A New QEM for Parameterization of Raster Images

Xuetao Yin, John Femiani, Peter Wonka, and Anshuman Razdan

Arizona State University
Xuetao.Yin@asu.edu

Abstract

We present an image processing method that converts a raster image to a simplical 2-complex which has only a small number of vertices (base mesh) plus a parameterization that maps each pixel in the original image to a combination of the barycentric coordinates of the triangle it is finally mapped into. Such a conversion of a raster image into a base mesh plus parameterization can be useful for many applications such as segmentation, image retargeting, multi-resolution editing with arbitrary topologies, edge preserving smoothing, compression, etc. The goal of the algorithm is to produce a base mesh such that it has a small colour distortion as well as high shape fairness, and a parameterization that is globally continuous visually and numerically. Inspired by multi-resolution adaptive parameterization of surfaces (MAPS) and quadric error metric (QEM), the algorithm converts pixels in the image to a dense triangle mesh and performs error-bounded simplification jointly considering geometry and colour. The eliminated vertices are projected to an existing face. The implementation is iterative and stops when it reaches a prescribed error threshold. The algorithm is feature sensitive i.e. salient feature edges in the images are preserved where possible and it takes colour into account thereby producing a better quality triangulation.

Keywords: image parameterization, image vectorization, decimation, quadric error metrics

ACM CCS: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems I.4.0 [Image Processing and Computer Vision]: General—Image processing software;

1. Introduction

Geometry and image processing have had a mutual influence resulting in many trends, terms, and methodologies shared by both communities. Parameterization is a term used by both, however, it refers to completely different practices in the two domains. Mesh parameterization [HPS08] maps a 3D geometry to a more managable base domain, usually in a plane. Image parameterization [KŠ09] mostly refers to the extraction of feature parameters from images.

In this paper, we propose a method that does two things simultaneously. It converts a raster image to a coarse triangle mesh, called a *base mesh*, that captures the structure of the image content; it also maps each pixel in the original image to a point on the base mesh resulting in a non-trivial globally continuous parameterization, that is represented in terms of the barycentric coordinates of base mesh faces. This parameterization is constructed through repeated conformal remapping; it meets the C^0 continuity condition numerically at based mesh edges. The parameterization is of high visual quality as shown in figure 1 (fourth row; detailed discussion about the figure is in section 3.4). Such a conversion of a raster image into a base mesh plus parameterization can be useful for many applications such as segmentation, image retargeting, multi-resolution editing with arbitrary topologies, edge preserving smoothing, compression, etc. The goal of the algorithm is to produce a base mesh with per-pixel association such that (a) the reconstruction colour error is small, and, (b) the quality of the resulting triangulation is high. The algorithm, combines non-planar mesh parameterization [LSS*98] and quadric error metrics [Hop99], converts all pixels in the image to a dense triangle mesh and performs error-bound simplification jointly considering geometry and colour. The eliminated vertices are projected on an existing facet and forms a non-trivial globally continuous parameterization. The implementation is iterative and stops when it reaches a prescribed error threshold. The algorithm is feature sensitive i.e. salient feature edges in the images are preserved where possible and it takes colour into account thereby producing a better quality triangulation compared to existing methods. The parameterization associates a set of pixels to a face in the base mesh; the boundaries of those patches follow the curvilinear features in the image. Our scheme is condusive to arbitrary topology (for example the input can be *cut outs* of irregular shape from images) and can be extended to higher dimensional feature spaces.

Contributions: First, we propose a new quadric error metric, a modified formula of [Hop99] with geometry error computed using a special case of the general form in [GZ05], for converting raster images into a triangle mesh that takes shape and colour distortion into consideration. Second, we produce a non-trivial globally continuous parameterization for each pixel in the original image. This parameterization enables us to perform segmentation and image editing. The algorithm runs in $O(N \log N)$ time; N being the number of pixels in the image. The metric is flexible and allows the user to tune the algorithm to be more (or less) sensitive to colour features by independently changing the weights for the geometry and colour parameters. Figure 1 shows an example of our algorithm and its comparison to that of [Hop99].

2. Prior Art

Mesh Parameterization was introduced for triangle meshes for mapping textures onto surfaces, normal mapping, morphing, remeshing, mesh editing, compression, etc. See [HPS08] for a survey. It is a mapping from a higher dimensional surface (3D surface for example) to a base parameter domain, which can be planar, spherical or a simplicial complex [HPS08]. A well known global parameterization method on non-planar parameter domain was proposed by Lee *et al.* [LSS*98]. Their multiresolution adaptive parameterization of surfaces (MAPS) algorithm utilizes a simplicial complex as the domain. MAPS produces globally smooth parameterization by iteratively collapsing vertices and performing conformal mapping. MAPS is useful for remeshing, texture mapping and geometry morphing. However, MAPS cannot be applied to images directly because its priority computation is not extensible to pixels with colour features. Lee [Lee00] replaces the error metric in MAPS with quadric error metric of geometric distortion to fit subdivision surfaces to triangle meshes. The MAPS parameterization is considered globally smooth for its continuity across the patches. Higher order of parametric continuity is achieved in Khodakovsky et al. [KLS03] through relaxation using a set of transition functions. However, for parameterization of higher dimensional data, like in our case, such smoothness is unnecessary and hard to achieve without introducing large amount of distortion.

Mesh Simplification was first proposed to produce levels of detail by subsequently removing elements from a complex object [Lue01]. Most early simplification algorithms focused on the geometry aspect of the mesh and tried to minimize the volume shrinkage. Many algorithms adapt quadrics [GH97] to approximate the error. Cohen *et al.* [COM98] proposed a method to preserve the appearance during simplification by maintaining the texture coordinates of original vertices. Garland and Heckbert [GH97] and Hoppe [Hop99] extended the quadric error metrics (QEM) to incorporate vertex attributes such as colour and texture coordinates. Garland and Heckbert [GH97] concatenate the feature components to the 3D geometry to form a high dimensional space and consider distances to the tangent hyperplanes. Garland and Zhou [GZ05]

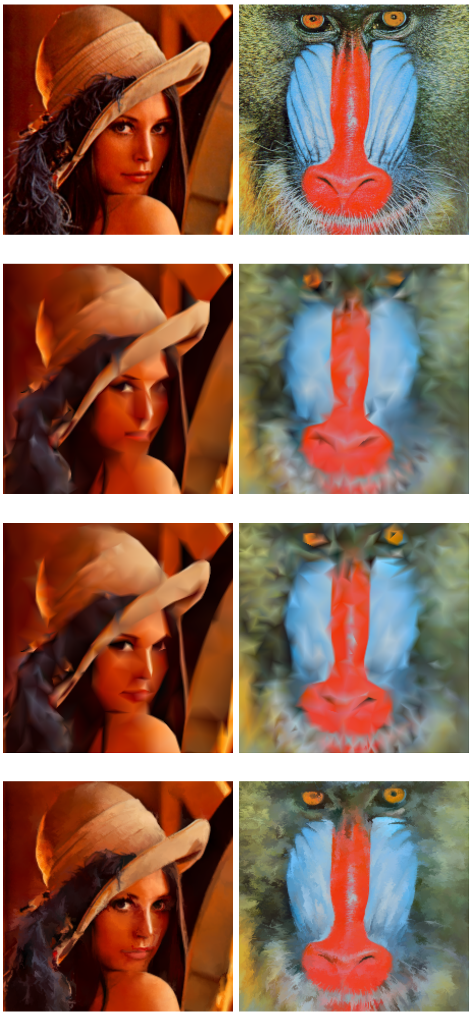


Figure 1: *From top down: original images of* Lena *and* Mandrill. *Second and third rows represent base meshes using Hoppe's and our method respectively; base meshes have 1300(Lena) and 1960 (Mandrill) faces. The fourth row shows the reconstruction by interpolating the colour using our parameterization. There is no visible discontinuity across the base mesh faces. It qualitatively shows that our parameterization is globally continuous and has small distortion in the feature space.*

generalize that formula to any dimension and distance to any hyperplanes. Hoppe [Hop99] distinguishes the feature components from the geometric ones and utilizes the geometric correspondence of the feature to compute the feature error. The feature error is computed with respect to the interpolated value at the projection in the geometry domain.

Existing QEMs are not ideal for images. QEMs are mostly defined as a summation of a spatial quadric that measures volume shrinkage and a colour quadric. Since the spatial domain of a raster image is planar, the volume shrinkage during simplifying an image mesh is zero everywhere except along the boundaries. These metrics [GH97, Hop99] numerically reduce to colour quadrics when applied to images. They are overly sensitive to noise in images of natural scenes and produce a suboptimal triangulation. To accommodate the specific demands, we adopt the methodology of Hoppe's and substitute its geometry error with one that penalizes moves in all directions, a special case of the general form in [GZ05]. similar idea was used in iso-surface simplification [ACSE05].

Image Triangulation and Parameterization Lai *et al.* [LHM09] convert raster images into vector images of similar appearance called *gradient meshes*. A key step of their algorithm can be considered as an analog of remeshing in images. In their parameterization step, a set of pixels is triangulated using constrained Delaunay triangulation. The resulting mesh is mapped to a planar parameter domain with *slits* [LHM09] using only the geometry information. The parameterization is later adjusted to take into account the colour information. The mapping is explicitly established for selected samples; it does not guarantee consistent and smooth parameter values for all pixels, e.g. at the ends of a slit. Unlike their scheme, our proposed scheme considers both geometry and colour information simultaneously and produces globally continuous parameter values for all pixels.

Many methods [XLY09, RFM05, LIL06, LL06, WTL*06] generate meshes from images. Xia *et al.* [XLY09] apply a local edge detector to find the curvilinear image feature which is later used to establish separate triangle meshes of pixel and subpixels for each colour channel, called *channel meshes*. These meshes are simplified using a simple error metric that measures the maximum error in each channel mesh. Ren *et al.* [RFM05] complete the curvilinear features in the input image by establishing a constrained Delaunay triangulation on the set of contours found by a local edge detector. The ARDECO method [LL06] fits a set of regions delimited by cubic splines to a raster image. Each region is filled with a constant colour or a gradient (linear or circular) to generate a vectorized version of the raster input. Lee *et al.* [LIL06] apply progressive meshes method proposed in [Hop96] to images and build a hierarchy of simplified meshes. [WTL*06] construct the neighbourhood graph of pixels with only pixel chroma values, and form an *appearance manifold*. Unlike our proposed methods, many of these triangulation methods require the detection of curvilinear features in the image as input and they do not keep track of the relationship between the resulting mesh and image pixels; hence they are not able to perform pixel related operations such as segmentation and editing. Because we introduce a location-preserving term in our metric, our base mesh is better suited for a variety of applications such as finite element analysis.

3. Multi-resolution Image Parameterization

Our algorithm performs simplification and parameterization simultaneously. In this section we describe the two procedures in detail and evaluate their performance.

3.1. Simplification

A raster image is converted to an initial dense triangular mesh; each pixel is represented by a vertex and the quadgrids are triangulated using one of the two diagonals. The diagonal chosen minimizes the colour differences between two end points. Notations are adopted from [Hop99, LSS*98]. We denote a triangular mesh as a pair $(\mathcal{P}, \mathcal{K})$, where $\mathcal{P}$ is a set of N regular indices of vertices; while the topology is represented as an *abstract simplicial complex* $\mathcal{K}$, set of singles (vertices), couples (edges), and triples (facets) of indices in $\mathcal{P}$. Each index $i \in \mathcal{P}$ is *realized* as a 5 dimensional point $\mathbf{v}_i = (\mathbf{p}_i, \mathbf{s}_i)^T = (x_i, y_i, L_i, u_i, v_i)^T \in R^5$ with $1 \leq i \leq N$, where $\mathbf{p}$ represents the geometry components, $\mathbf{s}$ represents the attribute components, and N is the number of vertices in the mesh. We confine the attributes to colour in this paper. We follow the discussion in [CM02] and choose L*u*v* for its ability to approximate perceptual colour distances with Euclidean distances. More on choosing colour spaces can be found in [CM97]. Two vertices i and j are neighbours if $(i, j) \in \mathcal{K}$. The 1-ring neighbourhood of a vertex i, $\mathcal{N}(i)$ is the set of vertices that are neighbours to i. For more discussion, see [Spa94].

We choose *edge collapse* as our basic operation in simplification. We prioritize the edges based on approximated *error* introduced by collapsing them; the approximation scheme is elaborated in the next subsection. The algorithm uses a bi-directional priority queue to manage, query and update the edges based on their associated error values. It picks the edge with the minimal error value at the start of each iteration as well as updates the error value of the edges affected by the edge collapse operation. To collapse an edge, we merge the two end points to one point, assign to it a new set of features (position as well as colour), and remove appropriate faces and edges during the process. One collapse removes one vertex, at most two faces, and at most three edges. We coarsen the initial mesh through a sequence of edge collapsing operations until a certain error threshold is reached. The resulting mesh is called the base mesh.

3.2. Novel Quadric Error Metric

Many cost metrics have been proposed to measure the error caused by removing elements from a mesh. Most metrics approximate volume shrinkage by computing the sum of a set of squared distances. In [Hop96], these squared distances are computed over a set of sample points on the original mesh to the approximating mesh; in [LT98, GH97, Hop99] the distances are computed from a target vertex to a set of planes spanned by its neighbourhood. The latter metric can be compactly represented as a quadric [GH97, Hop99]. Beside geometric error, quadric error metrics [GH97, Hop99] incorporate the attribute errors. Vertex attributes can be of arbitrary dimension such as colour channels.

Neither [Hop99] nor [GH97] is ideal in the image mesh scenario as we stated in the previous section. We choose to follow Hoppe's metric definition [Hop99] for its accuracy, efficient memory usage, and explicit separation between the geometry and the attributes domain. Based on this, we propose a novel *flattened* quadric error metric to deal with 5 dimensional image meshes that have flat geometry domain. Instead of volume distortion, we measure the amount of vertex movement because it better captures the shape distortion on a planar mesh.

Each face f of the original mesh defines a quadric as the sum [Hop99]:

$$Q^f(\mathbf{v} = (\mathbf{p},\mathbf{s})^{\mathbf{T}}) = Q^f_p(\mathbf{v}) + \sum_{j=1}^{m} Q^f_{s_j}(\mathbf{v})$$

Where $Q^f_p(\mathbf{v})$ represents the *geometric error* while $Q^f_{s_j}(\mathbf{v})$ represents the *attribute error* for any of the m attribute channels. We use the definition of attribute error in [Hop99], however we define the geometry quadric to be the squared distance from $\mathbf{p}$ to the geometric centroid $\mathbf{t} = (x_t, y_t)^T \in R^2$ of f. Hoppe's definition measures the distance between a 3D point and the plane defined by f. It is always zero for meshes with planar geometry domain. At places where colour varies subtly, such as noisy textures or the background, decimation driven by this error will happen in random order and produce triangles of suboptimal quality (see figure 7 in section 3.4). By replacing Q^f_p with an isotropic spherical quadric, we penalize sharp shape variation. Our geometric quadric $\mathbf{v}^{\mathbf{T}}\mathbf{A}\mathbf{v} + \mathbf{b}^{\mathbf{T}}\mathbf{v} + c$ is as follows:

$$Q^f_p = (\mathbf{A},\mathbf{b},c) = (\left(\begin{array}{c|c} I & \ddots 0 \ddots \\ \hline \ddots 0 \ddots & \ddots 0 \ddots \end{array}\right), \left(\begin{array}{c} -\mathbf{t} \\ \hline 0 \end{array}\right), \mathbf{t}^T\mathbf{t})$$

where the line divisors mark the first 2 rows/columns. Summing all quadrics together yields $Q^f = (\mathbf{A},\mathbf{b},c) =$

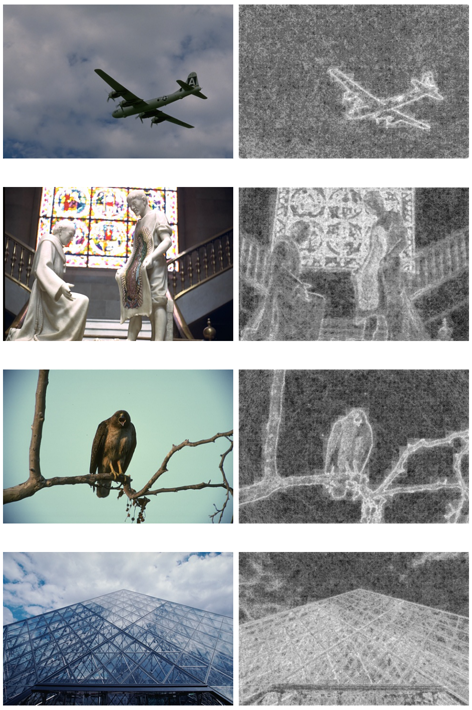
$$(\left(\begin{array}{c|c} I + \sum_j \mathbf{g}_j\mathbf{g}_j^T & -\mathbf{g}_1 \dots -\mathbf{g}_m \\ \hline -\mathbf{g}_1^T & \\ \vdots & I \\ -\mathbf{g}_m^T & \end{array}\right), \left(\begin{array}{c} -\mathbf{t} + \sum_j d_j \mathbf{g}_j \\ \hline -d_1 \\ \vdots \\ -d_m \end{array}\right), \mathbf{t}^T\mathbf{t} + \sum_j d_j^2)$$

Where d_j (offsets) and $\mathbf{g}_j, j \in 1...m$ (gradients) follow the definitions in [Hop99]. Each vertex $\mathbf{v}$ of the original mesh is assigned the sum of quadrics on its adjacent faces weighted by face area [Hop99]: $Q^v(\mathbf{v}) = \sum_{v \in f} \text{area}(f) \cdot Q^f(\mathbf{v})$ Each edge e is assigned a quadric that is the sum of vertex quadrics of its two endpoints. The new vertex introduced by edge collapse (after two vertices are removed) is assigned the position and attribute that minimizes the edge quadric (the minimizer). The minimum value of the quadric is defined as the error for edges. We use SVD to find the minimizer. In singular cases we choose the minimizer among the midpoint and the two endpoints. To validate the claim that our metric approximates the distortion of the image similar to human perception, we visualize the frequency of each pixel *visited* by the edge collapse operator (see figure 2). A pixel is counted as being *visited* once when its corresponding vertex either appears as an endpoint of a collapsing edge or requires re-projection (described in section 3.3) because it was previously projected onto the neighbourhood of the collapsing edge. We expect higher *visit counts* in areas of near constant colour and low *visit counts* where colour changes rapidly. Figure 2 shows grayscale coded *visit maps* (the less visited the brighter) for example images. This visually validates that our design of the quadric is a good approximation of the shape and colour distortion, and it reflects the local colour gradient in the image.

3.3. Parameterization

One of our main contributions is that we construct a globally continuous parameterization for image pixels. The input image is a 5D 2-manifold and we seek an almost isometric parameterization. We construct the parameterization as a mapping from a pixel (vertex $\mathbf{v}$ in the initial mesh) to a point in a base mesh face $\mathbf{v}' = \alpha\mathbf{v}_i + \beta\mathbf{v}_j + \gamma\mathbf{v}_k$, where (i, j, k) is the point's resident face in the base mesh, and α, β, and γ are its barycentric coordinates. This mapping is constructed and maintained through projection along with the simplification as shown in figure 3.

During the process of projection, 5D neighbourhoods are flattened to 2D. We use the same conformal map, z^{α}, as in [LSS*98]. The angles and distances are computed among 5D vectors using inner products. The discrete conformal mappings minimize angle distortion. Such angle based flattening also preserves relative areas of the triangles within a neighbourhood.



(a) Original (b) Visit map

Figure 2: *Visit maps of images of natural scenes [MFTM01] reveal the underlying structure in the images and visually validate our metric.*

There are two scenarios in the process of parameterization, *initial projection* and *reprojection*. Initial projection happens when collapsing an edge, $e = (i, j) \in \mathcal{K}$, and there is no point previously projected into any of the adjacent facets of its endpoints. In this case, we establish a bijection between both its endpoints, $\mathbf{v}_i$ and $\mathbf{v}_j$, to points in some remaining faces after collapsing. This is achieved by flattening the one-ring neighbourhood $\mathcal{N}(i)$ and $\mathcal{N}(j)$ of each endpoint, re-triangulating the region in 2D, and finding the resident face as well as the barycentric coordinates. When flattening $\mathcal{N}(i)$, we substitute the geometry and colour of $\mathbf{v}_j$ with the minimizer of the edge quadric; the reverse situation is treated analogously. The new vertex created after collapsing by merging $\mathbf{v}_i$ and $\mathbf{v}_j$ is placed at the minimizer.

In case there are vertices previously projected into these neighbourhoods, we need to update their parameterization, i.e. reproject their resident faces and barycentric coordinates (figure 3). In the figure, the endpoints $\mathbf{v}_i$ and $\mathbf{v}_j$ are blue; the minimizer, i.e. the new vertex after collapsing, is red. The small dots represent the vertices that were previously projected. If there are projected vertices in faces in the shared region, $\mathcal{N}(i) \cap \mathcal{N}(j)$, we need to first reproject them (figure 3 a). To do so, we flatten the shared region using the minimizer as the pivot (figure 3 b). Figure 3 b shows the white region in the thumbnail figure over the incoming arrow from a. It shows the region after been flattened in 2D and the dashed line originating from the pivot indicates one of the local coordinate frame directions. We use the previous parameterization to find the 2D projections for all projected vertices in this region. New parameterization is computed based on the updated triangulation (figure 3 c). In the next step we split the shared region into two disjoint neighbourhoods (split figure 3 d into e and f along the red edges). We again use conformal maps on both neighbourhoods and update the parameterization for the projected vertices in them the same way as described above (figure 3 (e, g) and (f, h)). Figure 3 e and f show the flattened neighbourhoods and the frame directions (dashed lines). When the procedure is complete, all previously projected vertices as well as the endpoints have a new parameterization in the simplified mesh. However, the new vertex is *not* directly associated with any pixel, hence it is flagged and is not projected in the future.

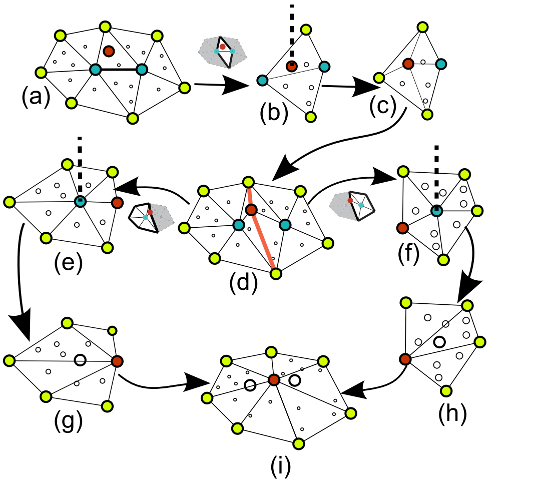


Figure 3: *Illustration of steps of our algorithm showing edge collapse and vertex projection. See the text for details.*

3.4. Results and Discussion

Our scheme works with the whole image (figure 4) as well as an image cut-out of arbitrary shape and topology (figure 5). Figure 5 b-d show multiresolution base meshes created using different error thresholds.

There are many ways to visualize and evaluate the parameterization. We interpolate the parameters in figure 1 (bottom row) to qualitatively assess the smoothness of the parameterization. We also consider pixels that are projected to the same base mesh face as forming one patch/cluster. The boundaries of these patches are shown in figure 5 e and f. During parameterization, different weights can be assigned to spatial and feature domains. Figure 5 e and f compare the patch boundaries when different colour weights are used. The difference is elaborated later in this section.

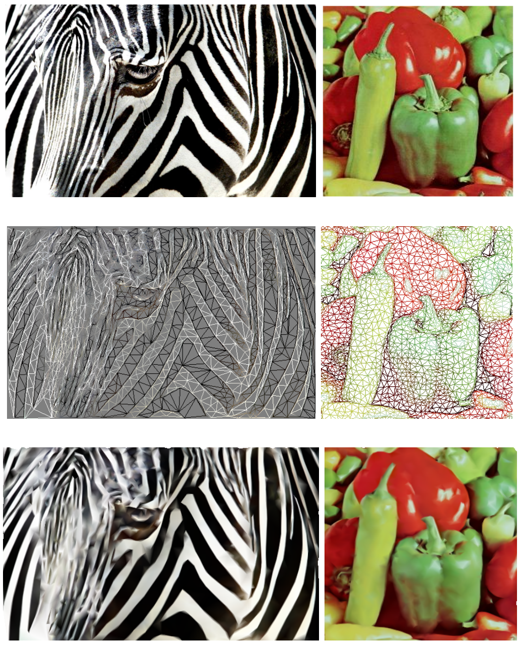
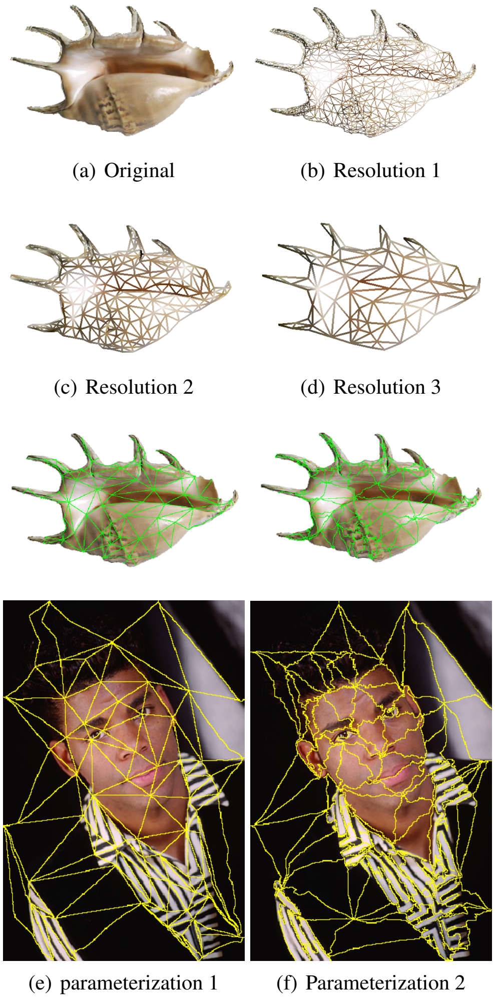


Figure 4: *Top row is original images of* Zebra *and* Bell peppers. *Middle row is the base mesh and bottom row is the base mesh rendered with OpenGL shading resulting in images close to the original. The base meshes for the two images were created with an optimization for colour distortion to preserve the features, black and white stripes in the* Zebra *and pepper boundaries in the* Bell peppers.

Running Time: The proposed algorithm runs in amortized $O(N \log N)$ time, where N is the number of pixels in the input image. The initial mesh has a constant ratio between the number of edges and vertices, so N can be considered as the number of edges in asymptotic analysis. The process stops when a user specified error threshold is reached. For each edge, we query and update the priority queue using $O(\log N)$ time. The calculation of the quadric minimizer takes constant time to solve a linear system with fixed dimension. To update the parameterization, we map all the vertices in the neighbourhood to new facets. If the maximum cluster size, i.e. the number of pixels mapped to a face in the base mesh is specified by the user, this operation is also constant. Hence, the algorithm runs in $O(N \log N)$ time. When the maximum cluster size is not fixed, we have amortized $O(N \log N)$ time complexity as the time spent on the priority queue decreases. We use CGAL library [cga] in our implementation. Detail run time statistics are presented in table 1. A set of square images are tested; the stopping error is the same for all images in our experiment. The computer used has the following configuration: Intel Xeon 3.33GHz; 4GB RAM; NVIDIA Quadro FX 3700; Win 7 64b. We use SVD as our least squares solver in our research implementation for its ability to cope with singularities; Time statistics using LU decomposition is also presented in table 1. During the process, an edge collapse operation is retracted if it introduces degenerated elements in geometry and/or parameter domain.



(a) Original (b) Resolution 1

(c) Resolution 2 (d) Resolution 3

(e) parameterization 1 (f) Parameterization 2

Figure 5: Shell image. *(a) original cut out. (b) - (d) show base meshes at different resolutions. (e) and (f) are parameterizations (patch view) of the same base mesh (d). The patch shapes are noticeably different when different weights are applied for projection. (e) both the shell and the man image place zero weight on the colour domain and produces trivial parameterization that provides no more information than the base mesh; (f) places a higher weight on the feature domain hence its boundaries are aligned with colour features in the original image, making its clustering useful.*

Resolution	Collapsing		Projection
	SVD	LU	Time
32	0.198	0.105	0.098
64	0.915	0.631	0.753
128	4.71	2.32	3.43
256	18.8	10.78	14.7
512	102.6	42.38	70.3

Table 1: *Performance statistics, all times are measured in seconds.*

Reconstruction Colour Error: Recall that one of our aims is to produce a parameterized base mesh with a low reconstruction error in the colour space. Therefore our analysis creates base meshes at varying resolutions and compares the reconstruction colour error and the quality of triangulation against that of Hoppe's. We measure the accuracy quantitatively by evaluating RMS error in colour values between the original and the reconstructed images. We compare our reconstruction error with the simplified meshes produced using our implementation of Hoppe's metric [Hop99] (we replaced its original RGB colour space to $L^*u^*v^*$). We show images of *Lena* and the *Mandrill* and the corresponding error plots in figure 1 and figure 6 respectively comparing ours with the base mesh produced using Hoppe's metric [Hop99]. The topmost row in figure 1 shows the original images; the second row shows Hoppe's base meshes while the third row shows ours. Gouraud shading is used in these images. The bottom row is generated as follows. Our parameterization maps each pixel to a point on the base mesh. Let us assume pixel (0, 0) is mapped to a base mesh face (i,j,k) with barycentric coordinates $(\alpha, \beta, \gamma(1-\alpha-\beta))$. We then assign to pixel (0, 0) a new interpolated colour of $\alpha c_i + \beta c_j + \gamma c_k$. c_i, c_j and c_k are the colours of the base mesh vertices respectively. This enables us to compute the RMS colour error for our parameterization. In the RMS error plots (figure 6) the blue and the red belong to Hoppe's and ours respectively (second and third row respectively in figure 1). These are very close, with ours consistently better at most resolutions. The plot in the green colour is the RMS error of the images in the fourth row.

Triangulation Quality: Many applications need well-shaped, *round* triangles in order to prevent them from running into numerical problems, e.g. numerical simulations based on FEM and image editing [BPK*07]. For this purpose, *round* or isotropic triangles are needed, e.g., the ratio of the radius of the circumcircle to the shortest edge should be as small as possible as well as the aspect ratio should be close to one and average vertex valence should be close to six.

We compare our base meshes with Hoppe's in figure 7. Our metric produces base mesh with better triangulation at such regions due to the fact that drastic shape distortion is penalized while Hoppe's metric is affected only by colour changes and produces suboptimal triangulation at regions *false features* present. Unlike salient and pronounced curvilinear features, these subtle colour changes (false features) are likely contribute to artifacts and random noise during image authoring. For the image shown in figure 7, the average valence of vertices in Hoppe's output is 5.919 while ours is 5.923. For edge ratios, our average is 1.69 versus 1.97 for Hoppe's. Histograms of the aspect ratios and radius ratios are plotted in figure 8. Based on these metrics, it is clear that our algorithm creates better quality triangulation.

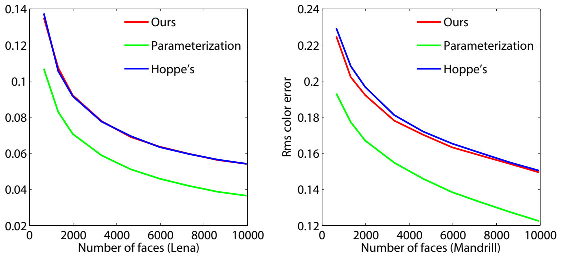


Figure 6: *RMS reconstruction error plots. Ours closely follows Hoppe's. The green curves show the RMS error for the image generated based on our parameterization and performs exceedingly well in maintaining the original image information.*

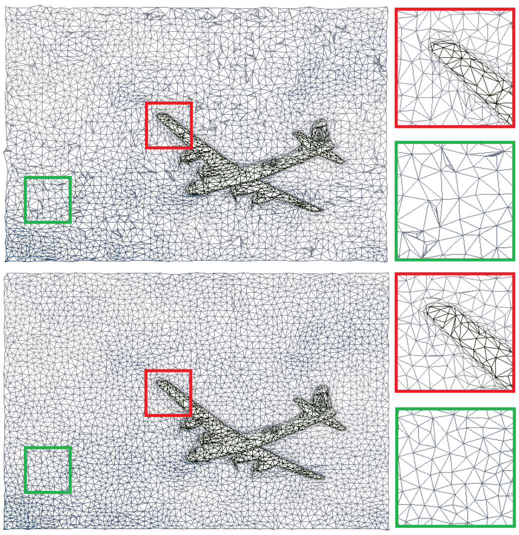


Figure 7: *Visual comparison of the triangulation quality of Hoppe's (top) and ours (bottom). Red boxes show exploded view that both algorithms preserver edge information while the green boxes show our algorithm generates better triangulation. The original image is shown in figure 2.*

Parameterization smoothness: The smoothness of our global parameterization comes through in figure 1 and fig-

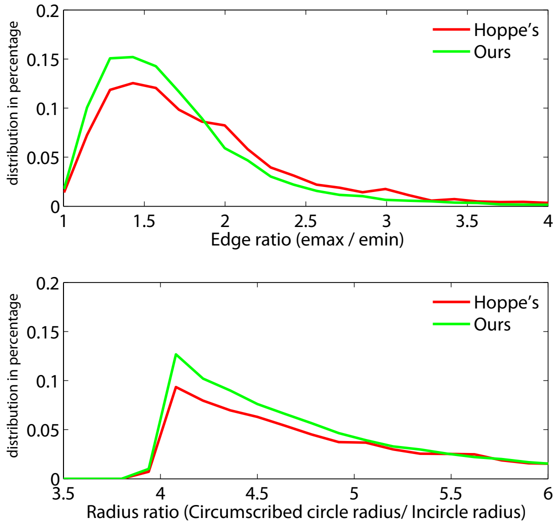


Figure 8: *Triangulation quality histograms.*

ure 6. We note that there are no visible seams between neighbouring patches and the overall colour distortion is very small for examples in figure 1. Such parameterization proves useful in propagating edits in image warping (section 4.2).

The parameterization is adjustable. As we pointed out, different weights can be placed on the colour component during projection to produce parameterizations for applications. When colour component gets zero weight, the projection degenerates to a *trivial* mapping using only the spatial information of the pixel (figure 5 e). Figure 5 e shows that the boundaries of the patches (defined at the beginning of section 3.4) follow the edges of the base mesh. This trivial parameterization is similar in concept with the texture mapping in [COM98]. When a proper weight is used, the boundaries of these patches are curved and follow the salient feature edges in the images (figure 5 f). This is useful as initial over-partition for segmentation (section 4.1).

4. Applications

In this section, we demonstrate the use of our algorithm in a couple of image processing and computer vision tasks.

4.1. Segmentation

Image segmentation divides the image into meaningful pixel regions at object level [BVMP04, CM02]. We compute for each patch, S, defined in section 3.4 a weighted average pixel colour $c = \sum_{k \in S} \min(\alpha, \beta, \gamma) \cdot \mathrm{c}(k)$, where $\mathrm{c}(k)$ is the colour of vertex k, while the weight, $\min(\alpha, \beta, \gamma)$, indicates the importance of a pixel within S. The inner most pixel has the weight of one third while the border pixels have zero weight. The difference of weighted average colours is used to measure the distance between two neighbouring clusters. Pairs with distances smaller than a threshold are merged.

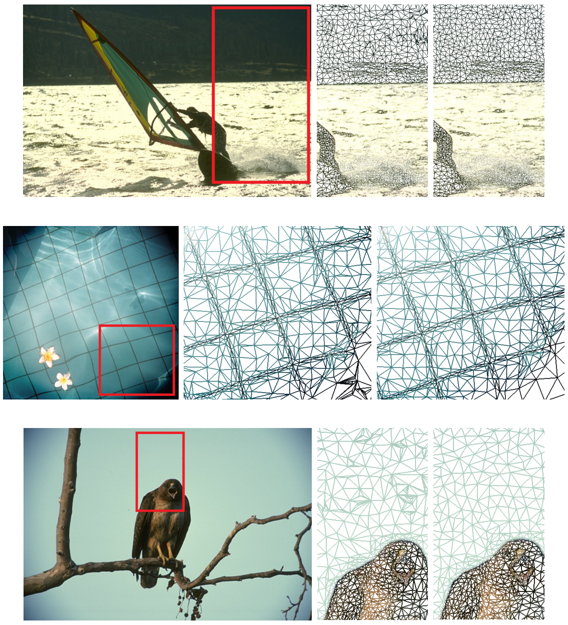


Figure 9: *Triangulation results (each row from left to right): original image, a small portion of the simplified image using Hoppe's and our metric respectively. All three highlight the fact that our metric produces better triangulation.*

Figure 10 shows a comparison of our result with mean shift [CM02] and Histogram multithresholding [BVMP04]. Our method identifies the textured background pad as a whole while isolates the ring. More results on natural scenes are shown in figure 11.

The vision community has seen recently an increasing interest in over segmentation. Over segmentation decomposes an image into much smaller patches of pixels of similar colour compared to the objects found by segmentation. State-of-art over segmentation algorithms, such as TurboPixels [LSK*09], strive for roughly round shaped patches of similar size with boundary pixels aligned with the salient feature lines. These are desirable in applications such as bottom-up segmentation and segment-based stereo matching and reconstruction. With a lower threshold, we produce over segmentation results with our clusters of pixels (figure 5f). The lack of compactness in our method can be remedied by constraining the maximal number of pixels in a patch in the simplification process. However, an extensive investigation and evaluation in both fields is out of the scope of this paper and is planned as future work.

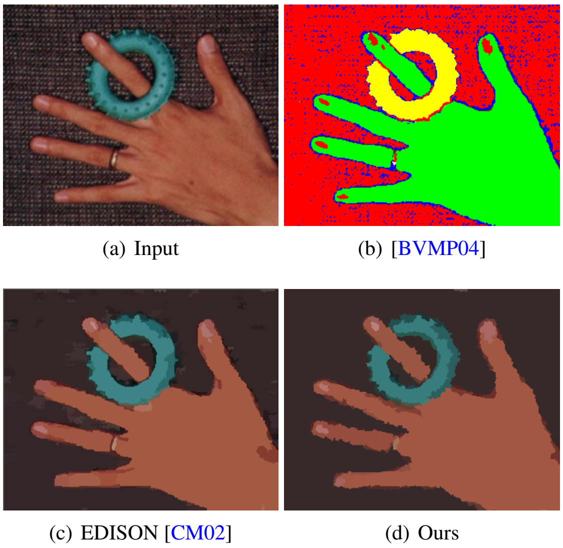


Figure 10: Hand. *Our method deals with the background better while at the same time manages to isolate the finger nails.*

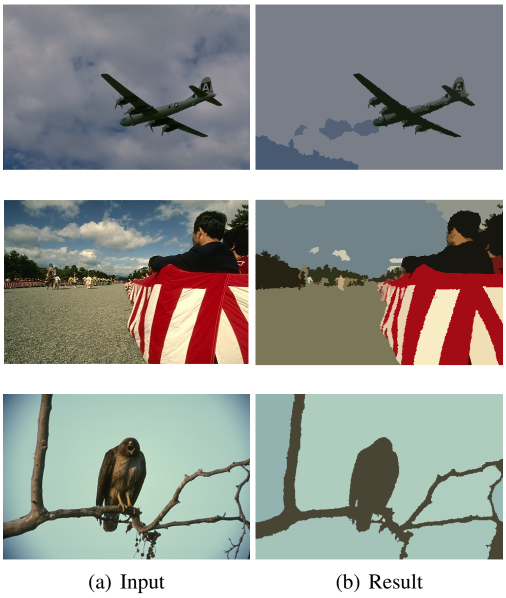


Figure 11: *Some more results of image segmentation (data from [MFTM01]).*

4.2. Multiresolution Image Warping

The image editing and morphing problem has been tackled from multiple angles in graphics and animation. The most popular approach is to construct a cage over the target shape and deform the image content according to user applied rigid transformations on the handles of the cage (its vertices and edges). Such algorithms include As-Rigid-As-Possible Skeleton manipulation [IMH05], moving least squares based image manipulation [SMW06], and bounded biharmonic weights [JBPS11].

Our approach combines the processes of fitting multi-resolution skeletons to the target content (through feature aligned simplification at various resolutions) and assigning smooth localized weights to pixels (using tracked parameterization). At any given level, the user operates on the skeleton and the edits are transferred to the pixels based on the parameterization to produce smooth and localized motions in the image. When a user places an edit on a handle, the pixels that are projected in its neighbourhood get affected in proportion to their corresponding barycentric coordinates. In figure 12 we show an image of a *steering wheel* that has been masked out (and therefore the resulting base mesh has a *genus* > 0). The handle can be stretched at the coarsest level (second row), however, deforming the handles (third row) require a medium level base mesh and buffing up the spokes at the center of the wheel requires an even higher level of resolution (fourth row). Editions are circled in red. Seamless transition between different resolutions allows a user to greatly minimize pre and post processing and a cumbersome relationship modeling between multiple resolutions. Our method handles content of arbitrary shapes and complex topologies with high genus. Our method also copes well with raster texture details as shown in figure 13 and 14.

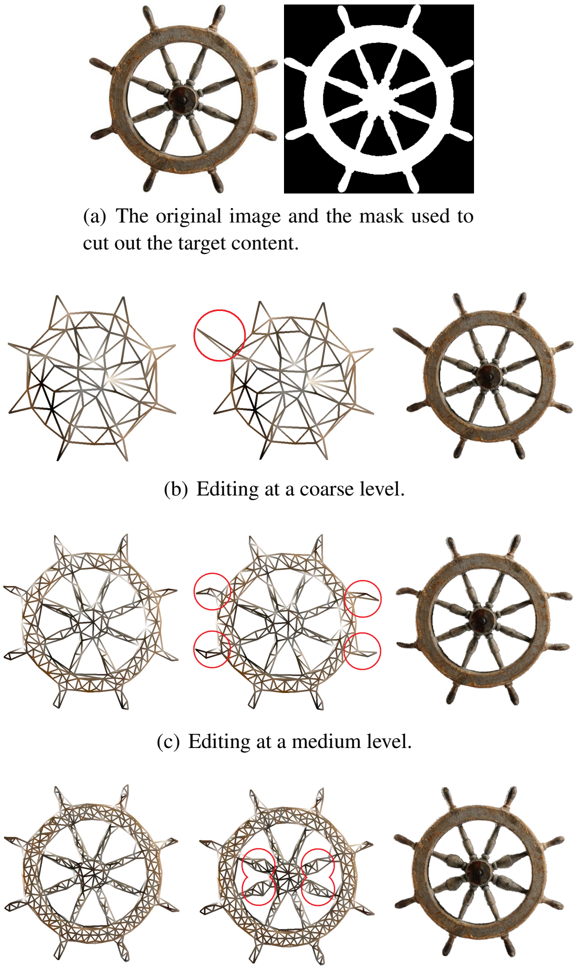
5. Conclusions and Future Work

We proposed a method to convert raster images of arbitrary topology to a coarse triangle mesh representation that has a low colour reconstruction error, produces good quality triangulation along with a non-trivial globally continuous parameterization for each pixel in the original image that is useful in clustering, segmentation and editing. We do this by developing a new quadric error metric suitable for colour images that is sensitive to both geometry and colour. It allows for different weights to be assigned to geometry and colour, making it extremely flexible for a variety of applications. We also present the concept of visit maps as a way to visually validate the metric. The parameterization is useful in image segmentation and editing.

For our future work, we plan to improve our segmentation results by applying ultra metric contour maps [AMFM09] based on our parameterization results and explore more applications that can benefit from our scheme, such as dynamic interactive editing.

References

[ACSE05] Attali D., Cohen-Steiner D., Edelsbrunner H.: Extraction and simplification of iso-surfaces in tandem. In *Proceedings of the third Eurographics symposium on Geometry processing* (2005), Eurographics Association, pp. 139–es. 3



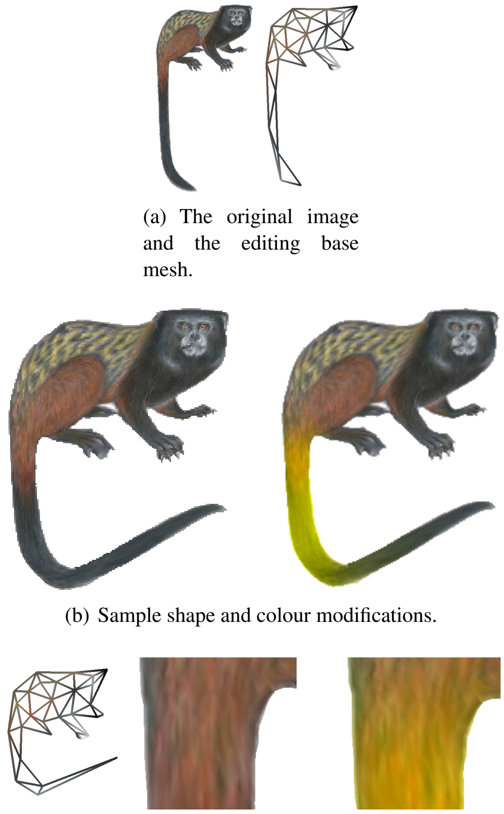
(a) The original image and the mask used to cut out the target content.

(b) Editing at a coarse level.

(c) Editing at a medium level.

(d) Editing at a finer level.

Figure 12: Steering wheel*: an example of multi-resolution image warping using our base mesh and parameterization. Changes made to the base meshes are circled in red.*



(a) The original image and the editing base mesh.

(b) Sample shape and colour modifications.

(c) The control mesh for the shape modification and a exploded view of the colour transition

Figure 13: Monkey*: an example of editing.*

[AMFM09] Arbeláez P., Maire M., Fowlkes C., Malik J.: From contours to regions: An empirical evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 2294–2301. 9

[BPK*07] Botsch M., Pauly M., Kobbelt L., Alliez P., Lévy B., Bischoff S., Rössl C.: Geometric modeling based on polygonal meshes. In *ACM SIGGRAPH 2007 courses* (2007), ACM, p. 1. 7

[BVMP04] Busin L., Vandenbroucke N., Macaire L., Postaire J.: Color space selection for unsupervised color image segmentation by histogram multithresholding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04). Singapore* (2004). 8, 9

[cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org. 6

[CM97] Comaniciu D., Meer P.: Robust analysis of feature spaces: color image segmentation. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)* (Washington, DC, USA, 1997), IEEE Computer Society, p. 750. 3

[CM02] Comaniciu D., Meer P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* (2002), 603–619. 3, 8, 9

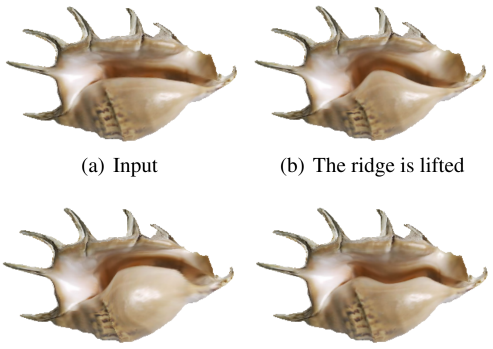
[COM98] Cohen J., Olano M., Manocha D.: Appearance-preserving simplification. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM, pp. 115–122. 2, 8

[GH97] Garland M., Heckbert P.: Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., p. 216. 2, 3, 4

[GZ05] Garland M., Zhou Y.: Quadric-based simplification in any dimension. *ACM Trans. Graph. 24* (April 2005), 209–239. 2, 3

[Hop96] Hoppe H.: Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 99–108. 3, 4

[Hop99] Hoppe H.: New quadric metric for simplifying meshes with appearance attributes. *Visualization Conference, IEEE 0* (1999), 10. 1, 2, 3, 4, 7



(a) Input

(b) The ridge is lifted

(c) Based on (b) the body highlight is shifted left

(d) The ridge is wriggled using a finer control

Figure 14: Spider Shell*: an example of editing.*

[HPS08] HORMANN K., POLTHIER K., SHEFFER A.: Mesh parameterization: theory and practice. In *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses* (New York, NY, USA, 2008), ACM, pp. 1–87. 1, 2

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J.: As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 1141. 9

[JBPS11] JACOBSON A., BARAN I., POPOVIĆ J., SORKINE O.: Bounded biharmonic weights for real-time deformation. In *ACM SIGGRAPH 2011 papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 78:1–78:8. 9

[KLS03] KHODAKOVSKY A., LITKE N., SCHRODER P.: Globally smooth parameterizations with low distortion. *ACM Transactions on Graphics 22*, 3 (2003), 350–357. 2

[KŠ09] KUKAR M., ŠAJN L.: Improving Probabilistic Interpretation of Medical Diagnoses with Multi-resolution Image Parameterization: A Case Study. In *Proceedings of the 12th Conference on Artificial Intelligence in Medicine: Artificial Intelligence in Medicine* (2009), Springer, p. 145. 1

[Lee00] LEE A.: Building your own subdivision surfaces. *Game Developer's Journal 35*, 9 (2000). 2

[LHM09] LAI Y.-K., HU S.-M., MARTIN R. R.: Automatic and topology-preserving gradient mesh generation for image vectorization. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), ACM, pp. 1–8. 3

[LIL06] LEE Y., IM J., LEE S.: Progressive Images: Applying Mesh Processing to Images. *International Journal of Shape Modeling 12*, 2 (2006), 193. 3

[LL06] LECOT G., LÉVY B.: ARDECO: Automatic region detection and conversion. *Rendering Techniques 2006* (2006), 349. 3

[LSK*09] LEVINSHTEIN A., STERE A., KUTULAKOS K., FLEET D., DICKINSON S., SIDDIQI K.: Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 31*, 12 (2009), 2290 –2297. 8

[LSS*98] LEE A. W. F., SWELDENS W., SCHRÖDER P., COWSAR L., DOBKIN D.: Maps: multiresolution adaptive parameterization of surfaces. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM, pp. 95–104. 1, 2, 3, 4

[LT98] LINDSTROM P., TURK G.: Fast and memory efficient polygonal simplification. In *Proceedings of the conference on Visualization'98* (1998), IEEE Computer Society Press, p. 286. 4

[Lue01] LUEBKE D.: A developer's survey of polygonal simplification algorithms. *IEEE Computer Graphics and Applications 21*, 3 (2001), 24–35. 2

[MFTM01] MARTIN D., FOWLKES C., TAL D., MALIK J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision* (July 2001), vol. 2, pp. 416–423. 5, 9

[RFM05] REN X., FOWLKES C., MALIK J.: Scale-invariant contour completion using conditional random fields. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 2, IEEE, pp. 1214–1221. 3

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 540. 9

[Spa94] SPANIER E.: *Algebraic topology*. Springer Verlag, 1994. 3

[WTL*06] WANG J., TONG X., LIN S., PAN M., WANG C., BAO H., GUO B., SHUM H.: Appearance manifolds for modeling time-variant appearance of materials. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 754–761. 3

[XLY09] XIA T., LIAO B., YU Y.: Patch-based image vectorization with automatic curvilinear feature alignment. *ACM Transactions on Graphics (TOG) 28*, 5 (2009), 1–10. 3