

Title:

**Approximating Triangular Meshes by Implicit, Multi-Sided Surfaces**

Authors:

Ágoston Sipos, asipos@edu.bme.hu, Budapest University of Technology and Economics

Tamás Várady, varady@iit.bme.hu, Budapest University of Technology and Economics

Péter Salvi, salvi@iit.bme.hu, Budapest University of Technology and Economics

Keywords:

Implicit surfaces, Multi-sided patches, Mesh approximation

DOI: 10.14733/cadconfP.2021.236-240

Introduction:

We investigate the capabilities and limitations of approximating a triangular mesh by a collection of smoothly connected, multi-sided implicit patches. We assume that the triangular mesh represents an object bounded by large free-form faces, excluding high curvature variations and tiny features. We also assume that there exists an initial network associated with the mesh; its vertices determine the corners of the patches to be constructed, and its edges define loops for the patchwork. Such a network may be created by a user, or produced by an automatic quad-mesh algorithm, or defined by a 3D cell structure—examples will be given later. We compute an approximating patchwork and evaluate the deviations. If the accuracy is not sufficient, we adaptively refine the initial structure by subdivision (permitting T-nodes) and then refit, as needed.

The use of general implicit surfaces for free-form modeling is a controversial issue. They are generally  $C^\infty$ -continuous and represent half-spaces, so point-membership classification is straightforward. No parameterization is needed for distance computations and approximating data points. Implicit surfaces are favorable for several geometric interrogations, e.g. ray-tracing or intersections. Counter-arguments include various shape problems, e.g. singularities, self-intersections, handling several disconnected surface portions, and the lack of shape parameters. The approach of using finite implicit patches that are evaluated only within a well-defined subspace, preserves the benefits and helps to eliminate the difficulties.

We will use I-patches [3], a class of implicit surfaces that interpolate a loop of boundary curves. Each segment is defined as the intersection of a *ribbon* (primary surface) and a *bounding surface*, both given in implicit form. I-patches smoothly join to the given ribbons. We will determine the ribbons and bounding surfaces in such a way that the I-patch interpolates the corner points on the mesh, and its boundary curves approximate the underlying data points. I-patches enable the computation of *faithful* distances, and have further degrees of freedom, by means of which we compute an optimized approximation in the interior.

Current authors recently published another paper [2], where I-patches were constructed for polyhedral design and setback vertex blending. This also included a summary of important implicit approaches.

In this paper we are going to focus on how to create appropriate ribbons and bounding surfaces in the mesh approximation context, how to optimize the free parameters to minimize deviation from the input data, and how to adaptively refine the patchwork.

### Preliminaries:

An  $n$ -sided I-patch is defined by  $n$  ribbons and bounding surfaces, given in implicit form as  $R_i = 0$  and  $B_i = 0$ , respectively ( $i = 1 \dots n$ ). The patch itself is constructed as the 0-isosurface of

$$I(x, y, z) = \sum_{i=1}^n w_i R_i \prod_{j \neq i} B_j^2 - w_0 \prod_{j=1}^n B_j^2, \quad (2.1)$$

where  $w_i$  ( $i = 0 \dots n$ ) are free scalar parameters. It is defined within the volume enclosed by the bounding surfaces, i.e., for points where  $B_i \geq 0$  for all  $i$ .

The intersection of  $R_i$  and  $B_i$  defines the  $i$ -th boundary curve, which is interpolated by the I-patch, since all terms of the above expression vanish for points on the boundary. Similarly, it is easy to show that at these points the gradient of  $R_i$  will be parallel to the gradient of the I-patch, so  $G^1$  continuity is ensured. (Raising the degrees in Eq. (2.1) would guarantee a higher order of continuity.) When two or more bounding surfaces coincide, special handling is needed; please refer to [2] for more details.

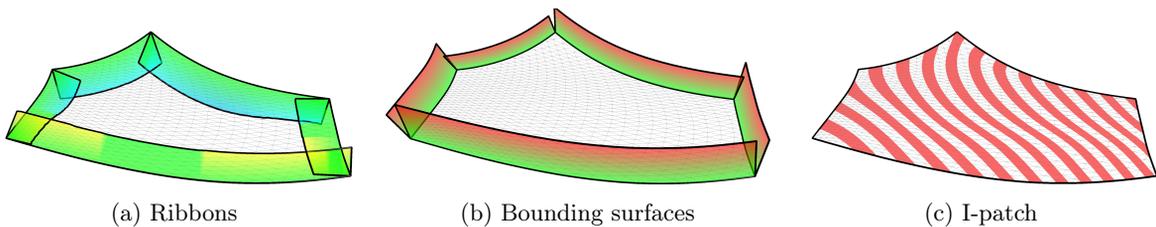


Fig. 1: Construction of an I-patch.

### Ribbons and Bounding Surfaces:

In this section we describe how to construct ribbons and bounding surfaces, whose intersection curves will approximate the underlying mesh data. As will be explained below, all surfaces in this context are constructed as a weighted combination of planes given at the corner points and auxiliary planes derived from the patch configuration.

Let us take two adjacent corner points  $\mathbf{p}_1$ ,  $\mathbf{p}_2$ , and incident *corner planes*  $\pi_1$ ,  $\pi_2$  with normal vectors  $\mathbf{n}_1$ ,  $\mathbf{n}_2$ . The corresponding ribbon interpolates both points and normal vectors. Similarly, the bounding surface interpolates both points, but it is transversal to the ribbon surface.

We are going to use three kinds of surface for both ribbons and bounding surfaces: (i) planes, (ii) *Liming-surfaces*, and (iii) *I-lofts*.

Liming [1] created conic curves as the combination of three implicitly given lines. A Liming-surface, in turn, is the combination of three planes given in implicit form. When used as a ribbon, these are the two corner planes and an auxiliary plane  $\tilde{\pi}$  that contains both  $\mathbf{p}_1$  and  $\mathbf{p}_2$ . The ribbon equation is then given as  $(1 - \lambda)\pi_1\pi_2 - \lambda\tilde{\pi}^2 = 0$ , where  $\lambda$  controls the fullness of the shape. When used as a bounding surface, orthogonal planes take the place of corner planes.

I-lofts are simple, two-sided I-patches defined by two pairs of orthogonal planes containing a corner point. When used as a ribbon, its primaries will be  $\pi_i$ ; and when used as a bounding surface, its primaries will be planes perpendicular to  $\pi_i$ .

The related boundary curves depend on the types of the intersecting surfaces. We may obtain straight lines, when both are planes; conic curves or I-segments (a cross-section of an I-loft), when one of the surfaces is planar; or general 3D intersection curves. In order to approximate a given boundary, we select the simplest applicable ribbon–bounding pair; various geometric constraints, such as inflections or twisted normal vectors require I-lofts, see [2]. Note that the ribbon and the bounding surface associated with a

given  $\mathbf{p}_1\mathbf{p}_2$  boundary is the same for the two adjacent I-patches that share this boundary, thus smooth connection is ensured by the algebra.

#### Mesh Approximation:

The construction outlined above has several degrees of freedom. Corner points are assumed to be given, and the corresponding normal vectors are approximated from the input mesh. But Liming-surfaces still have a free parameter  $\lambda$ , and I-lofts have two independent weights. Here we describe how we can optimize these, as well as the scalar weights of the I-patch itself, to obtain the best approximation in the interior of the patches.

For edges in the interior of the mesh, it is sufficient to use planar bounding surfaces, while along the boundary of the mesh typically curved bounding surfaces are needed. We trace a polyline along the boundary as the intersection of the mesh and the bounding surface, and fit ribbons to approximate sampled points on the polyline. When the Liming-surface and the I-loft are both valid, we choose the one with smaller error.

The figure below shows a Liming and an I-loft ribbon, with corner planes and sampled points on the mesh. The deviation map indicates that the ribbons are close to the mesh (green) in the vicinity of the related boundary.

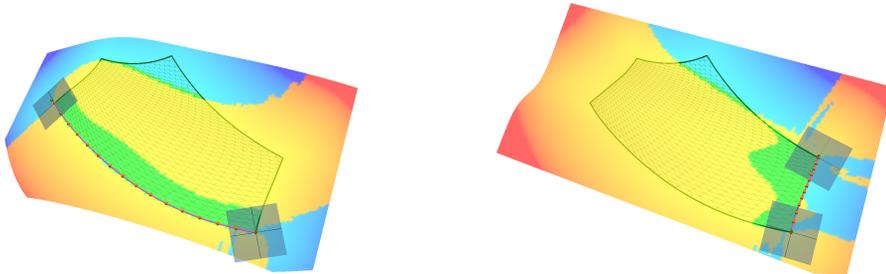


Fig. 2: A Liming (left) and an I-loft ribbon (right), showing deviation.

Next, mesh points are filtered out so only those remain that are enclosed by the bounding surfaces. Then we compute the mass center of the polyline boundaries of the patch, and project this point onto the mesh. The initial patch created goes through this center point.

Finally, the scalar weights of the I-patch are optimized to fit the data points. In the approximations we minimize an error function that is the squared sum of *faithful* algebraic distances (a special formulation producing a distance field close to Euclidean, see the full paper). For both of the above optimizations, derivative-free methods can be applied, such as the Nelder–Mead algorithm.

#### Adaptive Refinement of the Patchwork:

Once an initial patchwork has been generated, we check whether it approximates the data points within a prescribed tolerance and—if necessary—we perform an adaptive refinement following simple heuristic rules. When it is out of tolerance, each related boundary is subdivided halfway between its endpoints, and each related patch is split using a new mesh point in its interior. We connect the new subdivision points, and add new artificial points in order to create a valid structure. This leads to a new graph of edges with new ribbons and bounding surfaces. Care should be taken, however, to prevent the propagation of local refinements over the full patchwork. Fortunately, I-patches are well-suited for producing T-nodes, as will be shown below.

The first example in Figure 3 shows a boundary connecting  $\mathbf{p}_1$  and  $\mathbf{p}_2$  that needs to be subdivided; a new mesh point  $\mathbf{p}_m$  is inserted and four new boundaries are created. The second example shows another

configuration, where the patch needs to be split by  $\mathbf{p}_c$ . Here we preserve the ribbon connecting  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , and inherit its midpoint  $\mathbf{q}_m$  *exactly*, so the original ribbon and bounding surface constraints remain in effect for the subdivided patch, maintaining  $G^1$  continuity for this T-node.

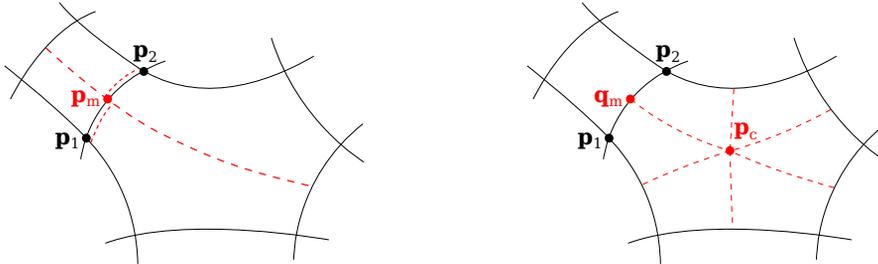


Fig. 3: Cases of adaptive refinement.

#### Examples:

Figure 4 shows the deviation map of a *shoe last* model. The network was created by a uniform cell structure yielding two 3-sided, six 4-sided and two 5-sided I-patches. The accuracy of the approximation in comparison to the bounding box axis is on average 0.069%, with a maximum of 0.35%.

The second example (Fig. 5) is a sheet metal part, showing the difference between the deviation maps before and after optimizing the interior; accuracies changed from an average of 0.055% to 0.035%, while the maximum deviation decreased from 0.399% to 0.226%.

Finally, Figure 6 shows the front part of a concept car. The first image depicts the ribbons of a sparse network that has been refined in two steps. The deviations decrease (average: 0.14%  $\rightarrow$  0.027%  $\rightarrow$  0.016%, maximum: 3%  $\rightarrow$  0.3%  $\rightarrow$  0.12%) as the number of I-patches increases (10  $\rightarrow$  15  $\rightarrow$  28).

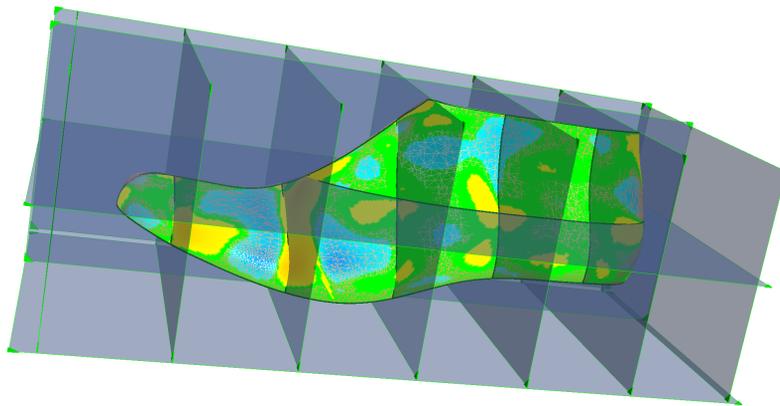


Fig. 4: Cell-based subdivision on a model of a shoe last.

#### Conclusion:

We have shown that it is possible to approximate a free-form object by a patchwork of smoothly connected implicit patches, defined by corner points placed on a mesh and a graph that determines the connectivity between them. Each patch is constructed by means of tangent planes and various fullness weights to obtain a good approximation. These surfaces are naturally more rigid than parametrics, but have various interesting properties, which motivates our future research. The full paper describes the creation of ribbons and bounding surfaces, faithful distance computation and refinement of the patchwork.

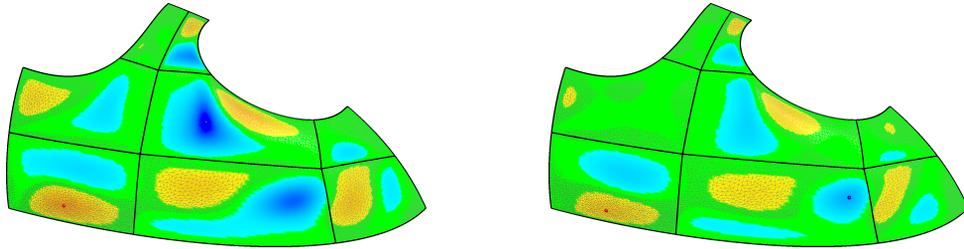


Fig. 5: Sheet metal part with deviation, before and after optimization.

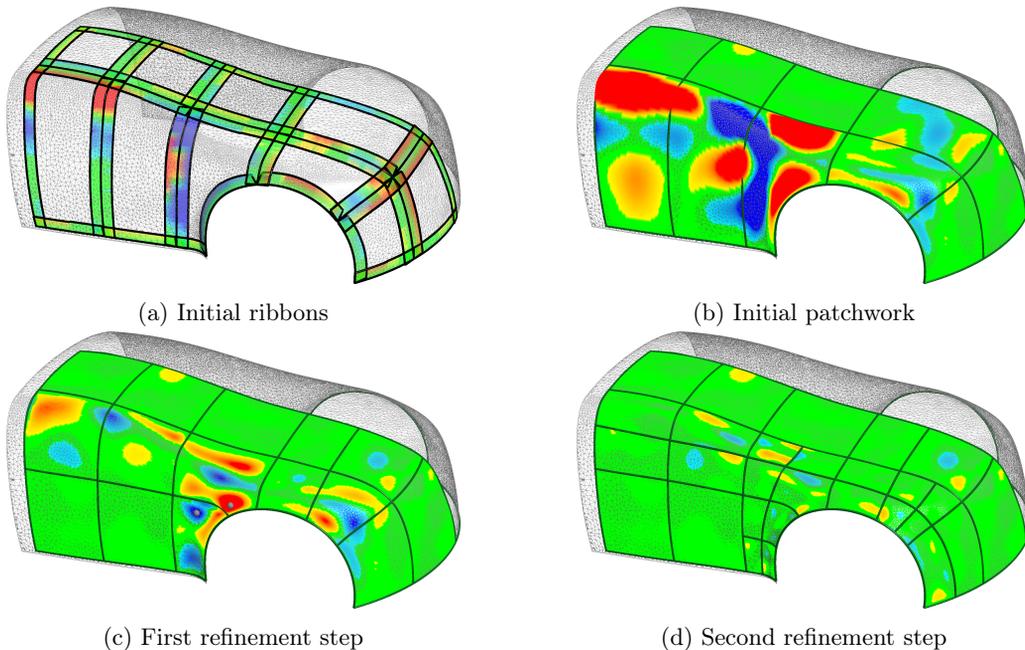


Fig. 6: Steps of adaptive refinement on a concept car model, showing deviation.

#### Acknowledgement:

This project has been supported by the Hungarian Scientific Research Fund (OTKA, No. 124727), and the EFOP-3.6.1-16-2016-00014 project, financed by the Ministry of Human Capacities of Hungary. The outstanding programming contribution of György Karikó is highly appreciated.

#### References:

- [1] Liming, R. A.: Conic lofting of streamline bodies: the basic theory of a phase of analytic geometry applicable to aircraft, *Aircraft Engineering and Aerospace Technology*, 19(7), 1947, 222–228. <https://doi.org/10.1108/eb031528>
- [2] Sipos, Á.; Várady, T.; Salvi, P.; Vaitkus, M.: Multi-sided implicit surfacing with I-patches, *Computers and Graphics*, 90, 2020, 29–42. <https://doi.org/10.1016/j.cag.2020.05.009>
- [3] Várady, T.; Benkő, P.; Kós, G.; Rockwood, A.: Implicit surfaces revisited – I-patches, *Geometric Modelling*, 2001, 323–335. [https://doi.org/10.1007/978-3-7091-6270-5\\_19](https://doi.org/10.1007/978-3-7091-6270-5_19)