

Multiple Aligned Characteristic Curves for Surface Fairing

Janick Martinez Esturo, Christian Rössl, and Holger Theisel

Visual Computing Group, University of Magdeburg, Germany

Abstract. Characteristic curves like isophotes, reflection lines and reflection circles are well-established concepts which have been used for automatic fairing of both parametric and piecewise linear surfaces. However, the result of the fairing strongly depends on the choice of a particular family of characteristic curves: isophotes or reflection lines may look perfect for a certain orientation of viewing and projection direction, but still have imperfections for other directions. Therefore, fairing methods are necessary which consider multiple families of characteristic curves. To achieve this, we first introduce a new way of controlling characteristic curves directly on the surface. Based on this, we introduce a fairing scheme which incorporates several families of characteristic curves simultaneously. We confirm effectiveness of our method for a number of test data sets.

1 Introduction

Visualization of characteristic curves provides a valuable and important tool for first-order surface interrogation (see [1] for a recent survey). Inspection of characteristic surface curves allows for rating and improving surface design as well as for intuitive detection of surface defects: on the one side, they simulate aesthetic appearance under certain lighting conditions and environment, while on the other hand continuity and smoothness of these curves visualize respective differential properties for surface derivatives.

Characteristic surface curves like reflection lines were originally (and still are) used for interrogation and design of physical models, and the concept is simulated for CAGD models in a virtual environment. Surprisingly these curves are mainly used for interrogation, and only few approaches exist which apply them for surface fairing and design [2–5].

Yet, the proposed methods that take advantage of characteristic curves in this setting all have in common that they only consider a single curve family, i.e., a main direction represented by these curves. This results in an optimized behavior of the curves for this single direction, but — as we will show in this paper — the single direction fairing does in general not also yield an optimized characteristic of all other curve directions at the same optimized location. In fact, our experiments indicate that the reverse is true.

Section 4 presents a new fairing scheme for triangulated surfaces that is capable of incorporating an arbitrary number of families *simultaneously*. Prior

to that, for efficient use of this scheme in practice, in section 3 we develop intuitive methods for real-time curve control, i.e., determining parameters such that specific interpolation or alignment constraints on the surface are fulfilled. These methods allow for interactive and automatic curve specification.

1.1 Related Work

In this paper, we consider isophotes, reflection lines, and reflection circles. All these classes of characteristic curves are illumination curves since every curve originates from light-surface interaction [6, 7].

Isophotes can be regarded as surface curves of constant incident light intensity which were extensively used to detect surface imperfections [8, 7, 1].

The reflection of a straight line on a surface is called reflection line. Just as isophotes, reflection lines possess special properties making them valuable for surface interrogation and surface fairing applications of parametric [2, 3, 9, 4] and piecewise linear surfaces [5]. Recently, [5] applied reflection lines for fairing triangular meshes employing a screen-space surface parametrization. This work provides profound analysis of the arising numerical minimization and careful discretization of the emerging differential operators [10]. It is most similar yet different to this work.

Reflection circles arise from the reflections of concentric circles on a surface similar to reflection lines. Although reflection circles are the more general class of surface curves [11], they haven't been used as thorough as the other more specialized classes in surface-fairing applications. Still, recently [12] argue that a simplified version of reflection circles called circular highlight lines also performs well in surface-fairing applications.

There is vast literature on general surface denoising and fairing methods as well as fair surface design based on polygonal meshes, which we do not consider here but instead refer to a recent survey [13]. Similarly, we do not discuss alternative use of light lines such as surface reconstruction applications (see, e.g., [14]).

2 Characteristic Curves

We use definitions of characteristic curves — isophotes, reflection lines, and reflection circles (see figure 1) — which only depend on the normal directions of the surface and not on its position [11]. This means we assume that both, viewer and light sources (which are lines and circles), are located at infinity. This is a common simplification for various kinds of environment mapping. In the following, \mathbf{e} denotes the normalized eye vector (viewing direction), and $\mathbf{n}(u, v)$ is the unit normal to the surface $\mathbf{x}(u, v)$.

Isophotes are surface curves of constant incident light intensity essentially taking into account Lambert's cosine law or diffuse lighting. Given are eye direction \mathbf{e} and an angle α , then an isophote consists of all surface points $\mathbf{x}(u, v)$ satisfying

$$\mathbf{e} \cdot \mathbf{n}(u, v) = \cos \alpha . \tag{1}$$

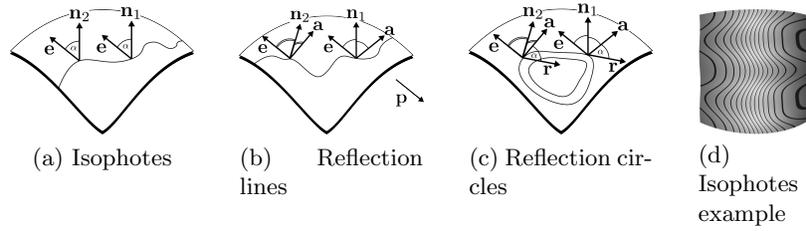


Fig. 1: Definitions of characteristic curves and example for family of isophotes on a wavy cylinder.

Variation of angle α yields a *family of isophotes*.

Reflection lines are surface curves showing the mirror image of a line shaped light source. Given are eye direction \mathbf{e} and a line at infinity defined by its unit normal \mathbf{p} , then a reflection line consists of all surface points satisfying

$$\mathbf{a} \cdot \mathbf{p} = 0 \quad \text{with} \quad \mathbf{a} = 2(\mathbf{e} \cdot \mathbf{n}) \mathbf{n} - \mathbf{e} . \quad (2)$$

Variation of \mathbf{p} along a line at infinity yields a *family of reflection lines*.

Reflection circles [11] provide a generalization of isophotes and reflection lines. They can be considered as mirror images of a family of concentric circles on the surface. Given are \mathbf{e} and a circle at infinity defined by a normalized center direction \mathbf{r} and an angle α , then a reflection circle consists of all surface points satisfying

$$\mathbf{a} \cdot \mathbf{r} = \cos \alpha \quad \text{with} \quad \mathbf{a} = 2(\mathbf{e} \cdot \mathbf{n}) \mathbf{n} - \mathbf{e} . \quad (3)$$

This can be easily transformed to the following condition

$$(\mathbf{e} \cdot \mathbf{n})(\mathbf{r} \cdot \mathbf{n}) = v , \quad (4)$$

where $v = \frac{1}{2}(\cos \alpha + \mathbf{e} \cdot \mathbf{r})$. Reflection circles provide generalizations of other classes of characteristic curves in a sense for $\mathbf{r} = \mathbf{e}$ or $\mathbf{r} = -\mathbf{e}$ they are equivalent to isophotes, whereas for $\mathbf{r} \cdot \mathbf{e} = 2v$ they are equivalent to reflection lines, respectively. *Families of reflection circles* are obtained by either variation of v within range $[-1, 1]$, or variation of \mathbf{a} , or simultaneous variation of both parameters, respectively. In the following we will consider only the first option of varying the scalar parameter v .

3 Characteristic Curve Control

For virtual surface interrogation, e.g., using reflection lines, a simple environment map is sufficient to show families of reflection lines while the user moves the geometric object under inspection. This is simple and intuitive. However, in our setting of surface fairing, we require specification of certain characteristic curves: for a region in focus the user wants to specify curves quickly and intuitively such that they are roughly aligned with a prescribed direction. This setting

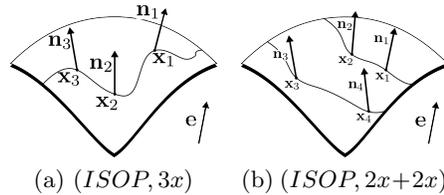
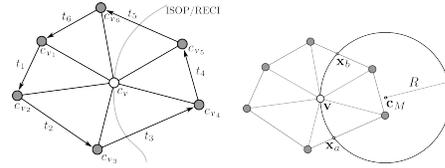
(a) $(ISOP, 3x)$ (b) $(ISOP, 2x+2x)$

Fig. 2: Isophotes alignment methods.

Fig. 3: Local curvature approximation of curve passing through vertex v

provides a problem of its own because the defining parameters of the curves do not directly relate to the resulting pathway on the surface. Moreover plain parameter variation often yields counterintuitive and unexpected results.

In this section we show how to facilitate control of characteristic curves on surfaces in order to enable their intuitive use in practice.

The basic idea of every presented alignment method is to let a user or a (semi-)automatic operation specify a small number of points on the surface which a curve or family of curves shall pass through. Such points will be called *selections*. Then parameters of the curves are calculated by different alignment methods from surface normals in a way that the respective defining conditions are satisfied. Alignment methods differ in the number of required point selections and in their semantics in relation to the curve class. As every alignment method only depends on a small, constant number of points and specifically on the surface normals in these points, they are independent of the complexity of the surface the controlled curves are embedded in. In practice, the user selects by ray intersections with the surface, and selections can be dragged on the surface to fine-tune a curve alignment in real-time. In a similar way, a stencil of selection prototypes can be projected onto the surface for automatic curve control.

3.1 Alignment of Isophotes

It turns out that three selections on the surface are sufficient to define a general isophote passing through these points. We give a closed form expression which yields the parameter \mathbf{e} and $\cos \alpha$.

Proposition 1. *Let \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 be three selections on a smooth surface and \mathbf{n}_1 , \mathbf{n}_2 , and \mathbf{n}_3 the respective unit surface normals. Then*

$$\mathbf{e} = \frac{(\mathbf{n}_1 \times \mathbf{n}_2) + (\mathbf{n}_2 \times \mathbf{n}_3) + (\mathbf{n}_3 \times \mathbf{n}_1)}{\|(\mathbf{n}_1 \times \mathbf{n}_2) + (\mathbf{n}_2 \times \mathbf{n}_3) + (\mathbf{n}_3 \times \mathbf{n}_1)\|} \quad \text{and} \quad (5)$$

$$\cos \alpha = \mathbf{e} \cdot \mathbf{n}_1 = \mathbf{e} \cdot \mathbf{n}_2 = \mathbf{e} \cdot \mathbf{n}_3$$

are the parameters defining an isophote $(\mathbf{e}, \cos \alpha)$ interpolating \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 .

Proof. Let $\mu = \|(\mathbf{n}_1 \times \mathbf{n}_2) + (\mathbf{n}_2 \times \mathbf{n}_3) + (\mathbf{n}_3 \times \mathbf{n}_1)\|$, then it follows from direct algebraic calculation that

$$\mu \cos \alpha = \mathbf{e} \cdot \mathbf{n}_1 = \mathbf{e} \cdot \mathbf{n}_2 = \mathbf{e} \cdot \mathbf{n}_3 = |\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3|. \quad \square$$

In the remainder we refer to this alignment method as $(ISOP, 3x)$, indicating that an isophote is aligned using three selections. Figure 2a illustrates the configuration.

The alignment of a family of isophotes is achieved using two pairs of selections:

Proposition 2. *Let $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_3, \mathbf{x}_4)$ be two pairs of selections on a smooth surface and $(\mathbf{n}_1, \mathbf{n}_2)$ and $(\mathbf{n}_3, \mathbf{n}_4)$ the respective unit surface normals. Then*

$$\mathbf{e} = \frac{(\mathbf{n}_1 - \mathbf{n}_2) \times (\mathbf{n}_3 - \mathbf{n}_4)}{\|(\mathbf{n}_1 - \mathbf{n}_2) \times (\mathbf{n}_3 - \mathbf{n}_4)\|}, \quad (6)$$

$$\cos \alpha_1 = \mathbf{e} \cdot \mathbf{n}_1 = \mathbf{e} \cdot \mathbf{n}_2 \quad \text{and}$$

$$\cos \alpha_2 = \mathbf{e} \cdot \mathbf{n}_3 = \mathbf{e} \cdot \mathbf{n}_4$$

are the parameters defining two isophotes $(\mathbf{e}, \cos \alpha_1)$ and $(\mathbf{e}, \cos \alpha_2)$ of the same family passing through the points $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3, \mathbf{x}_4$, respectively.

Proof. Let $\mu = \|(\mathbf{n}_1 - \mathbf{n}_2) \times (\mathbf{n}_3 - \mathbf{n}_4)\|$, then

$$\mu \cos \alpha_1 = \mathbf{n}_1 \cdot \mathbf{e} = \mathbf{n}_2 \cdot \mathbf{e} = \mathbf{n}_1 \cdot (\mathbf{n}_2 \times (\mathbf{n}_4 - \mathbf{n}_3))$$

and a similar statement holds for $\mathbf{n}_3, \mathbf{n}_4$, and $\cos \alpha_2$, respectively. \square

We call this method $(ISOP, 2x + 2x)$ because two isophotes of the same family are aligned requiring two selections for each curve. Two isophotes of the same family aligned using this method are depicted in figure 2b.

3.2 Alignment of Reflection Lines and Circles

Due to their relative simplicity, isophotes constitute a special case for which closed form solutions to the general alignment can be given. In contrast, general alignment of reflection lines and circles requires root finding of higher order polynomials to determine parameters. Hence, no closed form expressions can be given. Instead, we present a numerical and a constraint approach, respectively.

Numerical Approach. Solutions to the general alignment problem can be obtained numerically by solving systems of nonlinear equations given by curve definitions (2) and (4).

Four selections $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and \mathbf{x}_4 with respective surface normals $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ and \mathbf{n}_4 are sufficient to establish the following system of nonlinear equations. We call this method $(REFL, 4x)$. Its solution for \mathbf{e} and \mathbf{p} yields the parameters defining a single general reflection line on a smooth surface that passes through the four selection points. As both vectors \mathbf{e} and \mathbf{p} are three-dimensional, six equations are required to constitute the system:

$$\begin{aligned} \text{Solve } & (2(\mathbf{e} \cdot \mathbf{n}_1) \cdot \mathbf{n}_1 - \mathbf{e}) \cdot \mathbf{p} \stackrel{\perp}{=} 0, & (2(\mathbf{e} \cdot \mathbf{n}_2) \cdot \mathbf{n}_2 - \mathbf{e}) \cdot \mathbf{p} \stackrel{\perp}{=} 0, \\ & (2(\mathbf{e} \cdot \mathbf{n}_3) \cdot \mathbf{n}_3 - \mathbf{e}) \cdot \mathbf{p} \stackrel{\perp}{=} 0, & (2(\mathbf{e} \cdot \mathbf{n}_4) \cdot \mathbf{n}_4 - \mathbf{e}) \cdot \mathbf{p} \stackrel{\perp}{=} 0, \\ & \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1 & \text{and } \mathbf{p} \cdot \mathbf{p} \stackrel{\perp}{=} 1 \quad \text{for } \mathbf{e} \text{ and } \mathbf{p}. \end{aligned}$$

| Method | Calculation |
|--|---------------------|
| $(ISOP, 3x)$ | closed form |
| $(ISOP, 2x + 2x)$ | closed form |
| $(REFL, 4x)$ | system of equations |
| $(REFL, \mathbf{e} + 2x)$ | system of equations |
| $(REFL, 4x + 2x)$ | system of equations |
| $(REFL, 3x + 3x)$ | system of equations |
| $(REFL, 2x + 2x + 2x + Mid)$ | system of equations |
| $(REFL, 2x + \text{constr})$ | closed form |
| $(RECI, 5x)$ | system of equations |
| $(RECI, 4x + 2x)$ | system of equations |
| $(RECI, 3x + 3x)$ | system of equations |
| $(RECI, 2x + 2x + 2x + 2x)$ | system of equations |
| $(RECI, 2x + 2x + 2x + \text{constr})$ | system of equations |
| $(RECI, 2x + \text{constr})$ | closed form |

Table 1: Alignment methods for characteristic curves.

Compared to the alignment of a single general reflection line, a single general reflection circle requires one extra surface selection \mathbf{x}_5 , \mathbf{n}_5 to establish the system

$$\begin{aligned} \text{Solve } & (\mathbf{e} \cdot \mathbf{n}_1) \cdot (\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_2) \cdot (\mathbf{r} \cdot \mathbf{n}_2), \quad (\mathbf{e} \cdot \mathbf{n}_1) \cdot (\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_3) \cdot (\mathbf{r} \cdot \mathbf{n}_3), \\ & (\mathbf{e} \cdot \mathbf{n}_1) \cdot (\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_4) \cdot (\mathbf{r} \cdot \mathbf{n}_4), \quad (\mathbf{e} \cdot \mathbf{n}_1) \cdot (\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_5) \cdot (\mathbf{r} \cdot \mathbf{n}_5), \\ & \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1 \quad \text{and} \quad \mathbf{r} \cdot \mathbf{r} \stackrel{\perp}{=} 1 \quad \text{for } \mathbf{e} \text{ and } \mathbf{r}. \end{aligned}$$

We refer to this method as $(RECI, 5x)$. A solution of this system defines a reflection circle $(\mathbf{e}, \mathbf{r}, \cos \alpha)$ with

$$\cos \alpha = (2 (\mathbf{e} \cdot \mathbf{n}_i) \mathbf{n}_i - \mathbf{e}) \cdot \mathbf{r}, \quad i \in \{1, \dots, 5\}. \quad (7)$$

Therefore, the reflection circle determined this way interpolates the five selection points $\mathbf{x}_1, \dots, \mathbf{x}_5$.

We provide more variants of systems to align families of both, reflection lines and reflection circles. They are given in appendix A, and all methods are summarized in table 1. The presented systems of nonlinear equations can indeed be solved numerically in real-time. However, the following arguments suggest that they may be less appropriate in practice compared to the closed form solution for isophotes and — more importantly — the subsequently presented constraint alignment method:

- Both, reflection lines and circles possess more degrees of freedom compared to the simpler isophote curves. This is why the resulting curves, although traversing the specified selections on the surface, do not necessarily reflect the alignment the user intended to generate.
- The provided systems do not have a unique solution, as, e.g., both reflection lines, $(\mathbf{e}, -\mathbf{p})$ and (\mathbf{e}, \mathbf{p}) pass through the same given selections. For reflection

circles an analogous example are the parameter combinations of $(\mathbf{e}, \mathbf{r}, \cos \alpha)$ and $(\mathbf{e}, -\mathbf{r}, -\cos \alpha)$.

Numerical methods are well suited for surface interrogation as they enable interactive fine tuning of the alignment. The constraint approach is moreover especially suited for automatic alignment because of the lower number of selection required and the unambiguity of the alignment results.

Constraint Approach. The constraint approach requires only two selections in order to align both, reflection lines or reflection circles on smooth surfaces. Reflection circles are a generalization of reflection lines: setting its $\cos \alpha$ parameter to zero in fact specifies a reflection line with one lost degree of freedom which can be taken advantage of afterwards. We restrict the derivation of the alignment method to reflection lines in the first place and make it applicable for both curve classes by variation of the extra parameter.

Proposition 3. *Let $\mathbf{x}_1, \mathbf{x}_2$ be two selections on a smooth surface and $\mathbf{n}_1, \mathbf{n}_2$ their linear independent normals. Then*

$$\mathbf{e} = \frac{(\mathbf{n}_1 + \mathbf{n}_2)}{\|(\mathbf{n}_1 + \mathbf{n}_2)\|} \quad \text{and} \quad \mathbf{r} = \mathbf{p} = \frac{(\mathbf{n}_1 \times \mathbf{n}_2)}{\|(\mathbf{n}_1 \times \mathbf{n}_2)\|} \quad (8)$$

are the parameters defining a reflection line (\mathbf{e}, \mathbf{p}) as well as a reflection circle $(\mathbf{e}, \mathbf{r}, \cos \alpha = 0)$ passing through the points $\mathbf{x}_1, \mathbf{x}_2$.

Proof. The above can easily be verified as the identity

$$2 \cdot (\mathbf{e} \cdot \mathbf{n}_i) \cdot (\mathbf{r} \cdot \mathbf{n}_i) - (\mathbf{e} \cdot \mathbf{r}) = \cos \alpha = 0, \quad i = 1, 2$$

holds since $\cos \alpha$ was set to zero and $\mathbf{r} \cdot \mathbf{n}_i$ as well as $\mathbf{e} \cdot \mathbf{r}$ is zero by construction. \square

These alignment methods are referred to as *(REFL, 2x + constr)* and *(RECI, 2x + constr)*, respectively. We call the approach constraint as the parameter vectors of eye vector \mathbf{e} and normal \mathbf{p} of the line at infinity are restricted to be perpendicular, so $\mathbf{e} \cdot \mathbf{p} = 0$. Geometrically this means that the eye point at infinity is constraint to the respective line at infinity.

Critical points of characteristic curves. Both approaches may yield surface curves which show *critical points*, i.e., locations of self-intersection. As curves are defined implicitly, we observe a locally non-manifold setting where differential properties such as curvature, which will be subject to optimization, remain undefined. Such special cases occur rarely and should be avoided by alignment methods, they are not critical for iterative optimization, though (see next section).

4 Surface Fairing

We define the goal of our surface fairing method as follows: a smooth surface should be altered by minimal local displacements such that pathways of characteristic curves are straightened and homogenized. Therefore, our aim is to penalize curvature of characteristic curves, and we minimize an error functional which is defined as follows. For a family f of characteristic curves on a surface \mathcal{S} , we obtain an optimized, fair surface \mathcal{S}' as

$$\operatorname{argmin}_{\mathcal{S}'} \left\{ \iint_f \left(\int_{\mathbf{c}} \kappa^2(t) dt \right) d \cos \alpha \right\}, \quad (9)$$

where $\mathbf{c}(t)$ denotes that curve member of the family specified by $\cos \alpha$, and $\kappa(t)$ denotes curvature at location $\mathbf{c}(t)$. Note that this way we formulate the problem for isophotes and reflection circles.

Discrete Setting

Moving to a discrete setting, let \mathcal{C} define a set of discrete families of curves, e.g., specified by a finite set of angles. We consider piecewise linear surfaces \mathcal{M} , i.e., triangles meshes defined by $(\mathcal{V}, \mathcal{E}, \mathcal{F})$, sets of vertices, oriented edges, and faces, respectively. Then we define discrete error functionals of (9) as

$$E(\mathcal{V}) = \sum_{f \in \mathcal{C}} E_f(\mathcal{V}) \quad \text{with} \quad E_f(\mathcal{V}) = \sum_{v \in \mathcal{V}} \kappa_f^2(\mathbf{v}), \quad (10)$$

where \mathbf{v} denotes the position of vertex $v \in \mathcal{V}$. We call E_f *family error* and E *accumulated error*, respectively. Minimizing $E(\mathcal{V})$ by altering vertex positions yields an optimized, fair surface \mathcal{M}' .

Curvature of Characteristic Curves

Each family of characteristic curves defines a piecewise linear scalar field over the surface, i.e., the defining equations (1) or (3), respectively, are evaluated at every vertex. Then members of the family are given implicitly as iso-curves w.r.t. to a certain isovalue. We approximate curvature of such characteristic iso-curves per vertex $v \in \mathcal{V}$ as follows.

We find intersections of the iso-contour $\mathbf{c}(\mathbf{v}) = c_v$ with the edges $(i, j) \in \mathcal{E}_v^1$ bounding the 1-ring neighborhood \mathcal{N}_v of v by linear interpolation between $\mathbf{c}(v_i) = c_{v_i}$ and $\mathbf{c}(v_j) = c_{v_j}$. Figure 3 illustrates the setting. Generally, two intersection points are found in the regular case. We ignore vertex v if no or more than two intersections exist, i.e., the curve intersects itself yielding a critical point at \mathbf{v} . The same applies if no intersection is found, i.e., the curve is restricted to the 1-ring. Neither of these “degenerated” cases imposes any problem to our optimization. From \mathbf{v} and the positions of two intersections, curvature $\kappa_f(\mathbf{v})$ is given as the inverse radius of the interpolating circle. If the intersections are approximately collinear, i.e., circle degenerates to a line, we assume zero curvature.

Local Optimization by Vertex Displacement

In order to minimize the accumulated error E , we iteratively displace vertices along their normal direction. We analyze the local setting for vertex v and its neighborhood. The surface normal $\mathbf{n}(\mathbf{v}) := \mathbf{n}_v$ is approximated as the average of weighted triangle normals, i.e.,

$$\tilde{\mathbf{n}}_v = \sum_{(i,j) \in \mathcal{E}_v^1} (\mathbf{v}_i - \mathbf{v}) \times (\mathbf{v}_j - \mathbf{v}), \text{ and } \mathbf{n}_v = \frac{\tilde{\mathbf{n}}_v}{\|\tilde{\mathbf{n}}_v\|}. \quad (11)$$

Note that \mathcal{E}_v^1 includes all directed (counter-clockwise oriented) edges bounding the 1-ring of v . For simplicity we use an area weighting scheme here, however, applying more sophisticated normal approximation methods (see, e.g., survey [1]) yields similar formulas.

Displacing vertices as $\mathbf{v}' = \mathbf{v} + \epsilon \mathbf{n}_v$ for small scalar ϵ entails recomputation of vertex normals only within the 1-ring of v . It is easy to see that curvatures, however, are effected within the 2-neighborhood \mathcal{N}_v^2 and curvature variation is therefore locally bounded. Consequently, scalar values within the 3-neighborhood \mathcal{N}_v^3 of v have to be considered for computing the global variation of the error induced by the displacement.

Starting from (11), we derive the following expression

$$\tilde{\mathbf{n}}_v(\epsilon) = \sum_{(i,j) \in \mathcal{E}_v^1} \mathbf{v}_i \times \mathbf{v}_j + \epsilon (\delta_{ik} (\mathbf{n}_{v_k} \times \mathbf{v}_j) - \delta_{jk} (\mathbf{n}_{v_k} \times \mathbf{v}_i)) \quad (12)$$

as updated unnormalized normal direction of vertex v after displacement of vertex v_k by $\epsilon \mathbf{n}_{v_k}$. Normalizing $\tilde{\mathbf{n}}_v$ yields the new normal \mathbf{n}_v . The normal of the displaced vertex remains constant.

For fairing, a vertex is iteratively translated in several ϵ -steps as long as a single displacement reduces the global error.

The error evaluation is summarized in algorithm 1, which yields a *globalGain* of E : For each considered family the scalar field of \mathcal{N}_v^3 is established and the resulting curvature approximations of $\{v\} \cup \mathcal{N}_v^2$ is compared to the curvatures a possible translation of \mathbf{v} by $\epsilon \mathbf{n}_v$ would give.

Mesh Fairing

We use the analysis of the local setting to globally minimize the accumulated error $E(\mathcal{V})$ for all vertices (or for those within a region of interest, respectively). We take a randomized and serialized approach which iterates the following steps:

1. Randomly pick a vertex $v \in \mathcal{V}$.
2. Take a binary decision whether a translation direction of \mathbf{v} in direction \mathbf{n}_v or $-\mathbf{n}_v$ makes E decrease; otherwise restart at step 1.
3. Find the most effective displacement of \mathbf{v} by integrating ϵ -steps as long as the global error reduction is of significant magnitude. The step size ϵ is adapted during integration by logarithmical attenuation depending on the error reduction rate.

We terminate the global iteration if no more enhancement can be achieved over a specific number of iterations. Vertices on degenerate curves (critical points, near infinite curvature) do not impose any problem as they will not be picked in step 1. Generally their processing is delayed until their neighborhood is fixed appropriately. Our experiments show that $E(\mathcal{V})$ is effectively reduced at reasonable computational cost (see section 5). We remark that in every local optimization step curvature of characteristic curves is reduced not only for vertex v but also in its 2-neighborhood due to the overlap of the respective curvature stencils. Hence, for optimization within a region of interest, the boundary region is automatically processed such that smooth transition of optimized curves across the boundary is ensured.

Algorithm 1 Accumulated Curvature Gain by Vertex Step

```

1: procedure VERTEXSTEPGAIN( $v, \epsilon, \mathcal{C}$ )
2:    $globalGain \leftarrow 0$ 
3:    $\mathbf{v}_{tmp} \leftarrow \mathbf{v}$ 
4:   for all families  $f \in \mathcal{C}$  do
5:     for all vertices  $v_i \in \{v\} \cup \mathcal{N}_v^3$  do
6:        $c_{v_i} \leftarrow scalarValue_f(\mathbf{n}_{v_i})$ 
7:     end for
8:      $\mathbf{v} \leftarrow \mathbf{v}_{tmp}$ 
9:     for all vertices  $v_i \in \{v\} \cup \mathcal{N}_v^2$  do
10:       $oC_{v_i} \leftarrow approximateCurvature(\mathbf{v}_i)^2$ 
11:    end for
12:    for all vertices  $v_i \in \mathcal{N}_v$  do
13:       $c_{v_i} \leftarrow scalarValue_f(\mathbf{n}_{v_i}(\epsilon))$ 
14:    end for
15:     $\mathbf{v} \leftarrow \mathbf{v}_{tmp} + \epsilon \mathbf{n}_v$ 
16:    for all vertices  $v_i \in \{v\} \cup \mathcal{N}_v^2$  do
17:       $nC_{v_i} \leftarrow approximateCurvature(\mathbf{v}_i)^2$ 
18:    end for
19:    for all vertices  $v_i \in \{v\} \cup \mathcal{N}_v^2$  do
20:       $globalGain \leftarrow globalGain + (nC_{v_i} - oC_{v_i})$ 
21:    end for
22:  end for
23:   $\mathbf{v} \leftarrow \mathbf{v}_{tmp}$ 
24:  return  $globalGain$ 
25: end procedure

```

5 Results

We provide results which demonstrate the effectiveness of our approach. In addition we refer to the supplemental videos which illustrate interactive curve control and surface fairing.

Curve Control. The proposed alignment methods highly facilitate the interrogation of surfaces by characteristic surface curves. Figure 4 gives two examples (see also video). We found $(ISOP, 3x)$ and $(RECI, 2x + \text{constr})$ especially useful for automatic surface fairing applications as they require a low number of selections, are computable in constant real-time and yield stable alignment results.

Single family fairing. For evaluation of our surface fairing method we first consider reconstruction of a smooth surface region which has been perturbed by noise. In this context, we refer to successful “reconstruction” as recovering characteristic curves on the faired surface which should not deviate from their counterparts on the initial smooth surface. In this first experiment we consider only a single family of curves. Figure 5 shows results for both, isophotes (top row) and reflection lines (bottom row), together with color coded curvature of the respective curves of the family before (center right column) and after (right column) optimization. For color coding we use a smooth scheme which maps zero curvature to black and curvatures towards infinity to white. In both cases our goal was achieved within a number of iterations which we plot over the error E_f (logarithmic scale) for isophotes (left) and reflection lines (right). Furthermore, the diagrams illustrate how the error reaches the original value (smooth surface) and the value of the smooth surface undergoing additional fairing (dashed lines).

The fairing of a more complex model of a Chevrolet Corvette C4¹ engine hood by reflection circles, which were aligned by $(RECI, 2x + \text{constr})$, is shown in figure 6. Within 2000 iterations the accumulated error dropped by 75.91% for about 500 optimized vertices; the processing time was 40s. All timings were measured on a 2.2GHz AMD Opteron processor.

Multiple family fairing. One of our main goals is to show that *multiple families* of characteristic curves should be considered simultaneously. So far, only a single family had been used in prior work. We demonstrate that the latter generally yields improvement of this single family only, while other families may improve or not — or may even get worse in appearance. Figure 7 shows an example, where three differently aligned families (using $(ISOP, 3x)$) of isophotes are shown on the initial model of a BMW Z3 engine hood in the top row. The second row shows families resulting from solely fairing the left family: the two other families did not improve in the same way. This is because fairing of a surface by a single curve family does not necessarily improve the overall reflective properties of a surface. A subsequent example shows that the contrary can also be the case. Incorporating all three families into the fairing process gives better overall results, see bottom row. The total processing time for multiple families depends linearly on their number. The accumulated error of all families dropped by 32.5% after fairing the single family using 2000 iterations, however, it dropped by 63.7% fairing all directions. With the model scaled to the unit sphere, the average displacement per vertex is of length $8.33 \cdot 10^{-6}$, hence the induced approximation error to the initial surface is negligible.

¹ Corvette and BMW models from www.dmi3d.com.

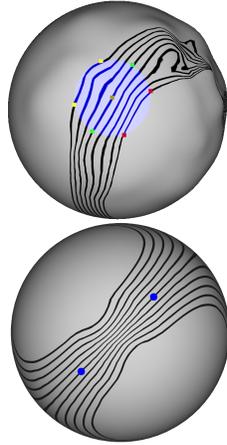


Fig. 4: Alignment examples: (*REFL*, $2x + 2x + 2x + Mid$) on a bumpy sphere (top) and (*RECI*, $2x + constr$) on a minimally bumped sphere (bottom).

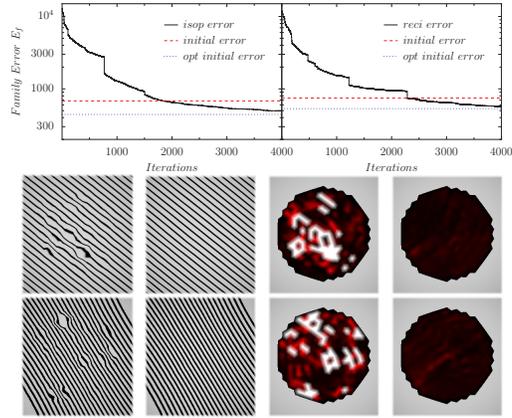


Fig. 5: Fairing of noisy surface by isophotes (left graph, center row) and reflection circle (right graph, bottom row), see section 5

In figure 8 we analyze the fairing of the car roof of a Volkswagen Beetle using multiple families of reflection circles which were automatically aligned by (*RECI*, $2x + constr$) to be uniformly distributed. This example illustrates several different families, and it illustrates several facts: first, the benefits of simultaneous optimization of multiple families, second, the potential corruption of families if only a single other family is considered. In the example the vertical family shows already good quality which degrades when only the horizontal family is optimized. In addition, the behavior of two skew families is shown, in the final experiment they are also considered in optimization. The error is plotted versus the number of iterations for all settings, accumulated error dropped by 32.26%, 41.32% and 58.28%, respectively. We moreover found that no other intermediate family direction showed an imperfect behavior on the surface fairing this way.

Curve class comparison. Both, isophotes and reflection circles, can be fairied by our generic approach. In our experiments cross validation showed comparable performance for both curve classes. We could not affirm the proposition in [12] stating that circular highlight lines are better suited for surface fairing than highlight lines, as all directions are captured.

6 Discussion.

Our results support our claim that simultaneous consideration of multiple curve families is advantageous for surface fairing. Furthermore, we provide new methods for controlling characteristic curves, which haven't been applied in any previous approach.

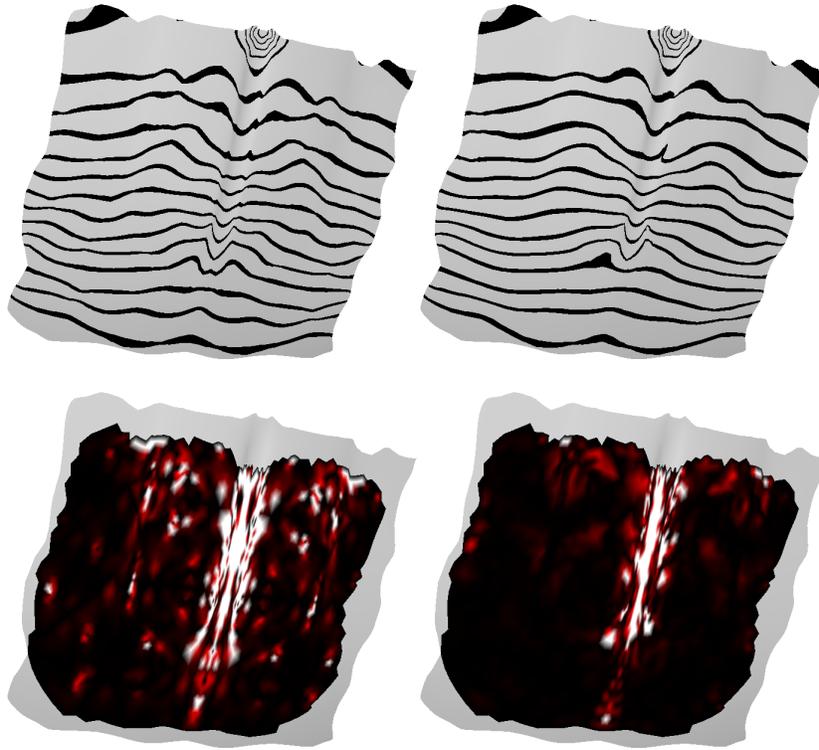


Fig. 6: Reflection circles on a Chevrolet Corvette C4 engine hood before (left) and after (right column) fairing.

Prior work most similar to our method is [5] who considered reflection lines for shape optimization based on triangle meshes. They concentrate on optimizing one single family of reflection lines. The family is provided by the user, control of curve parameters is not discussed. Emphasis on discretization and efficient numerical minimization using screen-space parametrization and other approximations yields a real-time algorithm with some view dependent limitations. In contrast our focus was on new aspects summarized above. Our optimization method uses a far simpler randomized greedy surface optimization which converges to local minima and is far from real-time application. It would be an interesting project for future work to see whether our method could be combined with the minimization framework in [5].

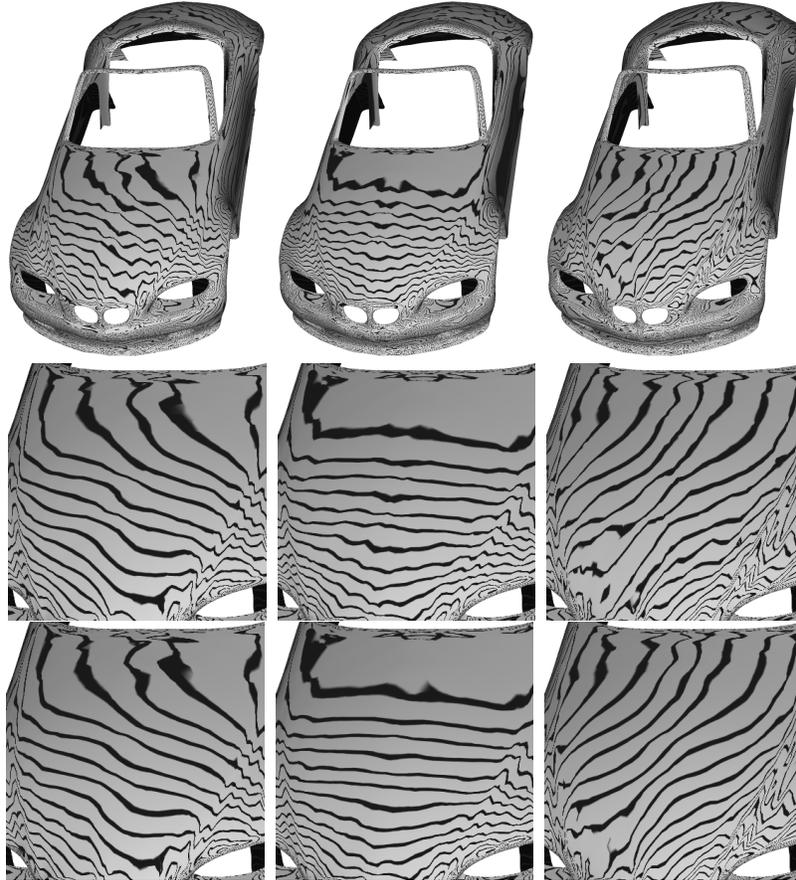


Fig. 7: Fairing of BMW Z3 model hood: Initial three families (first row), only first family faired (second row) and all families faired (third row).

7 Conclusions

In this paper we make the following contributions:

- We showed that the fairing of a particular family of characteristic surface curves (like isophotes, reflection lines, or reflection circles) does not necessarily yield a fairer surface in the sense that other families of surface curves become fairer as well.
- We introduced a number of techniques to align characteristic curves on surfaces by directly placing and interactively moving points on the surface instead of specifying viewing and projection parameters.
- Based on this, we presented an approach for simultaneous fairing of multiple families of characteristic surface curves which gives better results than a single-family fairing.

The following issues remain open for future research:

- Although the whole fairing process can be considered as a preprocess which is carried out once, the performance of the algorithm could be enhanced.
- We have no general solution on the question how many families should be faired simultaneously to get optimal results. Clearly, increasing the number of families enhances the results but also increases the computing time linearly. In all our examples, four families were sufficient to ensure the fairness of all families. However, we do not have a theoretical confirmation of this statement yet.

References

1. Hahmann, S., Belyaev, A., Buse, L., Elber, G., Murrain, B., Rössl, C.: 1. Mathematics and Visualization. In: Shape Interrogation. Springer, Berlin, Germany (2008) 1–52
2. Klass, R.: Correction of local surface irregularities using reflection lines. *Computer-Aided Design* **12** (1980) 73–77
3. Kaufmann, E., Klass, R.: Smoothing surfaces using reflection lines for families of splines. *CAGD* **20** (1988) 312–316
4. Loos, J., Greiner, G., Seidel, H.P.: Modeling of surfaces with fair reflection line pattern. In: Shape Modeling and Applications. (1999)
5. Tosun, E., Gingold, Y., Reisman, J., Zorin, D.: Shape optimization using reflection lines. In: Symposium on Geometry Processing. (2007) 193–202
6. Guid, N., Oblonsek, C., Zalik, B.: Surface interrogation methods. *Computers & Graphics* **19** (July 1995) 557–574
7. Hagen, H., Schreiber, T., Gschwind, E.: Methods for surface interrogation. *IEEE Visualization* (1990) 187–193
8. Poeschl, T.: Detecting surface irregularities using isophotes. *CAGD* **1** (1984) 163–168
9. Theisel, H., Farin, G.: The curvature of characteristic curves on surfaces. *IEEE Computer Graphics and Applications* **17** (1997) 88–96
10. Grinspun, E., Gingold, Y., Reisman, J., Zorin, D.: Computing discrete shape operators on general meshes. *Eurographics* **25** (2006) 547–556
11. Theisel, H.: Are isophotes and reflection lines the same? *Comput. Aided Geom. Des.* **18** (2001) 711–722
12. Nishiyama, Y., Nishimura, Y., Sasaki, T., Maekawa, T.: Surface fairing using circular highlight lines. *Computer-Aided Design and Application* **7** (2007) 405–414
13. Botsch, M., Pauly, M., Kobbelt, L., Alliez, P., Lévy, B., Bischoff, S., Rössl, C.: Geometric modeling based on polygonal meshes. In: SIGGRAPH Course Notes. (2007)
14. Halstead, M.A., Barsky, B.A., Klein, S.A., Mandell, R.B.: Reconstructing curved surfaces from specular reflection patterns using spline surface fitting of normals. In: SIGGRAPH. (1996) 335–342

A Appendix

We summarize on systems of nonlinear equations for solving the general alignment problem of reflection lines.

- (*REFL*, $\mathbf{e} + 2x$) — Given $(\mathbf{e}, \mathbf{p}_1)$, determine a member $(\mathbf{e}, \mathbf{p}_2)$ of the same family by two selection normals $\mathbf{n}_1, \mathbf{n}_2$ (all members $(\mathbf{e}, \mathbf{p}_i)$ need to fulfill $(\mathbf{p}_1 \times \mathbf{p}_2) \cdot \mathbf{p}_i = 0$ to vary along a common line at infinity):

$$\begin{aligned} \text{Solve } & (2(\mathbf{e} \cdot \mathbf{n}_1) \mathbf{n}_1 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_2) \mathbf{n}_2 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0 \\ & \text{and } \mathbf{p}_2 \cdot \mathbf{p}_2 \stackrel{\perp}{=} 1 \quad \text{for } \mathbf{p}_2. \end{aligned}$$

- (*REFL*, $4x + 2x$) — Determine $(\mathbf{e}, \mathbf{p}_1)$ (four interpolating selections $\mathbf{x}_1, \dots, \mathbf{x}_4$) and $(\mathbf{e}, \mathbf{p}_2)$ (two interpolating selections $\mathbf{x}_5, \mathbf{x}_6$):

$$\begin{aligned} \text{Solve } & (2(\mathbf{e} \cdot \mathbf{n}_1) \mathbf{n}_1 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_2) \mathbf{n}_2 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \\ & (2(\mathbf{e} \cdot \mathbf{n}_3) \mathbf{n}_3 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_4) \mathbf{n}_4 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \\ & (2(\mathbf{e} \cdot \mathbf{n}_5) \mathbf{n}_5 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_6) \mathbf{n}_6 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \\ & \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1, \quad \mathbf{p}_1 \cdot \mathbf{p}_1 \stackrel{\perp}{=} 1, \quad \text{and } \mathbf{p}_2 \cdot \mathbf{p}_2 \stackrel{\perp}{=} 1 \\ & \text{for } \mathbf{e}, \mathbf{p}_1 \text{ and } \mathbf{p}_2. \end{aligned}$$

- (*REFL*, $3x + 3x$) — Determine $(\mathbf{e}, \mathbf{p}_1)$ (three interpolating selections $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$) and $(\mathbf{e}, \mathbf{p}_2)$ (three interpolating selections $\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6$):

$$\begin{aligned} \text{Solve } & (2(\mathbf{e} \cdot \mathbf{n}_1) \mathbf{n}_1 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_2) \mathbf{n}_2 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \\ & (2(\mathbf{e} \cdot \mathbf{n}_3) \mathbf{n}_3 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_4) \mathbf{n}_4 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \\ & (2(\mathbf{e} \cdot \mathbf{n}_5) \mathbf{n}_5 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_6) \mathbf{n}_6 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \\ & \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1, \quad \mathbf{p}_1 \cdot \mathbf{p}_1 \stackrel{\perp}{=} 1, \quad \text{and } \mathbf{p}_2 \cdot \mathbf{p}_2 \stackrel{\perp}{=} 1 \\ & \text{for } \mathbf{e}, \mathbf{p}_1 \text{ and } \mathbf{p}_2. \end{aligned}$$

- (*REFL*, $2x + 2x + 2x + \text{Mid}$) — Determine $(\mathbf{e}, \mathbf{p}_1)$ (two interpolating selections $\mathbf{x}_1, \mathbf{x}_2$), $(\mathbf{e}, \mathbf{p}_2)$ (three interpolating selections $\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_7$) and $(\mathbf{e}, \mathbf{p}_3)$ (two interpolating selections $\mathbf{x}_5, \mathbf{x}_6$) of the same family ($\mathbf{p}_1, \mathbf{p}_2$ and \mathbf{p}_3 need to be coplanar to vary along a common line at infinity, thus $|\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3| \stackrel{\perp}{=} 0$):

$$\begin{aligned} \text{Solve } & (2(\mathbf{e} \cdot \mathbf{n}_1) \mathbf{n}_1 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_2) \mathbf{n}_2 - \mathbf{e}) \cdot \mathbf{p}_1 \stackrel{\perp}{=} 0, \\ & (2(\mathbf{e} \cdot \mathbf{n}_3) \mathbf{n}_3 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_4) \mathbf{n}_4 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \\ & (2(\mathbf{e} \cdot \mathbf{n}_5) \mathbf{n}_5 - \mathbf{e}) \cdot \mathbf{p}_3 \stackrel{\perp}{=} 0, \quad (2(\mathbf{e} \cdot \mathbf{n}_6) \mathbf{n}_6 - \mathbf{e}) \cdot \mathbf{p}_3 \stackrel{\perp}{=} 0, \\ & (2(\mathbf{e} \cdot \mathbf{n}_7) \mathbf{n}_7 - \mathbf{e}) \cdot \mathbf{p}_2 \stackrel{\perp}{=} 0, \quad |\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3| \stackrel{\perp}{=} 0, \\ & \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1, \quad \mathbf{p}_1 \cdot \mathbf{p}_1 \stackrel{\perp}{=} 1, \quad \mathbf{p}_2 \cdot \mathbf{p}_2 \stackrel{\perp}{=} 1 \quad \text{and } \mathbf{p}_3 \cdot \mathbf{p}_3 \stackrel{\perp}{=} 1 \\ & \text{for } \mathbf{e}, \mathbf{p}_1, \mathbf{p}_2 \text{ and } \mathbf{p}_3. \end{aligned}$$

Let $\cos \alpha_j = (2(\mathbf{e} \cdot \mathbf{n}_i) \mathbf{n}_i - \mathbf{e}) \cdot \mathbf{r}$, then we obtain for the general alignment problem of reflection circles:

- (*RECI*, $4x + 2x$) — Determine $(\mathbf{e}, \mathbf{r}, \cos \alpha_1)$ (four interpolating selections \mathbf{x}_i , $i \in \{1, \dots, 4\}$) and $(\mathbf{e}, \mathbf{r}, \cos \alpha_2)$ (two interpolating selections \mathbf{x}_i , $i \in \{5, \dots, 6\}$):

$$\begin{aligned} \text{Solve } & (\mathbf{e} \cdot \mathbf{n}_1)(\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_2)(\mathbf{r} \cdot \mathbf{n}_2), \quad (\mathbf{e} \cdot \mathbf{n}_1)(\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_3)(\mathbf{r} \cdot \mathbf{n}_3), \\ & (\mathbf{e} \cdot \mathbf{n}_1)(\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_4)(\mathbf{r} \cdot \mathbf{n}_4), \quad (\mathbf{e} \cdot \mathbf{n}_5)(\mathbf{r} \cdot \mathbf{n}_5) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_6)(\mathbf{r} \cdot \mathbf{n}_6), \\ & \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1 \quad \text{and} \quad \mathbf{r} \cdot \mathbf{r} \stackrel{\perp}{=} 1 \quad \text{for } \mathbf{e} \text{ and } \mathbf{r}. \end{aligned}$$

- (*RECI*, $3x + 3x$) — Determine $(\mathbf{e}, \mathbf{r}, \cos \alpha_1)$ (three interpolating selections \mathbf{x}_i , $i \in \{1, \dots, 3\}$) and $(\mathbf{e}, \mathbf{r}, \cos \alpha_2)$ (three interpolating selections \mathbf{x}_i , $i \in \{4, \dots, 6\}$):

$$\begin{aligned} \text{Solve } & (\mathbf{e} \cdot \mathbf{n}_1)(\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_2)(\mathbf{r} \cdot \mathbf{n}_2), \quad (\mathbf{e} \cdot \mathbf{n}_1)(\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_3)(\mathbf{r} \cdot \mathbf{n}_3), \\ & (\mathbf{e} \cdot \mathbf{n}_4)(\mathbf{r} \cdot \mathbf{n}_4) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_5)(\mathbf{r} \cdot \mathbf{n}_5), \quad (\mathbf{e} \cdot \mathbf{n}_4)(\mathbf{r} \cdot \mathbf{n}_4) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_6)(\mathbf{r} \cdot \mathbf{n}_6), \\ & \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1 \quad \text{and} \quad \mathbf{r} \cdot \mathbf{r} \stackrel{\perp}{=} 1 \quad \text{for } \mathbf{e} \text{ and } \mathbf{r}. \end{aligned}$$

- (*RECI*, $2x + 2x + 2x + 2x$) — Determine $(\mathbf{e}, \mathbf{r}, \cos \alpha_j)$, $j \in \{1, \dots, 4\}$ using four paired selections:

$$\begin{aligned} \text{Solve } & (\mathbf{e} \cdot \mathbf{n}_1)(\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_2)(\mathbf{r} \cdot \mathbf{n}_2), \quad (\mathbf{e} \cdot \mathbf{n}_3)(\mathbf{r} \cdot \mathbf{n}_3) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_4)(\mathbf{r} \cdot \mathbf{n}_4), \\ & (\mathbf{e} \cdot \mathbf{n}_5)(\mathbf{r} \cdot \mathbf{n}_5) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_6)(\mathbf{r} \cdot \mathbf{n}_6), \quad (\mathbf{e} \cdot \mathbf{n}_7)(\mathbf{r} \cdot \mathbf{n}_7) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_8)(\mathbf{r} \cdot \mathbf{n}_8), \\ & \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1 \quad \text{and} \quad \mathbf{r} \cdot \mathbf{r} \stackrel{\perp}{=} 1 \quad \text{for } \mathbf{e} \text{ and } \mathbf{r}. \end{aligned}$$

- (*RECI*, $2x + 2x + 2x + \text{constr}$) — Determine $(\mathbf{e}, \mathbf{r}, \cos \alpha_j)$, $j \in \{1, \dots, 3\}$ using three paired selections by imposing an order on the scalar values:

$$\begin{aligned} \text{Solve } & (\mathbf{e} \cdot \mathbf{n}_1)(\mathbf{r} \cdot \mathbf{n}_1) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_2)(\mathbf{r} \cdot \mathbf{n}_2), \quad (\mathbf{e} \cdot \mathbf{n}_3)(\mathbf{r} \cdot \mathbf{n}_3) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_4)(\mathbf{r} \cdot \mathbf{n}_4), \\ & (\mathbf{e} \cdot \mathbf{n}_5)(\mathbf{r} \cdot \mathbf{n}_5) \stackrel{\perp}{=} (\mathbf{e} \cdot \mathbf{n}_6)(\mathbf{r} \cdot \mathbf{n}_6), \quad \mathbf{e} \cdot \mathbf{e} \stackrel{\perp}{=} 1, \quad \mathbf{r} \cdot \mathbf{r} \stackrel{\perp}{=} 1 \\ & \text{and } (\mathbf{e} \cdot \mathbf{n}_3)(\mathbf{r} \cdot \mathbf{n}_3) \stackrel{\perp}{=} \frac{1}{2}((\mathbf{e} \cdot \mathbf{n}_1)(\mathbf{r} \cdot \mathbf{n}_1) + (\mathbf{e} \cdot \mathbf{n}_5)(\mathbf{r} \cdot \mathbf{n}_5)) \\ & \text{for } \mathbf{e} \text{ and } \mathbf{r}. \end{aligned}$$

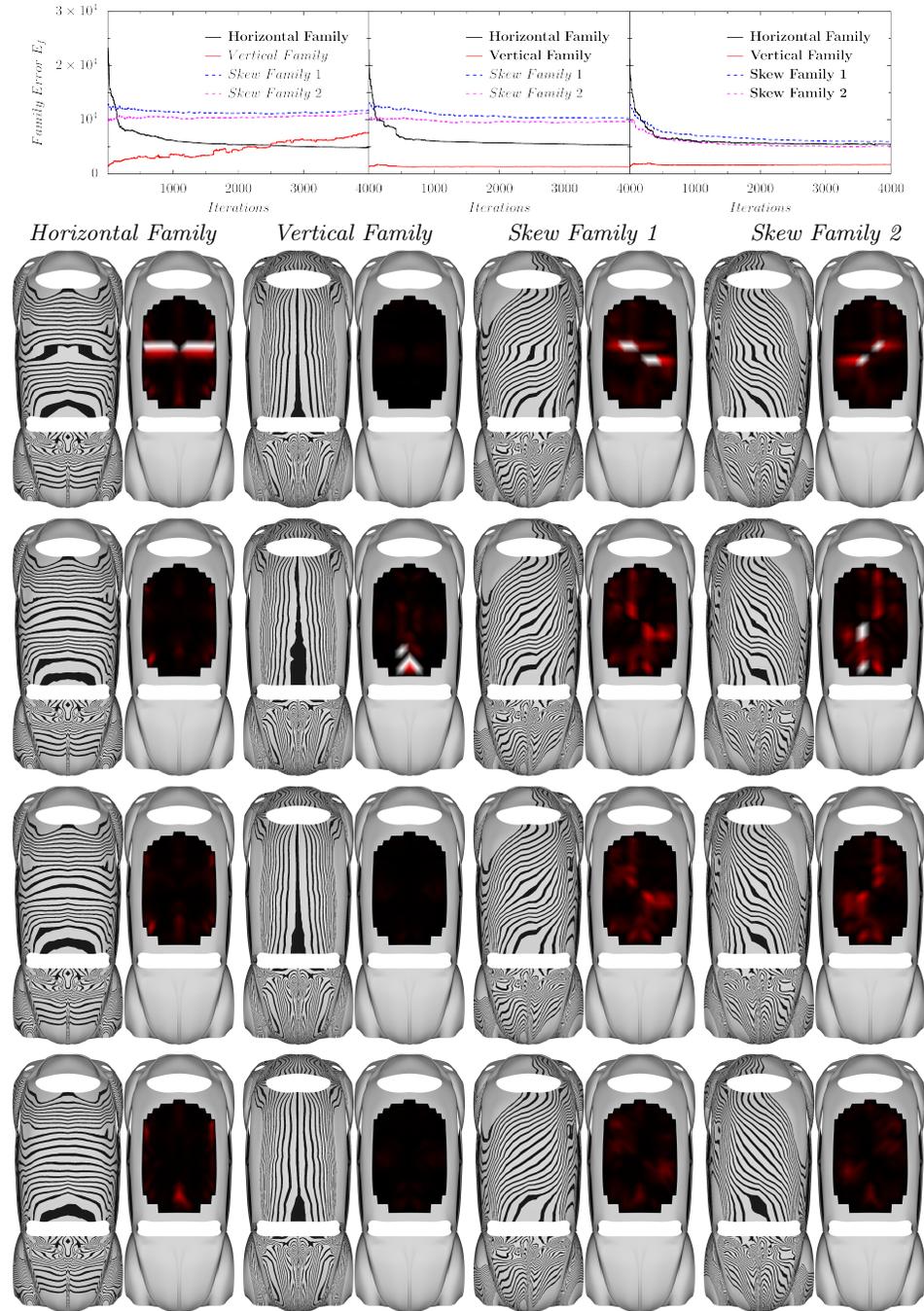


Fig. 8: Optimization of the Volkswagen Beetle car roof. Bold family names in the error plot correspond to optimized families. The top picture row shows initial curve families with a plot of their respective curve curvatures. Solely fairing of the horizontally oriented family corrupts the vertically aligned family (left plot, second row). Simultaneous fairing of both families enhances appearance the horizontal family while preserving the quality of the vertical family (center plot, third row). Even better results can be achieved by considering also the two skew families (right plot, last row)