularities, until no further expansion is possible. (c) Holes between cells are closed by adjusting cell boundaries. (d) Cells are split into quad patches. (e) T-joints are eliminated whenever possible by moving cell boundaries. (f) The number and shape of the cells are optimized to minimize an energy and satisfy the constraints.*

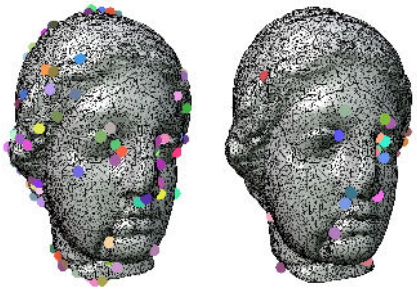


Figure 4: *Facet-based (left) vs. vertex-based cross-field optimization (right). For close salient fields, the vertex-based field optimization produces 34 singularities vs. 139 for facet-based.*

singularities than the original field, and these singularities are not at integer locations. To solve this problem, following [Bommes et al. 2009], constrained energy optimization is repeated several times with gradually increasing (*stiffened*) weights assigned to the terms corresponding to triangles in areas with high parametric distortion.

Our algorithm differs from the algorithm of [Bommes et al. 2009] in three main respects.

Salient feature detection. In [Bommes et al. 2009], salient feature detection is based on thresholding per-triangle total curvature and shape operator anisotropy (the ratio of principal curvatures). Instead, following [Ohtake et al. 2004; Hildebrandt et al. 2005] we use ridges and valleys, computed from a smoothed curvature field to identify salient facets and vertices. Ridges also require thresholding, and we use ridge strength as described in [Ohtake et al. 2004].

Field optimization. While we found that the triangle-based cross-field optimization produces good results in the case of meshes with well-shaped triangles, we also observed that a large number of singularities is often formed for surfaces with lower triangle quality. Instead of using tangent vectors at facets, we define a tangent plane at each vertex v, and a cross-field at v, with a reference direction for the angle θ_v chosen to be the projection of an edge connected to v to the tangent plane. Similarly to the triangle cross-field case, we define matchings on edges, but this time these indicate correspondences between cross-fields at two incident vertices, rather than triangles. For subsequent parametrization, the cross-field at vertices is converted to a facet-based field by averaging the cosines and sines of quadruple the angles in a common reference frame; this is justified by the 4^{th}-order tensor formulation of [Palacios and Zhang 2007].

Translation rounding. For T-mesh construction, rounding of translation variables and singularity positions is not fundamentally required, as the separating lines starting at singularities can be terminated at T-joints. However, we do perform a modest amount of rounding (on the order of triangle size of the original mesh) to make it possible to generate a fine-scale conforming quadrangulation that we use to implement our T-mesh construction algorithms, as explained in Section 5. As all singularity position changes are relatively small, we do not need to use the mixed-integer solver for setting singularity positions: they are all adjusted simultaneously as in [Kälberer et al. 2007].

5 Construction and optimization of domain T-meshes

Our goal is to construct a T-mesh with a small number of faces, with edges following the parametric lines of the global parametrization constructed in Section 4 and satisfying a number of quality constraints. In this section for simplicity we assume that the parametrization is fixed, and describe how the T-mesh can be constructed. In the next section we discuss a method for adjusting the parametrization to eliminate some non-essential T-joints.

The main steps of the construction are summarized in Figure 5. All steps in this section use the fine quadrangulation generated in the previous section.

5.1 Field-aligned T-meshes and operations on them

To describe our algorithms we introduce the basic terminology for T-meshes. We consider quadrilateral T-meshes: Conceptually, every face of this mesh is a quad, but some of the quad edges may be split into several subedges by T-joints. Each vertex is one of three types: labeled *T-joint* (always of valence 3), *regular* (valence 4 in the interior, 3 on the boundary) or *extraordinary* (non-T-joint interior vertices of valence different from 4). For exactly one edge incident at a T-joint vertex v we mark its endpoints at v as *stem*, and the other two as *nonstem*. Thus, we say that a vertex is T-joint with respect to a face if it is incident to two non-stem edges comprising the face (Figure 6).

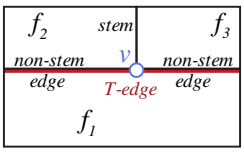


Figure 6: Notation. *Vertex v is T-joint with respect to face f_1, but a corner for f_2 and f_3.*

We distinguish between mesh edges and *T-edges*. Each face has exactly four *corner* vertices – those that are not T-joint with respect to the face; T-edges are unions of edges between two sequential corner vertices of a face. We assume that no face is glued to itself: the starting and ending vertices of a T-edge are always distinct.

Suppose we have a global parametrization defined. A *field-aligned T-mesh* is a mesh whose edges are curves on the surface satisfying two requirements: (a) each is a subset of a parametric line; (b) the

field is nonsingular on each face, except possibly at the corners.

Field-aligned edge moves. The most basic operation used by our algorithms is *moving* a T-edge or edge along the field of parametric lines. For a parametrization with no cuts, this corresponds to simple translation of the edge in the parametric plane. For each endpoint w of an edge e not located at a singularity, there is a unique parametric line $\ell(w)$ passing through w and orthogonal to the line of the edge in the parametric domain. We define a *valid move* to be a repositioning of the T-edge on the surface so that the new endpoints w_1' and w_2' are on $\ell(w_1)$ and $\ell(w_2)$, and move by the same amount along these lines, and the edge remains aligned with a parametric line. Furthermore, no singularity is contained in the curvilinear rectangle with corners w_1, w_2, w_2' and w_1'.

Attached sets of T-edges. For a given T-edge e, we call a T-edge e' attached to e, if their intersection contains at least one edge. If E is a set of edges, then $A(E)$ is the set of all edges attached to edges in E. The *attached set* $A^c(e)$ (Figure 7) is the transitive closure of A for an edge e. If a T-edge e is moved while maintaining alignment with the parametrization, all T-edges in $A^c(e)$ need to be moved by the same amount if no new faces in the T-mesh are created.

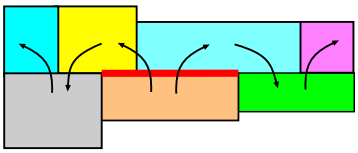


Figure 7: *The attached set of an edge (marked in red).*

In addition to simple attachment, we define *regular attachment*. An edge e' is regularly attached to e if it is attached to it, or it is on the same parametric line and shares a regular endpoint with e. Similarly, the regularly attached set of edges $RA^c(e)$ is defined as the transitive closure of the regular attachment relation.

Implementation. While all operations with T-edges can be implemented by tracing parametric lines on the surface, the implementation is considerably simplified by generating a fine (with quad size on the order of triangle size or smaller) quadrangulation of the original surface using the global parametrization. The downside of this approach is that it requires a moderate amount of rounding at the parametrization stage. In practice, we found that the quadrangulation can be chosen to be sufficiently fine for this not to lead to folds not removable by stiffening (see Section 7). In this case, the edge moves are no longer continuous but are discretized at the resolution of the fine quad mesh.

5.2 Initial T-mesh construction

As singularities of the field have to be vertices of the mesh, it is natural to start the construction using singularities as the initial set of vertices with no faces attached.

Singularity cell expansion. In the absence of parametric lines to align with, a natural and commonly used approach would be to use a Voronoi partition on the mesh to get a mesh of k-gons, and apply one step of Catmull-Clark subdivision. We mimic a simple Voronoi partition construction but force the edges of cell to be field-aligned. An initial curved k-gonal patch is defined by tracing integral lines of the parameterization gradient very close to singularity. (On the quad mesh, this tracing reduces to following edges along the fine quadrangulation in 1-neighborhood of the singularity.) If the singularity index is $i/4$, $i \leq 2$ then $k = 4 - i$. We refer to k as singularity valence. As a result we get a set of field-aligned edges, which are moved away from the singularity at a constant speed in parametric units, until the T-edges of the cell become attached to other T-edges and cannot be moved without shrinking other cells, or reach the mesh boundary.[1]

Unlike Voronoi cells, the field-aligned cells need not fill the whole mesh, leaving some *hole quads* uncovered. It is possible to show (see the Electronic Appendix) that the local configuration of edges at any hole quad is of the type shown in Figure 8a.

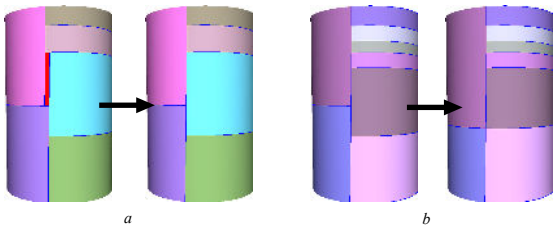


Figure 8: *(a) Closing a hole in the initial mesh; (b) regularization step*

Hole closing. Most of these holes can be eliminated by moving some of the cell boundaries. For the hole-elimination step, the hole quads are sorted by size, with small holes first. A single hole-closing operation, shown in Figure 8a, proceeds as follows. First, a pair (e_1, e_2) of opposite edges bounding the hole is selected. We select the pair of edges along the longer parametric dimension of the hole quad, with parametric distance d between them. The attachment sets $A^c(e_1)$ and $A^c(e_2)$ are either disjoint or coincide, as they are defined as transitive closures. In the latter case, we say that this pair of edges fails *the loop condition* (Figure 9), and consider the other pair of edges of the hole quad.

If the attachment sets are distinct, we determine the maximal valid move distances d_1^{max} and d_2^{max} for $A^c(e_1)$ and $A^c(e_2)$, in the direction towards the interior of the quad, defined as minima of the valid move distances of the edges in each set. If $d_1^{max} + d_2^{max} \geq d$, we set $d_1 = \min(d_1^{max}, d/2)$, and $d_2 = d - d_1$, and move the attached sets $A^c(e_1)$ and $A^c(e_2)$ to close the hole. If $d_1^{max} + d_2^{max} < d$, no valid move in this direction closes the hole, and we consider the opposite pair of edges. If neither pair can be used, we create an additional cell to fill the hole. The result of the initial T-mesh construction is a parametrization-aligned T-mesh, but with k-gonal faces.

Once all cells are expanded to the maximal extent, and all holes are filled, the k-gonal cells are split into parametrization-aligned quads, by tracing k parametric lines from the central singularity to the boundaries of cells. All three-valent vertices of the resulting mesh with two incident edges along the same parametric line become T-joints. If there are no holes in the mesh, the total number of cells in the mesh we obtain is $\sum_{\text{singularity } v}(4-4i_v)$, where i_v is the index of the singularity at v. The number of holes is bounded from above by the same number, as there is at most one hole at each k-gonal cell corner.

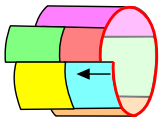


Figure 9: *Loop condition.*

T-mesh regularization. The regularization step reduces the number of T-joints in a mesh by an operation similar to closing holes, collapsing some edges separating two T-joints (Figure 8b). We find all edges in a mesh with two non-stem endpoints w_1 and w_2 at T-joints, such that the stem edges at these T-joints are on opposite sides of the edge. These edges are sorted by length, with shorter edges eliminated first. Let e_1 and e_2 be the stem edges at w_1 and w_2. Then we apply the same procedure as for the hole filling to the

[1] Note that while the shape of an initial cell of this type on a regular grid is identical to an L^∞ disk (i.e. a square), resulting cells are *not* Voronoi cells with respect to L^∞ metric: Our cells always have coordinate-aligned edges, while the L^∞ metric cells may have diagonal edges.

pair (e_1, e_2) except we use the regularly attached sets $RA^c(e_1)$ and $RA^c(e_2)$ instead of the attached sets, to avoid creating new T-joints in the process of removing old. A similar $RA^c(e_1) \neq RA^c(e_2)$ needs to be checked to verify validity of the move.

5.3 T-mesh optimization

The operations used in construction of the initial mesh are pure connectivity operations, not taking into account any quality criteria, other than reducing the number of T-joints at the regularization step. As the next step we optimize the patch layout. Our overall goal is to create the largest possible patches, while maintaining good patch quality. Our approach is similar to mesh simplification and improvement techniques: we define a set of operations on the set of faces of the T-mesh preserving the validity of mesh, and define an energy we want to minimize by a sequence of these operations, while satisfying a set of constraints.

Energy and constraints. We choose an energy favoring larger well-shaped patch sizes; for an individual patch, we use its perimeter-area ratio to balance the priorities of avoiding small patches while favoring square-shaped patches. Since we quadrangulate finely to minimize distortion due to rounding, we approximate actual lengths by parametric lengths. The energy of individual patches needs to be combined in a global energy function; we found that the ℓ_1 norm of the vector of parametric perimeter-area ratios yields the best results compared to ℓ_2 and ℓ_∞:

$$E_{area} = \sum_P \frac{1}{\mathcal{L}(P)} + \frac{1}{\mathcal{W}(P)} = \sum_P \frac{\mathcal{P}(P)}{2\mathcal{A}(P)}$$

where the summation is over patches P; and $\mathcal{L}$, $\mathcal{W}$, $\mathcal{P}$, and $\mathcal{A}$ are the parametric length, width, perimeter, and area operators, respectively. The constraints are as important as the energy itself: the T-mesh is likely to be useful in a much more limited context with no constraints on patch quality. The choice of constraints depends on the goal of constructing the T-mesh. If the primary goal is to partition the surface into a small number of logically rectangular domains, similarly to [Carr et al. 2006] or [Eppstein and Erickson 1999] the optimization can be done without constraints. If, however, we would like to use the T-mesh as a coarse control mesh for a high-order or multiresolution surface representation, controlling the approximation error and patch aspect ratios are likely to play an important role.

This leads us to our two main constraints (additional optional constraints, e.g. maximal patch size can be imposed if desired).

Geometric approximation constraint. For each face of the T-mesh, we estimate how well the surface can be approximated by a smooth piecewise polynomial surface on this patch. While we could use a globally smooth surface approximation directly (T-NURCCs or PT-splines), determining the precise approximation error is expensive as it requires solving a global linear system. Instead, we use a simple Bezier curve network fit to approximate the per-patch error locally and efficiently. For each face, we fit 8 Bezier curves (4 aligned with each parametric direction) to surface points uniformly sampled in the parametric domain (we use an 8×8 grid of samples). Each Bezier curve interpolates the boundary samples, so the fit reduces to solving a 2×2 system of linear equations per curve. We compute the approximation to L^2-norm of the error ε_P of the fit, as the sum

$$\varepsilon_P^2 \approx \frac{1}{2}\frac{\mathcal{A}(P)}{n^2} \sum_{i=0}^{3}\sum_{j=0}^{7} \left(\mathbf{b}_i(hj) - \mathbf{p}_{2i,j}\right)^2 + \sum_{j=0}^{3}\sum_{i=0}^{7} \left(\mathbf{b}_j(hi) - \mathbf{p}_{i,2j}\right)^2$$

where $\mathbf{p}_{ij}$ are the samples, and $\mathbf{b}_i(t)$ and $\mathbf{b}_j(s)$ are the Bezier curves along two parametric directions, with parameters t and s in the range $[0,1]$ on the face, and $h = 1/7$.

Patch aspect ratio constraint. While in many cases constraining geometric error results in automatic restrictions on the aspect ratio, the patches on nearly cylindrical areas of the surface may become very long. In other cases, the energy may favor creating very thin and narrow patches in flat areas to increase the area of nearby patches. To limit these effects we impose an additional constraint on the patch aspect ratio.

T-mesh modification operations. We use three operations for T-mesh modification: two connectivity-modifying, and one only affecting the patch size, two of which have a single length parameter.

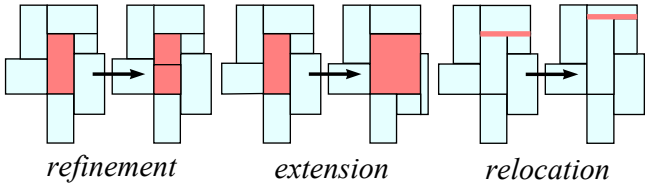


Figure 10: *Refinement, extension, and relocation operations*

The *Refinement* operation acts on a (T-edge,face) pair (e, f), splitting the face f into equal halves by inserting a new edge along the parametric direction perpendicular to e, generally creating two T-joints (Figure 10). Refinement always decreases the patch size, and increases the number of patches.

The *Extension* operation acts on a T-edge/face pair (e, f), extending the face f across the edge e into adjacent faces. It increases the size of one patch, while other patches shrink, or even eliminated. A face f (Figure 10) is extended to the maximal length possible across the T-edge e, so that we do not modify faces with no T-edges attached to the T-edge e. Depending on local connectivity, it may increase or decrease the number of patches, although it most commonly decreases the energy most when it reduces to a merge of several faces. We define this operation in a more general way, as we found that in some cases this less constrained operation produces better quality T-meshes.

The *Relocation* operation acts on an edge e. The attachment set $RA^c(e)$ is found and all edges in the set are moved by the same distance a, positive or negative, not exceeding the maximally valid distance in this direction.

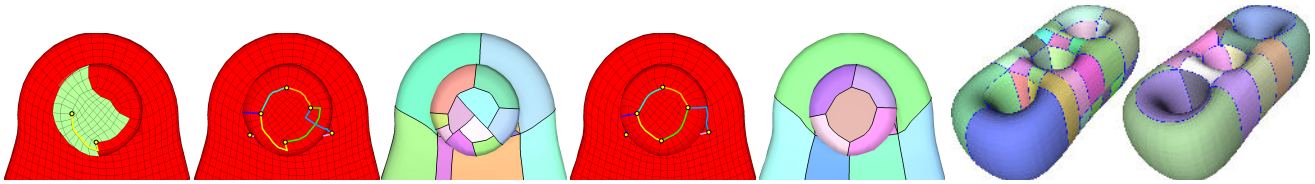
Complete optimization algorithm. We impose the constraints using a multiplicative penalty method (cf. [Torn and Zilinskas 1989]) by combining them with the energy function:

$$E_{total} = \sum_P \frac{\mathcal{P}}{2\mathcal{A}(P)} \left(1 + w(\alpha_P/\alpha_0 - 1)\right)\left(1 + w(\varepsilon_P/\varepsilon_0 - 1)\right)$$

where α_P is the parametric domain aspect ratio of patch P, ε_P is the geometric error estimate described above, α_0 and ε_0 are user-specified upper bounds for the constraints. The function $w(t)$ is chosen to be zero if $t < 0$, and increases rapidly for $t > 0$, we use t^3. Multiplicative penalty functions are similar to the more common additive penalties (taking log of the energy converts them to additive) but have the advantage of not requiring to choose a proper scale factor.

The complete optimization algorithm proceeds as follows.

For each (edge,face) pair of the T-mesh, we consider *Refinement* and *Extension* operations, and for each edge the *Relocation* operation. For each parametrized operation (*Extension* and *Relocation*) we determine the parameter range corresponding to valid moves, and find the parameter value corresponding to the maximal decrease in energy. Among all operations, we choose the operation that results in the maximal decrease in energy, and perform this operation. The process is iterated until the energy cannot be further decreased. Since the energy decreases at every step, the algorithm always terminates.



(a) (b) (c) (d) (e) (f) (g)

Figure 11: *Singularity alignment process: a: detecting a mismatch between adjacent singularities; b: a part of the singularity adjacency graph (observe "near-misses"); c: T-mesh before alignment; d: singularities and separating lines after alignment; e: T-mesh after alignment; f:* `holes3` *before alignment; g:* `holes3` *after alignment with no T-joints.*

Reevaluating all possible operations at every iteration would be prohibitively expensive. Instead, we update the invalidated operations incrementally.

We assign a timestamp to all edges and faces of the mesh (initially zero). In the beginning for all faces and edges, we generate all potential operations, compute the energy change resulting from each operation, and place the operations on the priority queue, with the energy decrease as the priority (energy-increasing operations are discarded). All operations are also given a timestep zero.

Then we repeatedly perform the operation with highest priority, unless the faces and edges it affects have a later timestamp than the operation, in which case it is discarded. All edges and faces modified as a result of the operation get a new timestamp t, and a new set of operations is generated for these facets and edges and pushed on the priority queue with timestamp t, if they decrease the energy.

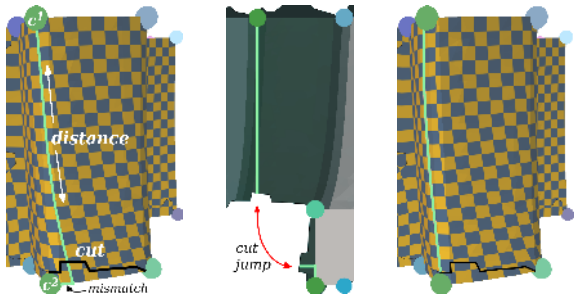
6 Singularity alignment

The T-mesh construction algorithm of Section 5 is limited by the fact that the parametrization is fixed, and patch boundaries stay aligned with parametric lines. However, only important feature lines (sharp edges in particular) are fixed by the geometry. The cross-field and the parametric lines of the global parameterization away from features can be modified to improve the quality of the mesh, and decrease the number of T-joints. The problem of separating lines passing within short distance of each other, (discussed in Section 3; Figure 2, right), can be reduced by adjusting the global parameterization.

Overview. We observe that ideally we want the separating lines starting at a singularity to terminate at a nearby singularity, if it passes sufficiently close to it. The algorithm that we describe in this section identifies close singularities and detects "near-misses", constructs the *singularity adjacency graph*, and solves for a new parametrization with additional constraints that force perfect alignment for identified pairs; the steps of the process are shown in Figure 11.

Defining a singularity adjacency graph. We observe that the first step of the initial T-mesh construction algorithm, with minor modifications, provides a mechanism for detecting "near-misses" of field separating lines. As before, we expand a cell from each singularity. However, instead of growing all cells at once, we expand the cell, until each edge reaches an adjacent singularity, a boundary, or another edge of the same cell. Singularities on the boundary of the maximally expanded cell for a singularity c are considered adjacent to c. This relation is not necessarily reciprocal. We construct a singularity adjacency graph connecting by edges all adjacent singularities. Example graphs are shown in Figure 11 and 18. Each edge (c^1, c^2) is annotated with a separating line *mismatch* (the parametric length from the singularity c^2 to the closest parametric line starting at c^1) and *distance* (the parametric L^∞ distance between the singularities) illustrated in Figure 12.

We want to change the parametrization so that a parametric line starting at c^1 passes through c^2 i.e., so that the mismatch becomes zero at a maximal number of edges of the adjacency graph.



misaligned pair *parametric domain* *after alignment*

Figure 12: *A path connecting two misaligned singularities before and after alignment.*

Constructing constraints. If a mesh can be parametrized without seams, the requirement of singularity alignment easily translates into a constraint on parametrization: two singularities should be on the same parametric line, i.e. share the same u or v value. The constrained optimization framework of [Bommes et al. 2009] that we are using makes adding such constraints easy.

For parametrizations with cuts, the situation is more complicated. A parametric line on the surface undergoes a jump to a different point and direction in the parametric space when it crosses a cut (Figure 12). While the rotation is entirely determined by the cross-field to which the parametrization is aligned, the positional jump depends on the parametrization itself. When we add a constraint on singularity coordinates, the parametrization may change, changing the jumps at cut edges. The resulting constraint will depend not only on the pair of singularities (c^1,c^2), but also on the (variable) translational parts of the transforms at the cut edges we cross, t_e in Section 4. E.g., if the cut is crossed once, and the crossing is at a cut edge e with associated transform $p' = R_e p + t_e$, where $p = (u, v)$ is a parametric point, then the constraint is $(R_e p^1 + t_e)_u = p^2_u$, if the aligned parameter line at c^2 is along the u coordinate direction, and p^1 and p^2 are parametric positions of c^1 and c^2), and the subscript u means taking the u coordinate.

In the general case, consider a path crossing cut edges e_i $i = 0, \dots m$ between c^1 and c^2. Assuming the final direction of the path is u, then the complete constraint has the form

$$\left((\Pi_{i=0}^{m} R_{m-i}) p^1 + \sum_{i=1}^{m} \Pi_{j=i}^{m} R_{m-j} t_i \right)_u = p^2_u$$

we still have a single linear constraint, but involving a larger number of variables t_i, p^1 and p^2 (Figure 13).

We observe that the form of the constraint depends on the choice of path between singularities. One can show that for

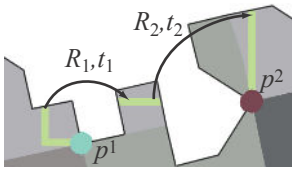


Figure 13: *Forming a constraint for a pair of singularities.*

two paths P_1 and P_2 connecting two singularities and such that the loop formed by P_1 and P_2 does not enclose any singularities and encloses a topological disk, the constraints are equivalent (see the Electronic Appendix, Proposition 1)[2] This allows us to choose a path between singularities consisting of two segments of parametric lines, one passing through c_1 and the other through c_2, tracing the boundary of the rectangle with c_1 and c_2 at diagonal corners.

Filtering singularity constraints. The singularity constraints can be redundant. As these are homogeneous constraints with zero right-hand side, they cannot be incompatible, so we eliminate the redundant ones with Gaussian elimination.

We choose a threshold for the minimal aspect ratio of the rectangle for an adjacent pair of singularities, and remove all constraints exceeding this threshold (we set it in the range 5-10); choosing this threshold high enough ensures that the field does not deviate too far from soft creases. A singularity can satisfy only a single constraint along each outgoing parametric line, so if several constraints correspond to a direction, we choose the one with the closest singularity (smallest parametric L^∞ distance). The singularity constraints can interact with sharp edge constraints: if a singularity has a sharp edge constraint in a particular parametric direction, we remove singularity constraints in this direction.

7 Results and Comparisons

Our algorithm is automatic once various thresholds are set. Other than mixed integer quadrangulation parameters, we use (1) aspect ratio threshold for cone alignment pair selection, (2) desired max aspect ratio of T-patches, and (3) max geometric approximation error. Optionally, manual adjustments can be made to singularity placements as described in [Bommes et al. 2009]. This was done for the `maxplanck` head to make the placement of singularities more symmetric.

We evaluate the results of our algorithm in several ways. The main criterion is the number of patches in the final mesh subject to the constraints on aspect ratio and geometric approximation quality.

For several meshes, we compare domain T-meshes we obtain to the minimal number of patches we could obtain in a conforming mesh using mixed-integer quadrangulation, with the same feature field and the number of stiffening iterations bounded by 40. We observe that in most cases, especially involving alignment, the main restriction on quad size is due to the lack of robustness with respect to large rounding. Numerical data on the number of faces in different meshes are presented in Table 1, and Figures 16 and 15 show the results for several meshes. We emphasize that these coarsest meshes are not necessarily suitable as control meshes for a T-spline or other fit. Rather, these are useful as seamlessly and smoothly aligned rectangular geometry images, for texturing, or solving equations on the surface.

Periodic Global Parametrization (PGP) [Ray et al. 2006], does not have a target quad size limitation due to foldovers PGP avoids the need for rounding, and the need for stiffening, but as the target quad size becomes coarser, the quality and alignment of the mesh rapidly deteriorates. Figure 14 shows a parametrization obtained using PGP with approximately the same number of quads as our T-mesh shown in Figure 16. (Caveat: the best effort was made to smooth the field for the PGP parametrization, but it is unclear if the quality of the field matched the quality of the cross-field we used for our result.)

For high-order approximation, the geometric error constraint has to be taken into account. Figure 17 shows the effect of decreasing the geometric error constraint for one model and decreasing the aspect ratio constraint.

In Figure 18 we show the full singularity adjacency graph and its pruning. Singularity alignment is highly useful for eliminating most of the near misses in matches. At the same time, we observe that one cannot expect to eliminate most of the T-joints in the mesh using this method due to two reasons. First, we filter the singularity alignment constraints by feasibility and by the mismatch-to-distance ratio, to avoid deviation from the feature field. As a result, the total number of valid alignments we enforce is relatively small, compared to the total number of T-joints. Second, the resulting mesh is optimized using our general T-mesh optimization procedure, which has to trade T-joint creation for better geometric approximation or aspect ratio.

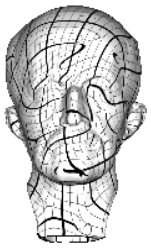


Figure 14: *PGP parameterization with target size chosen to match our number of patches in Figure 16*

We observe that the number of control points in the T-mesh and the number of patches is within a factor of 2-4 of the number of singularities in the original mesh. We believe that with constraints imposed on patches it is difficult to improve on these numbers: the best possible number one can expect is approximately equal to the number of singularities. In this case, every quad of the mesh is supposed to have corners at singularities, which is extremely difficult to achieve for typically highly nonuniform singularity locations produced by feature-aligned fields.

Fitting T-splines. Once the T-mesh is constructed it can be used to define a T-spline/T-NURCCs surface which can be fitted to the original mesh. Our approach to fitting T-spline surfaces is identical to that of [Li et al. 2006], with two important differences. Because our T-mesh is constructed by tracing parametrization lines, *the sums of parametric intervals on the opposite sides of each face are guaranteed to be equal* if we simply use parametric length to determine the knot interval for each edge. This eliminates the need to introduce the extraordinary vertices at T-joints which were necessary to handle the T-meshes constructed using PGP.

The second difference is that we pass the information about sharp edges to the T-spline construction, and insert degenerate faces with zero knot intervals along sharp edges.

We obtain meshes suitable for fitting in our framework by setting the geometric error to 0.02 of the model diameter. Figure 19 shows the fit for several models. The L_2/L_∞ relative errors for the models shown in the figure are: for `joint`, 0.08%/2%, for `sculpt`, 0.1%/2%, for `botijo`, 0.1%/1% and for `fertility`, 0.06%/0.5%.

Performance. Performance numbers for different stages of the process are included in Table 1. The time needed for construction of the initial T-mesh patch layout is negligible in all cases and we ignore it. For larger meshes, in general, the dominant cost is mixed-integer field optimization (vertex-based version is more expensive than facet, and the running time of mixed-integer optimization increases faster than linearly due to correlation in the number of vertices and number of integer variables). This potentially may be alleviated by rounding in groups or the method of recent paper of Ray and Levy [Ray et al. 2009]. For some meshes, the parametrization cost dominates, because of a large number of stiffening iterations. For smaller meshes, the cost of field optimization is far lower, and

[2]Because the singularities are located on the cut by construction, there is an ambiguity in determining the crossed cut edges for the paths P_i. This ambiguity is resolved by defining the starting and ending point of each path as infinitesimally displaced from the singularities along the parametric line towards the other singularity.

					unaligned						sing.		aligned with a.r. constraint				aligned with geom. constraint			
Model	faces	N_q	N_s	time$_{fld}$	niter	time/iter	N_Q	N_P	N_v	T-joints	pairs	N_Q	N_P	time$_P$	N_v	T-joints	N_P	time$_P$	N_v	T-joints
holes3	11776	154	16	25.1	1	0.590	15943	47	69	52	29	13211	20	0.173	16	0	232	53.5	401	346
sculpt	7342	1255	16	9.00	2	0.295	11553	65	106	82	24	10792	38	0.725	58	36	104	12.9	146	88
joint	8766	776	23	8.85	2	0.629	13715	51	70	42	23	13440	45	0.701	59	32	68	9.17	100	68
fandisk	14454	4185	32	27.1	8	1.02	54421	63	93	56	6	51897	50	1.31	71	38	153	15.3	193	76
casting	36828	47845	98	250	7	12.1	756338	272	403	294	51	745319	233	70.7	326	218	297	114	429	296
screw	9596	1034	10	10.8	4	1.36	13384	40	74	60	4	14353	18	6.72	30	20	182	15.4	235	102
rockerarm	20088	1076	24	49.7	4	1.50	16311	87	141	108	24	16129	54	1.58	79	50	195	73.3	276	162
screwdriver	54300	906	20	352	11	12.94	5445	55	94	74	18	5306	40	1.59	62	40	118	41.5	165	90
maxplanck	50790	2198	15	555	9	1.82	36725	57	96	72	12	36725	50	2.92	84	60	671	232	1224	1086
bojito	82332	1837	72	960	6	3.02	32974	351	523	360	94	31987	165	8.98	230	146	403	76.7	554	318
fertility	27954	914	43	152	8	1.05	14211	172	265	194	39	14211	146	1.31	247	188	420	67.5	581	334
elephant	49918	6395	96	468	31	5.48	59342	396	635	486	44	57762	281	4.44	449	344	366	42.6	587	450

Table 1: *Column titles: N_q is the number of quads in the coarsest quadrangulation; N_s is the number of singularities; time$_{fld}$ is the time to smooth the cross-field; for parameterization, niter is the number of stiffening iterations and time/niter is the average time taken per iteration; N_Q is the number of quads in the fine quadrangulation used for patch generation; N_P is the number of patches; time$_P$ is the time to generate the T-mesh; N_v is the number of vertices in the T-mesh.*

T-mesh layout may be the most significant expense. The cost of T-mesh optimization strongly depends on the number of patches produced, so it raises substantially, when lots of small patches are required. For example, for the `maxplanck` mesh, which has a lot of geometric detail at different scales, a very large number of patches is needed to obtain the same error. For this type of meshes, direct high-order patch approximation is not appropriate, and a displacement map or a hierarchical approximation is needed so that fewer patches can be used. The cost also raises if a very fine quad mesh is used (as it was necessary for `casting`). This is not a fundamental problem of the method, and it is primarily due to a suboptimal implementation of searching for an optimal parameter values for the parametrized optimization. We note that this numbers are strongly implementation, compiler and hardware dependent: for example, in our setup we were unable to match field optimization timings presented in [Bommes et al. 2009], although the same MI code was used. The T-mesh optimization algorithm complexity is hard to estimate theoretically, so its scaling is difficult to predict. We have observed that it does not depend much on the number of singularities, but has a stronger dependence on geometric complexity of the shape (as more complex shapes require finer patches).

Limitations. There are several important limitations to our method.

Field quality. To the greatest extent, the quality of the T-mesh is determined by the quality of the field. Current techniques do not allow automatic enforcement of symmetries, and in many cases lead to unnecessary bending and twisting of patch boundaries. (often observed for organic shapes lacking sharp edge alignment constraints).

In an effort to eliminate unnecessary T-joints, our technique for singularity alignment changes parametric placement of singularities, but does not alter their position on the surface. Adjusting singularity positions whenever possible (if a singularity is located in an isotropic flat area) has potential for considerably improving the quality of the layouts.

Optimality. There is no guarantee that our result is close to the global optimum. For smaller meshes, the results seem hard to improve. For more complex meshes better quality seems possible. In all cases, a significant reduction in energy was achieved.

Robustness. There are three limiting factors in robustness of our method. The most substantial restriction is the need to obtain a locally one-to-one parametrization. Even with no rounding, for complex models with sharp features, foldovers in the parametrization are common, and stiffening iterations are not guaranteed to eliminate these. Another problem is detection of sharp features. In this work, we relied on the ability to tag sharp features based on the dihedral angles, or on having relatively rounded features. Finally, the global impact of sharp edges and cone alignment is difficult to predict as these constraints could force the parameterization to collapse due to global dependencies. The T-mesh construction is quite robust, but can produce excessive number of small patches for low-quality input fields.

Scalability. While we were able to obtain parametrizations for meshes of moderate size (approximately 100 thousand triangles), the main scalability bottleneck is the field optimization. The running time of the mixed-integer solver for larger meshes is dominated by the Gaussian elimination step for the constraints. Furthermore the number of iterations of the solver is proportional to the number of singularities, which, in turn, grows with geometric complexity of the model.

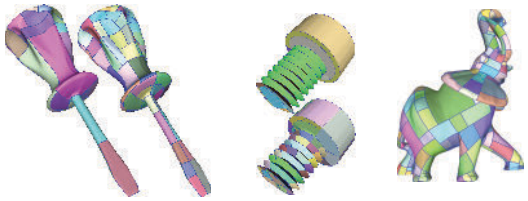


Figure 15: *The* `screw` *and* `screwdriver` *with patches of maximal size, and obeying geometric error constraints; the* `elephant` *with geometric error optimization.*

8 Conclusions

The method presented in this paper demonstrates the possibility of constructing coarse domain meshes while maintaining feature alignment, if T-joints are allowed in the mesh. Domain meshes constructed in this way are a natural fit for T-spline surfaces and related high-order constructions. Clearly, our T-mesh construction algorithm can be improved and extended in many ways to achieve more compact T-mesh structures with fewer nonessential T-joints. Furthermore, we observe that in many ways the quality of patch layouts is determined by the initial cross-field, and improving the quality of these fields is an important direction for future work.

While many algorithms and constructions for conforming meshes can be easily extended to T-meshes, overall, the theory and algorithms for this type of meshes is far less developed, presenting many interesting questions for future research.

Acknowledgements We thank Evangelos Kalogerakis et al. for making available their software for robust curvature computation; Henrik Zimmer and David Bommes for their library and clarifications on the working of their mixed integer programming library; T-splines Inc., Tom Finnigan and Adam Helps for providing us with their T-spline library and support.

The research was, in part, supported by the NSF awards IIS-0905502 and DMS-0602235 and the EG 7FP IP "3D-COFORM project (2008-2012, n. 231809)".

References

BAZILEVS, Y., CALO, V., COTTRELL, J., EVANS, J., HUGHES, T., LIPTON, S., SCOTT, M., AND SEDERBERG, T. 2009. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*.

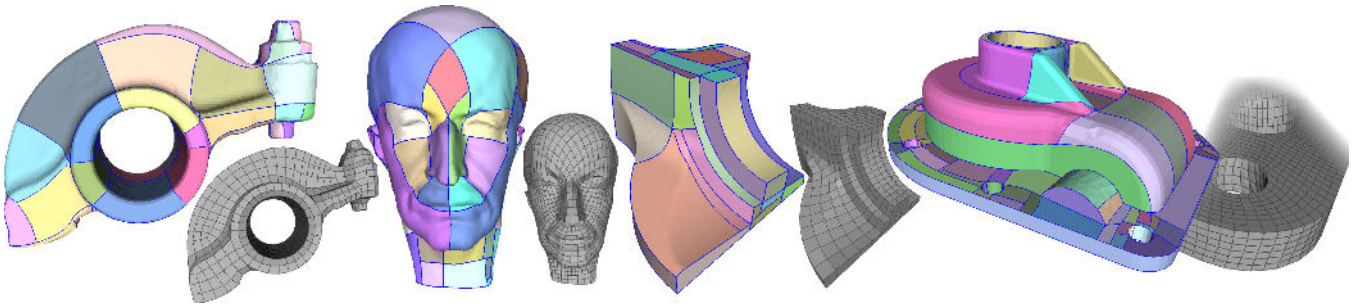


Figure 16: *Maximal patch sizes obtained by our algorithm while maintaining aspect ratio constraint 2.5. From left to right:* `rockerarm`, `fandisk`, `maxplanck`, `casting`. *Smaller images show the coarsest quad meshes we could obtain.*

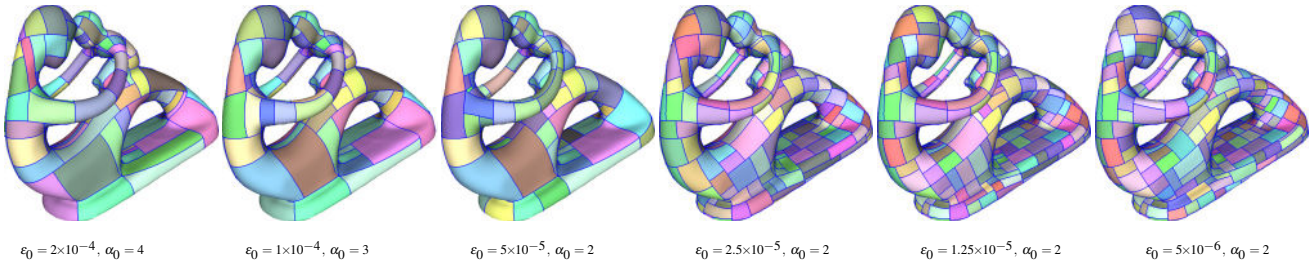


Figure 17: *Effects of changing the minimum thresholds for geometric error ε_0 and aspect ratio α_0 for the* `fertility` *mesh. Geometric error is measured relatively to the diameter of the bounding box of the mesh.*

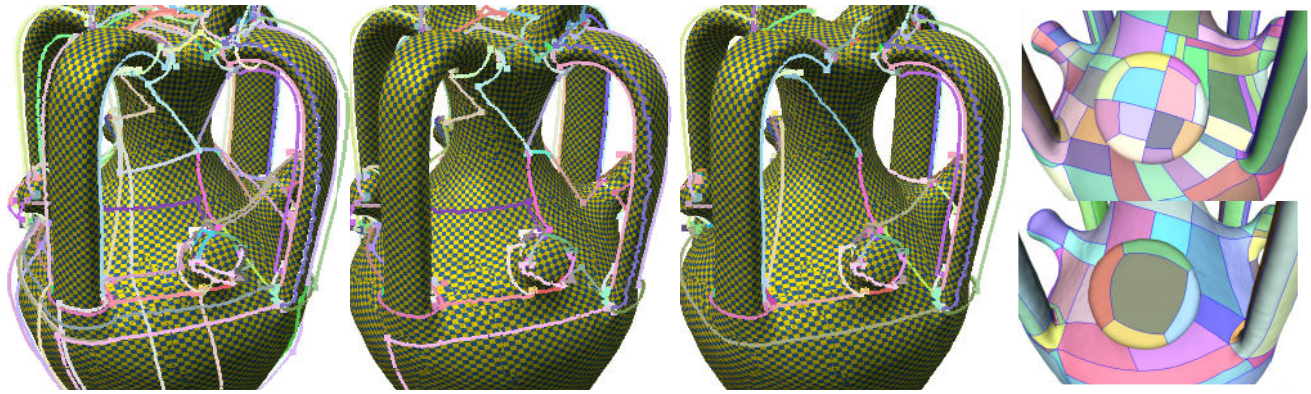


Figure 18: *From left to right: a full singularity adjacency graph with 209 edges for 72 singularities for the* `botijo` *mesh; adjacency graph pruned by compatibility criterion; 120 edges; adjacency graph pruned by aspect ratio 3 followed by compatibility, 94 edges; Comparison of aligned and non-aligned T-meshes.*

Figure 19: *Several T-spline models obtained by least-squares fitting from T-meshes generated by our algorithm:* `fertility`, `rockerarm`, `sculpt`.

BEN-CHEN, M., GOTSMAN, C., AND BUNIN, G. 2008. Conformal Flattening by Curvature Prescription and Metric Scaling. In *Computer Graphics Forum*, vol. 27, Blackwell Synergy, 449–458.

BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph. 28*, 3, 77.

CARR, N., HOBEROCK, J., CRANE, K., AND HART, J. 2006. Rectangular multi-chart geometry images. In *Symposium on Geometry Processing*, Eurographics Association, 190.

DANIELS, J., SILVA, C., AND COHEN, E. 2009. Semi-regular quadrilateral-only remeshing from simplified base domains. In *Computer Graphics Forum*, vol. 28, Blackwell Publishing Ltd, 1427–1435.

DANIELS, J., SILVA, C. T., AND COHEN, E. 2009. Localized quadrilateral coarsening. *Computer Graphics Forum 28*, 5, 1437–1444.

DENG, J., CHEN, F., LI, X., HU, C., TONG, W., YANG, Z., AND FENG, Y. 2008. Polynomial splines over hierarchical T-meshes. *Graphical Models 70*, 4, 76–86.

DONG, S., BREMER, P., GARLAND, M., PASCUCCI, V., AND HART, J. 2006. Spectral surface quadrangulation. *ACM Trans. Graph. 25*, 3, 1057–1066.

ECK, M., DEROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., AND STUETZLE, W. 1995. Multiresolution analysis of arbitrary meshes. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 173–182.

EPPSTEIN, D., AND ERICKSON, J. 1999. Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions. *Discrete and Computational Geometry 22*, 4, 569–592.

GU, X., AND YAU, S. 2003. Global conformal surface parameterization. *Symposium on Geometry Processing*, 127–137.

HE, Y., WANG, K., WANG, H., GU, X., AND QIN, H. 2006. Manifold T-spline. *Lecture Notes in Computer Science 4077*, 409.

HERTZMANN, A., AND ZORIN, D. 2000. Illustrating smooth surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 517–526.

HILDEBRANDT, K., POLTHIER, K., AND WARDETZKY, M. 2005. Smooth feature lines on surface meshes. In *Symposium on Geometry Processing*, Eurographics Association, 85.

HORMANN, K., LÉVY, B., AND SHEFFER, A. 2007. Mesh parameterization: Theory and practice. *SIGGRAPH Course Notes*.

HUANG, J., ZHANG, M., MA, J., LIU, X., KOBBELT, L., AND BAO, H. 2008. Spectral quadrangulation with orientation and alignment control. In *International Conference on Computer Graphics and Interactive Techniques*, ACM New York, NY, USA.

KÄLBERER, F., NIESER, M., AND POLTHIER, K. 2007. QuadCover: Surface Parameterization using Branched Coverings. *Computer Graphics Forum 26*, 3, 375–384.

KALOGERAKIS, E., SIMARI, P., NOWROUZEZAHRAI, D., AND SINGH, K. 2007. Robust statistical estimation of curvature on discretized surfaces. In *Symposium on Geometry Processing*, 13–22.

KHODAKOVSKY, A., LITKE, N., AND SCHRÖDER, P. 2003. Globally smooth parameterizations with low distortion. *ACM Trans. Graph. 22*, 3, 350–357.

KOVACS, D., MYLES, A., AND ZORIN, D. 2009. Anisotropic harmonic quadrangulation. In *Symposium on Geometry Processing 2009 Poster*.

LEE, A., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. 1998. MAPS: multiresolution adaptive parameterization of surfaces. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM New York, NY, USA, 95–104.

LI, W., RAY, N., AND LÉVY, B. 2006. Automatic and interactive mesh to T-spline conversion. In *Symposium on Geometry Processing*, Eurographics Association, 200.

LI, X., DENG, J., AND CHEN, F. 2007. Surface modeling with polynomial splines over hierarchical T-meshes. *The Visual Computer 23*, 12, 1027–1033.

LI, X., DENG, J., AND CHEN, F. 2009. Polynomial splines over general T-meshes. *The Visual Computer*, 1–10.

MARINOV, M., AND KOBBELT, L. 2005. Automatic generation of structure preserving multiresolution models. In *Computer Graphics Forum*, vol. 24, Amsterdam: North Holland, 1982-, 479–486.

OHTAKE, Y., BELYAEV, A., AND SEIDEL, H. 2004. Ridge-valley lines on meshes via implicit surface fitting. In *International Conference on Computer Graphics and Interactive Techniques*, ACM New York, NY, USA, 609–612.

PALACIOS, J., AND ZHANG, E. 2007. Rotational symmetry field design on surfaces. *ACM Trans. Graph. 26*, 3, 55.

PIETRONI, N., TARINI, M., AND CIGNONI, P. 2009. Almost isometric mesh parameterization through abstract domains. *IEEE Transactions on Visualization and Computer Graphics 99*, RapidPosts.

RAY, N., LI, W., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph. 25*, 4, 1460–1485.

RAY, N., VALLET, B., LI, W., AND LÉVY, B. 2008. N-Symmetry Direction Field Design. *ACM Trans. Graph. 27*, 2.

RAY, N., VALLET, B., ALONSO, L., AND LEVY, B. 2009. Geometry-aware direction field processing. *ACM Trans. Graph. 29*, 1, 1–11.

SEDERBERG, T., ZHENG, J., BAKENOV, A., AND NASRI, A. 2003. T-splines and T-NURCCs. In *ACM SIGGRAPH 2003 Papers*, ACM, 484.

SEDERBERG, T., CARDON, D., FINNIGAN, G., NORTH, N., ZHENG, J., AND LYCHE, T. 2004. T-spline simplification and local refinement. *ACM Trans. Graph. 23*, 3, 276–283.

SHEFFER, A., PRAUN, E., AND ROSE, K. 2006. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision 2*, 2, 171.

SPRINGBORN, B., SCHRÖDER, P., AND PINKALL, U. 2008. Conformal equivalence of triangle meshes.

TARINI, M., PIETRONI, N., CIGNONI, P., PANOZZO, D., AND PUPPO, E. 2010. Practical quad mesh simplification. *Computer Graphics Forum 29*, 2.

TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. *Symposium on Geometry Processing*, 201–210.

TORN, A., AND ZILINSKAS, A. 1989. Global Optimization, volume 350 of. *Lecture Notes in Computer Science*.

WEINKAUF, T., AND GÜNTHER, D. 2009. Separatrix Persistence: Extraction of Salient Edges on Surfaces Using Topological Methods. In *Computer Graphics Forum*, vol. 28, Blackwell Publishing Ltd, 1519–1528.