Mesh Extension for Labeled Surface Fitting

M. Vaitkus and T. Várady

Budapest University of Technology and Economics, Budapest, Hungary

Abstract

Approximating data points of an irregular, trimmed triangular mesh by tensor-product surfaces, such as NURBS, is an important task in computer-aided geometric design, as well as computer graphics. One crucial issue is the parameterization of the data points, that will have a strong influence on the quality of the surface to be fitted. Another important issue is how to avoid weak control points, that are not constrained by the data points and might lead to numerically unstable, wildly oscillating results.

In a recent paper,[11] current authors proposed techniques that facilitate tensor-product fitting. First, a 2D parameterization of the data points is computed, based on labels assigned to some of the boundary segments, and a corresponding virtual guiding frame. Second, the surface is extended in 3D, i.e. after adding supplementary data points to fill the domain rectangle, the data points are mapped back to optimized positions in 3D. In this paper, after giving an overview of these techniques, we focus on the extension step and demonstrate its effect using a few examples.

1. Introduction

The approximation of data points or triangle meshes by tensor-product parametric surfaces, such as B-splines or NURBS, is an important task in geometric modeling and reverse engineering. Surface fitting is a complex problem that depends on various parameters, including tolerances, knot vectors, smoothing terms and many others. In order to formulate surface fitting as a linear least-squares problem, (u, v) parameter values must be assigned to the data points. In particular, we aim to approximate *trimmed regions* of irregular triangle meshes that are bounded by a multi-sided loop of boundary segments, in which case we have no explicit information about the orientation of the four boundaries to be created, i.e the parameterization of the surface, and its surface geometry beyond the trimmed region. There are infinitely many tensor-product surfaces that approximate a set of given data points – we aim to produce surfaces that are well-suited for CAGD applications.

In a recent paper[11] current authors proposed two useful "preprocessing" techniques, that facilitate the fitting of high-quality surfaces. First, a constrained 2D *parameterization* of the data points is computed, based on *labels* assigned to some of the boundary segments, and a corresponding *virtual guiding frame*.

The second subsequent technique is called surface *extension*. After adding *supplementary data points* to fill the domain rectangle, an extended triangulation is created, and the data points are mapped back to optimized positions in 3D. In this way a four-sided mesh is obtained, that smoothly extrapolates the original trimmed region and is ready to be approximated by a tensor-product surface. The extension of the trimmed regions is critical for the stability of surface fitting. Without extension, the support of certain basis functions might contain very few data points or none at all. Consequently the location of the corresponding *weak control points* will be unconstrained, and the solution of the related badly conditioned linear system might yield a surface with unpredictable oscillations beyond the trimmed regions.

In this paper, we give a brief summary of these methods, with a particular focus on guiding frame computation and mesh extension. Due to spatial limitations, we omit certain technical details and precise formulas – for these, the reader is referred to the Appendix of.[11]

Our paper is structured as follows. After a brief overview on previous research in section 2, we discuss the concept of labeling to orient the surface to be fitted and the workflow of our approach (section 3). We describe our method for constrained parameterization, including the computation of guiding frames in section 4. The extension of meshes in 3D is the topic of section 5. Finally, we demonstrate the effectiveness of labeling and extension on a few test cases in. section 6

2. Previous Work

The literature on tensor-product fitting algorithms is vast, we refer to the works[12,13] and the references therein. See [Ref. 11, Sec. 2] for a more thorough survey.

2.1. Handling Weak Control Points

The problem of weak control points (or overfitting) in the context of trimmed tensor-products have been studied in depth by.[13] There, a heuristic solution has been proposed, which constrained the weak control points to remain in the vicinity of a previous, loosely fitted surface.

When fitting with the usual least-squares energy, weak control points lead to a linear system with a large condition number, and thus to numerically unstable results. The same problem arises when tensor-product splines are used as basis functions for numerical analysis over trimmed geometries – see the recent survey[9] and the references therein. In the context of analysis, the instability of trimmed splines can be handled using a non-standard basis constructed by weighting or extrapolating the untrimmed functions, as discussed in e.g.[4]

2.2. Extrapolating surface meshes

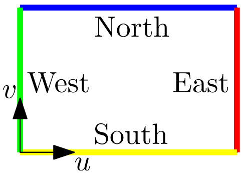
Smooth extrapolation of surface meshes beyond their boundaries is a well-researched area, see[1,3] and the references therein. Our approach is most closely related to variational and parameterization-based techniques, such as.[2,5,7]

3. Preliminaries

Our input comprises the following set of data:

- a manifold triangle mesh M (possibly multiply connected),
- a segmentation of the perimeter boundary loop of M into a sequence of polylines,
- a labeling assigned to the boundary segments of M, as explained below.

3.1. Labeling



A central element of our work is the concept of *labeling*, which is a powerful technique to orient the yet-unknown surface. This makes it possible to accomplish certain requirements and narrow down the set of infinitely many parameterizations. Some of the input boundary segments can be labeled – using our notations – as *North*, *West*, *South*, *East*, prescribing that a segment needs to lie somewhere on the boundary of the surface to be fitted. In other words, we prescribe that a segment needs to be mapped to a particular side of the domain rectangle (see inset figure). Other segments may remain *Unlabeled*. It is not necessary that all four label types are used, and the same label may be attached to more than one segment. Note that the exact location and length of the labeled segments on the corresponding sides of the rectangle remain undefined and will only be determined later by constrained parameterization.

Labeling determines where a trimmed region needs to be extended, and this simultaneously influences the number of weak control points and thus the stability of the final fit. We are not aware of any abstract characterization between the geometry of a surface and its ideal parameterization, nevertheless, it is intuitively clear that by a better orientation of the surface we can reduce the "weak" areas in the parametric domain.

The labeling of our test examples have been produced manually and developing algorithms for automatic labeling is subject of our future research. In the forthcoming sections we will exploit the labeling information, but disregard its meaning and origin.

3.2. Workflow

In this section we summarize the basic phases of the workflow we proposed in[11] for evolving a trimmed triangular mesh into a four-sided, fully parameterized mesh ready for tensor-product surface fitting. The workflow is depicted in Figure 1.

Phase I: Mapping from 3D to 2D

First, we parameterize the data points of the given region utilizing the label information, details will be discussed in section 4. We first compute a 3D *guiding frame*, as a rough approximation of the four boundaries of the surface. It approximates the labeled segments, and comprises missing boundary pieces and corners, where needed. The guiding frame is used to set the location and the relative proportion of the labeled segments. Next, a *constrained parameterization* is computed by assigning u, v coordinates to each data point, so that geometric distortion is minimized, and the constraints obtained from the guiding frame are satisfied.

Phase II: Mapping from 2D to 3D

Having the data points parameterized, we can apply least-squares fitting. As discussed earlier, the presence of weak control points might lead to numerical instabilities; this can only be alleviated with excessive smoothing/regularization. To avoid this problem we extend the original data into a four-sided mesh, that can be fitted directly, permitting tight tolerance within the region and no wiggling beyond its boundaries. We illustrate the stabilizing effect of mesh extension

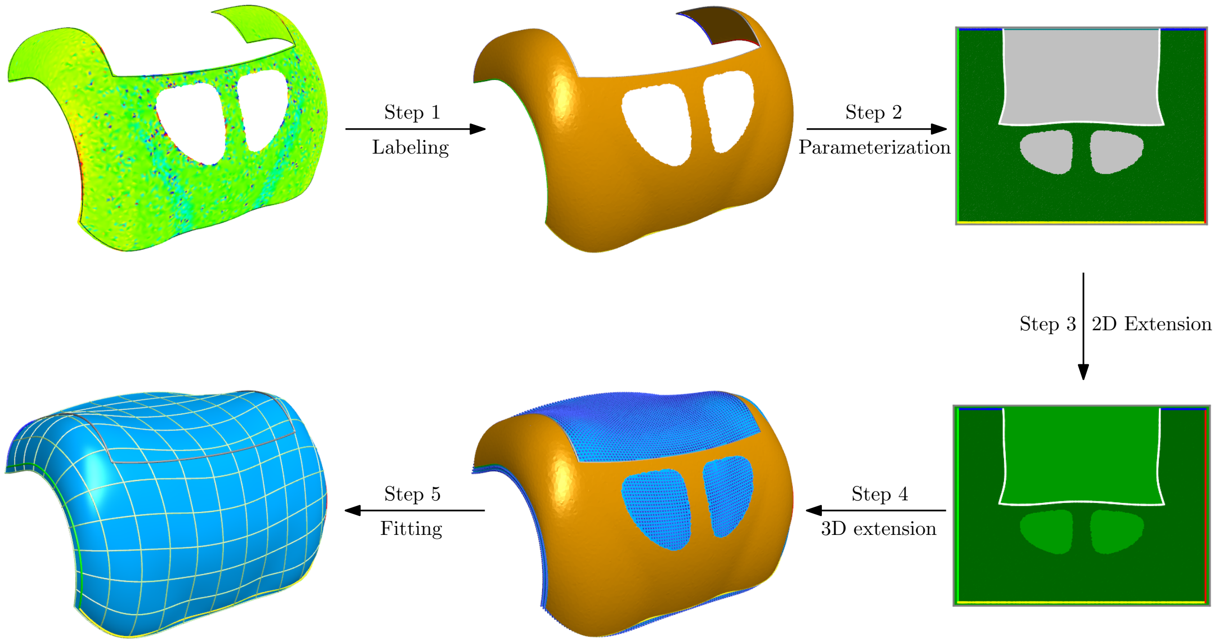


Figure 1: Workflow

in Figure 4. Note that this phase can be beneficial even when no labels and guiding frame have been supplied in the previous phase. First, the already parameterized input data points get supplemented by new, "artificial" data points so that they fill in the entire domain. Often the default bounding rectangle is further enlarged to provide margins for the surface to be fitted. The supplementary points are then triangulated yielding an extended parameterized mesh in 2D. Then the 3D locations of the supplementary data points need to be determined, as well, so that the extended mesh smoothly extrapolates the original trimmed region and forms a four-sided parameterized mesh. Details are discussed in section 5.

Finally the extended mesh is approximated by a tensor-product surface. Here we make no assumption on the applied fitting algorithm, nevertheless, we believe that our proposed techniques potentially benefit any tensor-product fitting procedure.

4. Mapping from 3D to 2D

In the first phase we map the input triangle mesh to the inside of a yet-unknown rectangular domain within the (u,v) plane: $f : (x_i, y_i, z_i) \mapsto (u_i, v_i)$.

We minimize the geometric distortion of the flattening and make use of labeling information, when it is available, by mapping the labeled boundary segments onto sides of the domain rectangle. We do not wish to prescribe the position and proportion of the domain rectangle, or where the labeled segments are located along the sides.

A naive approach that simply constrains the labeled boundary segments to u or v isolines is generally not sufficient to produce an acceptable parameterization. Generally speaking, it is important that the 2D mesh possesses similar relative magnitudes as the 3D original. In particular, we aim to preserve the relative lengths of the labeled segments within the corresponding surface boundaries. However, as the untrimmed boundaries are not known *a priori*, we have to make a rough approximation of them based on the available data – which is the topic of the next subsection.

4.1. Guiding Frames

Our goal is to infer the boundary curves of the untrimmed patch. In general, we have to estimate arcs that connect the label segments, and the locations of indefinite corners.

Arcs between consecutive label segments are computed by interpolating the endpoints and tangents of the two curves with a *cubic Hermite spline*. Setting the tangent lengths equal to the Euclidean distance between the segment endpoints gives good results in practice.

It is more challenging to find *corners* of the patch by extending the labeled boundaries. To make such a guess without fitting the entire surface, we propose a general method that computes the *guiding frame* – a collection of free-form curves in 3D that roughly approximates the presumed surface boundaries and gets optimized iteratively.

We describe the method for the case when *all four labels*

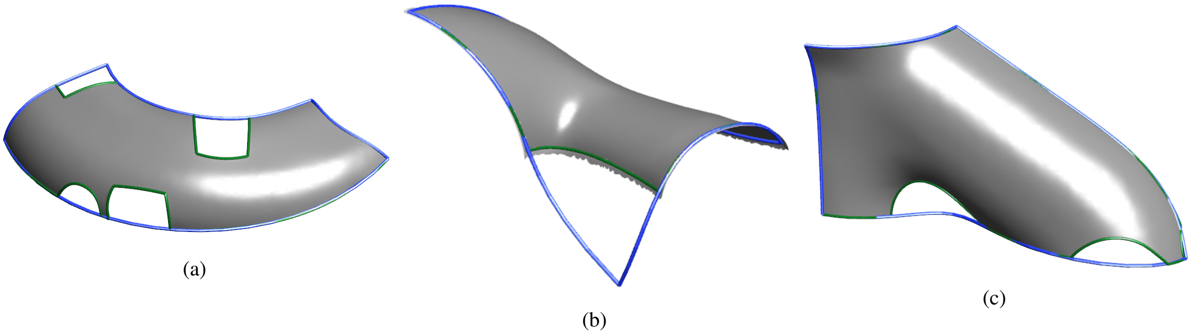


Figure 2: Examples of (quintic) guiding frames in blue

are present. Take the points in $\mathbb{R}^3$ on the labeled boundaries, denoted by $(\mathbf{l}^i_1,\dots,\mathbf{l}^i_{n_i})$, $(i=1,2,3,4)$; we look for parametric curves $\mathbf{c}^i(t):[0,1]\to\mathbb{R}^3$, $(i=1,2,3,4)$ with $\mathbf{c}^i(1)=\mathbf{c}^{i+1}(0)$ $(i \bmod 4)$, approximating the points in a least-squares sense. In particular, we aim to solve the optimization problem

$$\underset{\mathbf{c}^i(t)}{\text{minimize}} \quad \underbrace{\sum_{i=1}^{4}\left(\sum_{j=1}^{n_i}\left\|\mathbf{c}^i(t_j)-\mathbf{l}^i_j\right\|^2+\lambda E_{\text{sm.}}(\mathbf{c}^i)\right)}_{E_{\text{frame}}} \qquad (1)$$

where t_j is the parameter value assigned to the data point l_j, $E_{\text{sm.}}$ is a regularization term, measuring the smoothness of the curve i, weighted by λ.

At each iteration, we first project the data points on the corresponding sides of the current frame configuration, which gives them a parameterization. Then, we optimize the frame geometry by solving (1), and iterate in this manner.

We represent the frame using *Bézier curves*, as the global support of Bernstein polynomials helps avoiding weak control points and bad conditioning. For the smoothness energy, we choose the usual (parametric) bending measure:

$$E_{\text{sm.}}(\mathbf{c})=\int_0^1\|\ddot{\mathbf{c}}(t)\|^2\,dt, \qquad (2)$$

in which case the functional remains quadratic in the control points.

To avoid overfitting the data, or getting stuck in local optima, we fit frames of progressively higher degree. We start with a linear Bézier frame, i.e. a 3D quadrilateral, defined by the shared corners of the adjacent labels or the average of their endpoints. We optimize the linear frame, and when it cannot be improved any further, we raise its degree and proceed with fitting the quadratic frame and iterate until we approximate the labeled data points within a prescribed tolerance.

In Figure 2 we show three trimmed meshes, where guiding frames were computed with the described algorithm. We call attention to Figure 2b, where the labeled segments are nearly parallel at their endpoints, so naive corner estimation methods by extending tangents, etc. would fail.

When some labels are missing, we optimize a frame with only three or two curves. We note that in these cases, parameterization is less constrained and distortion minimization is generally sufficient for setting correct 2D proportions.

4.2. Labeled Parameterization

With the guiding frame at hand, we can compute a parameterization that respects the constraints implied by the labels and the frame. This is achieved in two steps, as illustrated by Figure 3:

- We first map the triangles to the plane with rigid motions, so they are oriented according to the label constraints.
- Then, we "stitch" this triangle soup together to form a connected planar mesh, while respecting the alignment of the constrained regions computed previously, only optimizing their scale to minimize distortion and retain the proportions of the guiding frame.

Our approach is an extension of the Local-Global algorithm for As-Rigid-As-Possible (ARAP) parameterization.[6,10] For details on this step, we refer to [Ref. 11, Sec. 4.2].

5. Mapping from 2D to 3D

A parameterized trimmed region can readily be fitted by a tensor-product surface. However, we might have weak control points outside the trimmed region, with very few data points in the support of their basis functions and this might lead to numerical instability and excessive wiggling of the fitted surface. We protect against the effects of weak control points by extending the original triangle mesh in the 3D space. The stabilizing effect is illustrated in Figure 4. Our method is a variant of hole-filling and surface extrapolation techniques; starting with the parameterization of the mesh, we extend the domain to an enlarged domain rectangle, triangulate it, and optimize the new vertex positions in 3D for smoothness and shape-preservation.

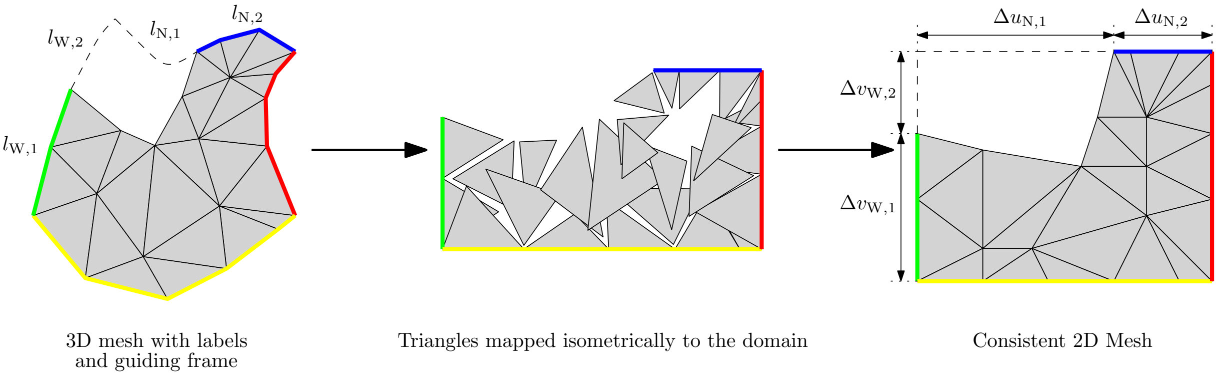
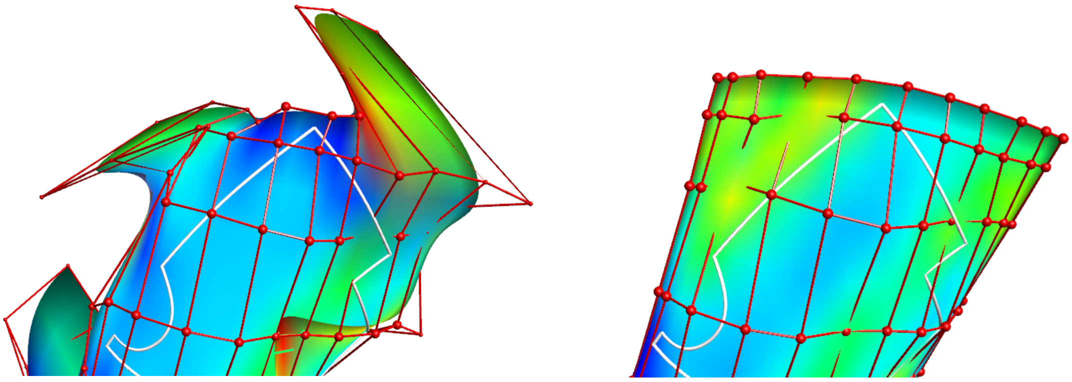

$l_{\mathrm{W},2}$
$l_{\mathrm{N},1}$
$l_{\mathrm{N},2}$
$l_{\mathrm{W},1}$
$\Delta u_{\mathrm{N},1}$
$\Delta u_{\mathrm{N},2}$
$\Delta v_{\mathrm{W},2}$
$\Delta v_{\mathrm{W},1}$
3D mesh with labels and guiding frame
Triangles mapped isometrically to the domain
Consistent 2D Mesh

Figure 3: Steps of labeled parameterization



(a) Fit without extension

(b) Fit with extension

Figure 4: Stabilizing effect of extension (colors indicate mean curvature; the size of control points is proportional to the number of data points in their support)

5.1. Extension in 2D and Triangulation

We sample the region outside the parameterized mesh in the (u,v) plane with a regular grid of points. The grid size is chosen, such that in both the u- and v-directions the number of samples is the square root of the number of mesh vertices scaled by the ratio of the areas of the domain rectangle and the parameterized mesh. The supplemented vertices $\mathcal{V}_{\mathrm{new}}$, together with the original boundary vertices $\partial V_{\mathrm{old}}$ are then triangulated into the 2D mesh $\mathcal{M}' = (\mathcal{V}', \mathcal{E}', \mathcal{T}')$. In practical scenarios, it is often required that the domain of the fitted surface extends beyond the bounding rectangle of the parameterized data points — with this in mind, we add additional layers of supplementary vertices (see for example Figure 6).

5.2. Extension in 3D

We want to determine the 3D positions $\mathbf{p_i} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}$ of the new vertices $v_i \in \mathcal{V}_{new}$ so they provide a smooth extension of the original 3D mesh.

We minimize an energy of the vertex coordinates, that is composed of two terms:

$$E_{\mathrm{3D}} = (1 - w_{\mathrm{Fair}})E_{\mathrm{Ext.}} + w_{\mathrm{Fair}}E_{\mathrm{Fair}}, \tag{3}$$

where $E_{\mathrm{Ext.}}$ ensures that we smoothly extrapolate the original mesh, E_{Fair} adjusts the fairness of the extended surface, and w_{Fair} is a weight controlling the trade-off between them. We next describe both of these energies.

5.2.1. Extension energy

To ensure smooth extrapolation of the original mesh boundaries, we have chosen to optimize the quadratic *polyharmonic energy*[3] of the coordinate functions:

$$E_{\mathrm{Ext.}} = \left\| \nabla^k \mathbf{p} \right\|^2, \tag{4}$$

which enforces approximate G^{k-1} continuity along the boundary. In practice we set $k = 2$, (enforcing G^1 continuity), or $k = 3$ (enforcing G^2 continuity).

We illustrate the effect of these two energies in Figure 5, where a three-sided patch with two labeled edges is extended in two different ways. Both extensions are meaningful from an engineering point of view, and both surfaces will smoothly approximate the given data points within a prescribed tolerance, but their shapes are significantly different. The properties of the final CAD model and the forthcoming operations determine which representation should be chosen. Our experience shows that – as a default – G^2 extensions are preferable, since these generally yield the most natural shapes beyond the trimmed regions. At the same time, there exist cases, when enforcing strongly curved extensions may yield counter-intuitive geometric constraints, and switching to G^1 produces better results.

5.2.2. Fairness Energy

Minimizing the extension energy ensures a smooth connection to the original surface, but the unconstrained boundaries of the four-sided mesh might suffer unnatural shrinkage and wiggling. To avoid this, we add a fairness energy composed of two terms:

$$E_{\text{Fair.}} = (1 - w_{\text{Iso}})E_{\text{Mesh}} + w_{\text{Iso}}E_{\text{Iso}}, \qquad (5)$$

where E_{Mesh} measures the fairness of the mesh, E_{Iso} that of a set of constant parameter isocurves. w_{Iso} controls the tradeoff between these objectives.

5.2.2.1. Isocurve Fairness This term aims to achieve an even curvature distribution along the boundary for the extension, as well as ensuring that labeled boundary segments are smoothly extrapolated as isocurves to the boundary. We choose to minimize, for a set C of isocurves (which are contained in the mesh as polylines) the energy:

$$E_{\text{Iso}} = \int_{c \in C} \left(\left| \frac{d^k x(t)}{dt^k} \right|^2 + \left| \frac{d^k y(t)}{dt^k} \right|^2 + \left| \frac{d^k z(t)}{dt^k} \right|^2 \right) dt$$

which is discretized over the polylines using simple finite differences. Note that for $k = 2$, it measures the curvature and for $k = 3$, the curvature variation of the curves.

We include in the set of optimized isocurves, the four boundaires of the extension, as well as those containing the labeled segments of the original mesh, see Figure 6 for an illustration of its effect.

5.2.2.2. Mesh Fairness The isocurve energies serve as localized (soft) constraints on the mesh, which leads to localized distortions and artifacts. This is further enhanced using again the method of,[8] i.e. by adding the energy term E_{Mesh} measuring the squared sum of differences between the ARAP distortions of adjacent triangles. We set the rotation matrices based on an initial extension computed by minimizing only the energy $E_{\text{Ext.}}$.

Note, that we optimize the smoothness of the ARAP energy, but not its actual value. This is motivated by the fact that the 2D parameterization might be distorted due to the imposed constraints, in which case the inverse mapping into 3D is expected to have high ARAP distortion, as well.

6. Discussion and Results

In this section we present our results by means of several examples. We do not discuss the tuning of fitter parameters, as this would force us to compare various fitting methods as well, rather we prefer using a black-box surface fitter with a prescribed number of control points and negligible smoothing weights, that permit comparing the effects of parameterization in different cases. We will compare “ad-hoc” surfaces, where parameterization has not been optimized, with surfaces where labeling and/or extensions have been applied.

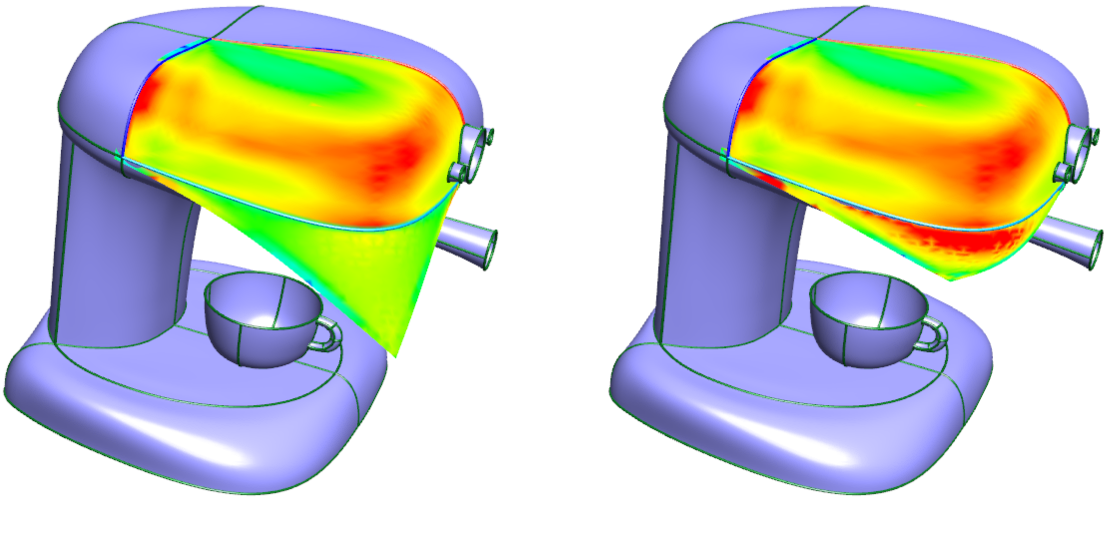
6.1. Test Cases

6.1.1. Test Case 1

We demonstrate the importance of stable surface extensions by a simple reverse engineering example. Segmentation produces a free-form and an extruded region bounded by two trimming loops, the mesh is shown in Figure 7a. After fitting, these surfaces are exported into a CAD system to build a B-rep model to be modified by the user. We consider two use cases: in the first, a sharp edge needs to be created; in the second, a blend with a smaller radius. Both the intersection curve and the bottom rail curve of the blend must lie on the free-form surface beyond the original trimming loops. A simple regularized fit may produce oscillations and imperfect curves, as shown in Figure 7b and Figure 7c. Our proposed approach yields curves of higher quality due to the stability of the extension method, see Figure 7d and Figure 7e. It should be noted that setting the regularization weights is a delicate issue; low values may produce unwanted wiggling, high values may supersede accuracy. In contrast, extension based fitting is numerically stable and less sensitive to fine tuning. If the model is modified in a later phase, for example the extruded surface is shifted or the radius of the fillet is changed, this surface is suitable for these operations.

6.1.2. Test Case 2

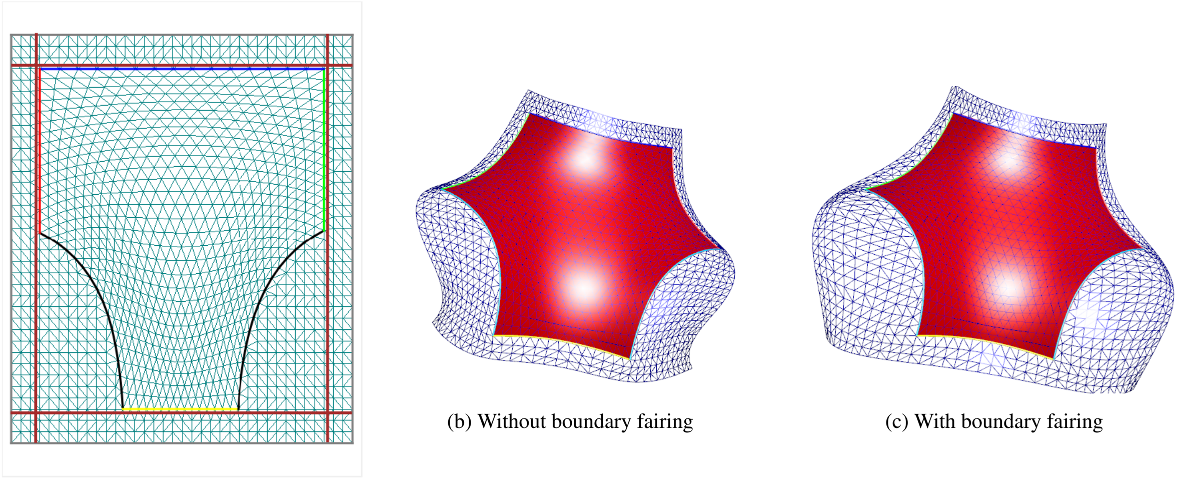
The trimmed mesh in this test comes from a fairly complex, molded mechanical part, originally defined by several Boolean and filleting operations. Our task is to reproduce a nicely extendable primary surface over the trimmed region, as this will be necessary to compute intersections and fillets for reverse engineering the object. The stability of this primary surface is crucial for all dependent geometric entities.A simple ARAP parameterization results in the fit shown in Figure 8a. We show another surface using the same parameterization, but now applying extension, as well, on Figure 8b: the obtained surface is clearly better. To further improve the fit quality, labeling can be applied which results in the surface seen on Figure 8c. Observe, that the North



(a) G^1 extension

(b) G^2 extension

Figure 5: Comparison of mesh extension energies with different continuity (colors indicate mean curvature)



(a) Parameterization

(b) Without boundary fairing

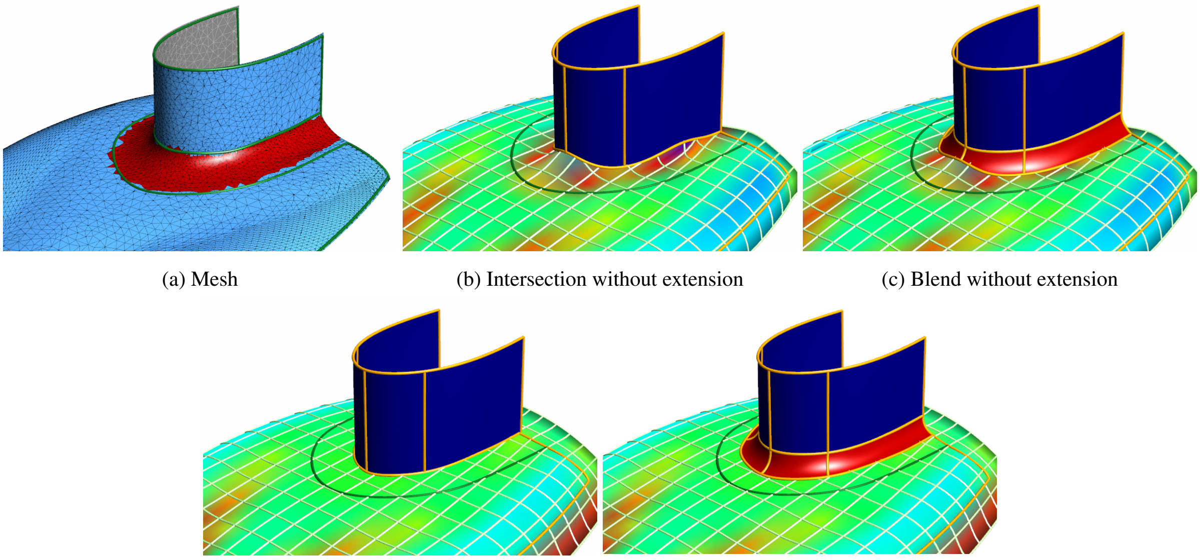
(c) With boundary fairing

Figure 6: Effect of boundary fairing (brown lines indicate constrained isocurves)

and South labels consist of several segments, and a corner is also missing. Also note, that the previously shown guiding frame (Figure 2a) suggests that the lengths of the North and the South sides significantly differ, so adherence to labeling requires a distorted parameterization.

6.1.3. Test Case 3

Our last example is a sheet metal part shown in the workflow (Figure 1); challenges are due to noisy data, the two holes and the large missing portion of the surface to be fitted. Fitting a high quality surface would be difficult without labeling and extensions. The parameterization must be properly aligned and the weak control points must be stabilized. "Ordinary" flattening of the mesh would produce a parameterization that results in a fit with excessive wiggling due to the weak control points as shown in Figure 9a. If we apply extension, the surface quality is much better (Figure 9b), although the surface remains quite large with a peculiar control net. As expected, the best results are obtained using both labeling and surface extension. A labeled parameterization,



(a) Mesh

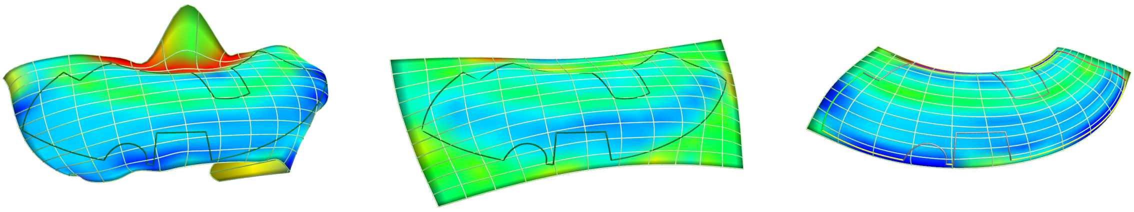
(b) Intersection without extension

(c) Blend without extension

(d) Intersection with extension

(e) Blend with extension

Figure 7: Test Case 1 (colors indicate mean curvature)



(a) No labeling/No extension

(b) Only extension

(c) Labeling & extension

Figure 8: Test Case 2 (colors indicate mean curvature)

as shown in Figure 1, produces smooth transitions over the missing parts as can be seen in Figure 9c.

7. Conclusions

In this paper we discussed useful techniques that significantly enhance the quality of tensor-product surfaces fitted to triangle meshes. We have focused on parameterization and extension techniques. Using labels and guiding frames combined with ARAP mappings led to a special, constrained parameterization for the trimmed mesh, that indirectly defines a good orientation and layout for the surface. Extending the trimmed parametric map in the domain and then accordingly extrapolating the trimmed region in 3D has led to a smooth, four-sided mesh, that facilitates fitting predictable, well-controlled surfaces without undesirable oscillations.

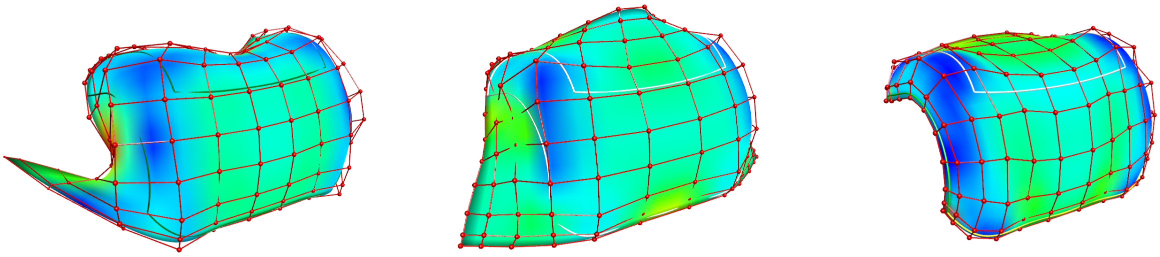
Acknowledgements

This project has been supported by the Hungarian Scientific Research Fund (OTKA, No.124727). We acknowledge several thought-provoking technical discussions with our colleague, Peter Salvi. All images were generated by the Sketches prototype system (ShapEx Ltd, Budapest); the exceptional development contribution by György Karikó is highly appreciated.

References

1. M. Attene, M. Campen, and L. Kobbelt. Polygon mesh repairing: An application perspective. *ACM Computing Surveys (CSUR)*, 45(2):15, 2013. 2

2. M. Botsch and L. Kobbelt. An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics (TOG)*, 23(3):630–634, 2004. 2



(a) No labeling/No extension (b) Only extension (c) Labeling & extension

Figure 9: Test Case 3, sheet metal part

3. M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon Mesh Processing*. CRC Press, 2010. 2, 5

4. K. Höllig and U. Reif. Nonuniform web-splines. *Computer Aided Geometric Design*, 20(5):277–294, 2003. 2

5. B. Lévy. Dual domain extrapolation. *ACM Transactions on Graphics (TOG)*, 22(3):364–369, 2003. 2

6. L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler. A local/global approach to mesh parameterization. *Computer Graphics Forum*, 27(5):1495–1504, 2008. 4

7. E. Marchandise, J.-F. Remacle, and C. Geuzaine. Quality surface meshing using discrete parametrizations. In *Proceedings of the 20th International Meshing Roundtable*, pages 21–39. Springer, 2012. 2

8. J. Martinez Esturo, C. Rössl, and H. Theisel. Smoothed quadratic energies on meshes. *ACM Transactions on Graphics (TOG)*, 34(1):2, 2014. 6

9. B. Marussig and T. J. R. Hughes. A review of trimming in isogeometric analysis: Challenges, data exchange and simulation aspects. *Archives of Computational Methods in Engineering*, pages 1–69, 2017. 2

10. M. Vaitkus and T. Várady. A general framework for constrained mesh parameterization. In *Proceedings of the 31st Spring Conference on Computer Graphics*, pages 15–21. ACM, 2015. 4

11. M. Vaitkus and T. Várady. Parameterizing and extending trimmed regions for tensor-product surface fitting. *Computer-Aided Design*, 2017. 1, 2, 4

12. W. Wang, H. Pottmann, and Y. Liu. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics (TOG)*, 25(2):214–238, 2006. 2

13. V. Weiss, L. Andor, G. Renner, and T. Várady. Advanced surface fitting techniques. *Computer Aided Geometric Design*, 19(1):19–42, 2002. 2