



HAL
open science

Extrapolating curvature lines in rough concept sketches using mixed-integer nonlinear optimization

Claudia d'Ambrosio, Jon R. Lee, Leo Liberti, Maks Ovsjanikov

► **To cite this version:**

Claudia d'Ambrosio, Jon R. Lee, Leo Liberti, Maks Ovsjanikov. Extrapolating curvature lines in rough concept sketches using mixed-integer nonlinear optimization. *Optimization and Engineering*, 2019, 20 (2), pp.337-347. 10.1007/s11081-018-9405-6 . hal-02350272

HAL Id: hal-02350272

<https://hal.science/hal-02350272>

Submitted on 6 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Extrapolating curvature lines in rough concept sketches using mixed-integer nonlinear optimization

Claudia D'Ambrosio, Jon Lee,
Leo Liberti, Maks Ovsjanikov

April 5, 2018

Abstract We present several mathematical-optimization formulations for a problem that commonly occurs in geometry processing and specifically in the design of so-called smooth direction fields on surfaces. This problem has direct applications in 3D shape parameterization, texture mapping, and shape design via rough concept sketches, among many others. A key challenge in this setting is to design a set of unit-norm directions, on a given surface, that satisfy some prescribed constraints and vary smoothly. This naturally leads to mixed-integer optimization formulations, because the smoothness needs to be formulated with respect to angle-valued variables, which to compare one needs to fix the discrete jump between nearby points. Previous works have primarily attacked this problem via a greedy ad-hoc strategy with a specialized solver. We demonstrate how the problem can be cast in a standard mathematical-optimization form, and we suggest several relaxations that are especially adapted to modern mathematical-optimization solvers.

Keywords mixed-integer nonlinear optimization · mixed-integer linear optimization · computer graphics · sketching

Mathematics Subject Classification (2010) 90C11 · 90C26

C. D'Ambrosio
CNRS LIX, École Polytechnique, 91128 Palaiseau, France
E-mail: dambrosio@lix.polytechnique.fr

J. Lee
Dept. of IOE, University of Michigan, Ann Arbor, Michigan, USA
E-mail: jonxlee@umich.edu

L. Liberti
CNRS LIX, École Polytechnique, 91128 Palaiseau, France
E-mail: liberti@lix.polytechnique.fr

M. Ovsjanikov
CNRS LIX, École Polytechnique, 91128 Palaiseau, France
E-mail: maks@lix.polytechnique.fr

Introduction

Design and analysis of generalized vector fields on surfaces is an active area of research in computer graphics and geometry processing (see [21] for a recent overview). A common problem in this domain is to design tangent vector fields that vary smoothly on the shape surface and are aligned with some prescribed directions, such as the directions of principal curvature (see [6, 5, 12]). One key application of vector-field design is in the context of shape parameterization, where the goal is to compute mappings between 3D surfaces and the 2D plane, and which is used for mesh generation, texture synthesis, and other applications (see, for example, [17] and [1] for an overview). In this context, one is often required to design *generalized direction fields*, rather than the classical tangent vector fields. The main distinction between direction and vector fields is that the former might have multiple directions at every point, and moreover the labeling of each direction at a point is irrelevant, so that at a given point the different directions can be relabeled without changing their geometric meaning (see Section 2 of [21] for an overview of common direction fields). The most basic and commonly used type of direction fields are called *unit cross-fields* (or *4-RoSy fields*) which consist of four unit-length, orthogonal directions at every point of the surface, and which are used especially in shape remeshing (see [2]).

Designing smooth direction fields has been addressed with two classes of approaches: either by considering each direction as a complex number and using this representation for enforcing smoothness and alignment (see e.g. [17] and Section 5.2 of [21] for an overview) or by representing each direction via an angle with respect to some fixed reference direction at each point. The advantage of the former representation is that under certain conditions it allows us to compute a *globally-optimal direction field* (see [12]), at the expense of removing the unit-norm constraint at every point. On the other hand, the angle-based representations (see [2]) enforce the unit-norm condition as a hard constraint, but lead to potentially difficult optimization problems. The key difficulty arises in enforcing the smoothness of the sought direction field, which requires comparing the angles across different points, and thus requires solving for additional integer variables that resolve the cyclical angle ambiguity (i.e., multiples of 2π). Nevertheless, it has been shown in recent works (see [21]) that removing the unit-norm constraints can lead to undesired singularities in the computed direction fields, and therefore the angle-based representation and resulting optimization schemes are commonly used despite the computational burden.

Previous works have primarily attacked the resulting optimization problem via an ad-hoc greedy strategy implemented as a specialized solver (see [2, 3]). Our goal in what follows is to demonstrate how the problem can be cast in a standard mathematical-optimization form, and we suggest several relaxations that are especially tailored to modern mathematical-optimization solvers.

As the basis for our formulation, we use the problem suggested in [10], where the direction-field design is used for inference of 3D shapes based on

rough 2D concept sketches. Despite the specialized application domain, the key optimization problem (see Section 6 and especially Eq. (5) in [10]) shares the properties of the difficulties in designing smooth direction fields in general. Therefore, we first present the optimization problem as it is stated in [10], and then we describe several reformulations, with emphasis on linearization and suitability for modern mathematical-optimization solvers.

We also note that the problem formulation in [10] is particularly simple because the direction field is optimized on the 2D (image) domain, which reduces some of the difficulties associated with dealing with curved surfaces. As a result, below we refer to direction fields defined on “pixels” rather than more generic points and assume that the underlying domain is a fixed 2D grid.

1 The optimization problem

Our goal in this section is to precisely set out a mathematical-optimization formulation for the problem. The formulation is the one implied in [10], and so we follow their terminology; also see [9].

We let I denote the set of pixels. We have continuous *cross variables* α_i, β_i for each pixel $i \in I$, which we constrain as follows:

$$0 \leq \alpha_i \leq \frac{\pi}{2}, \text{ for } i \in I; \quad (1)$$

$$-\frac{\pi}{4} \leq \beta_i \leq \frac{\pi}{4}, \text{ for } i \in I. \quad (2)$$

Next for each pixel $i \in I$, we let \mathcal{N}_i comprise the upper and right neighbors of pixel i (when those neighbors exist). Then, for each pixel $i \in I$ and $j \in \mathcal{N}_i$, we define variables $p_{ij} \in \mathbb{Z}$.

Over these variable domains, we consider minimizing the energy function

$$E_{\text{angle}} := E_{\text{smooth}} + E_{\text{strokes}} + E_{\beta}.$$

We have

$$E_{\text{smooth}} := \sum_{i \in I} \sum_{j \in \mathcal{N}_i} E_{\text{smooth}}^{ij}, \quad (3)$$

where

$$E_{\text{smooth}}^{ij} := 2 \left[\left(\alpha_i - \alpha_j + \frac{\pi}{2} p_{ij} \right)^2 + \left((-1)^{p_{ij}} \beta_i - \beta_j \right)^2 \right]. \quad (4)$$

Note the integer variables p_{ij} as the exponent of -1 in (3); we will return to this issue in the next section.

For a subset $S_c \subset I$, we have *target angles* θ_i , for $i \in S_c$, and so we define

$$E_{\text{strokes}} := \sum_{i \in S_c} w_i (\alpha_i + \beta_i - \theta_i)^2, \quad (5)$$

where the weights w_i are chosen in a complicated manner (see [10]).

Next,

$$E_\beta := w_\beta \sum_{i \in I} \beta_i^2, \quad (6)$$

where w_β is a weight coefficient.

Finally, typical values for the remaining constants are $w_{\text{strokes}} = 1$ (which could rather be absorbed into the w_i) and $w_\beta = 10^{-6}$.

2 Reformulation

It was observed in [10] that the manner in which the integer p_{ij} variables arise in E_{smooth}^{ij} is vexing:

“The additional difficulty in our case is that the integers p_{ij} contribute nonlinearly due to expressions $(-1)^{p_{ij}}$, which additionally make continuous relaxation impossible without switching to complex numbers.”

Of course complex numbers are quite troublesome in optimization (see [16], for example), and so we propose a reformulation, essentially by replacing each variable $p_{ij} \in \mathbb{Z}$ by a pair of variables

$$p_{ij,1} \in \{0, 1\} \quad \text{and} \quad p_{ij,2} \in \mathbb{Z}, \quad \text{for } i \in I, j \in \mathcal{N}_i. \quad (7)$$

We do this by letting $p_{ij,1}$ be an indicator variable for the event that p_{ij} is odd, and we let $2p_{ij,2}$ be the “even part” of p_{ij} . Note that variations of this technique have been used before, but in completely different contexts (see e.g. [14, 13]). There is obviously a trade-off between increasing the number of variables and reducing the number of nonlinear terms. Because of the current technological gap in solvers for linear versus nonlinear optimization problems, we prefer doubling the number of integer variables in order to achieve a reduction in nonlinearity.

Specifically, we write

$$p_{ij} = p_{ij,1} + 2p_{ij,2},$$

Note that this correspondence between $p_{ij} \in \mathbb{Z}$ and pairs $p_{ij,1} \in \{0, 1\}$, $p_{ij,2} \in \mathbb{Z}$ is bijective between \mathbb{Z} and $\{0, 1\} \times \mathbb{Z}$.

Considering now (4), we replace

$$\alpha_i - \alpha_j + \frac{\pi}{2} p_{ij}$$

with

$$\alpha_i - \alpha_j + \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2})$$

and we replace

$$(-1)^{p_{ij}}$$

with

$$1 - 2p_{ij,1}.$$

The first substitution preserves the linearity that is there, but the second one linearizes a very troublesome term. Because $p_{ij,1}$ is the indicator function for p_{ij} being odd, we can easily see that the substitution is an exact reformulation. So, we have recast (4) as

$$E_{\text{smooth}}^{ij} := 2 \left[\left(\alpha_i - \alpha_j + \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}) \right)^2 + ((1 - 2p_{ij,1})\beta_i - \beta_j)^2 \right]. \quad (8)$$

Next we take further advantage of the binary variables, reducing nonlinearity. The expression

$$((1 - 2p_{ij,1})\beta_i - \beta_j)^2$$

expands to

$$\beta_i^2 + \beta_j^2 - 2\beta_i\beta_j - \underline{4\beta_i^2 p_{ij,1}} + 4\beta_i\beta_j p_{ij,1} + \underline{4\beta_i^2 p_{ij,1}^2}.$$

But because $p_{ij,1}$ is a $\{0,1\}$ variable, we may replace $p_{ij,1}^2$ with $p_{ij,1}$, and the two above-underlined terms cancel, reducing the degree-4 polynomial to a degree-3 polynomial with only a single cubic monomial.

Similarly, the expression

$$\left(\alpha_i - \alpha_j + \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}) \right)^2$$

expands to

$$\begin{aligned} & \alpha_i^2 + \alpha_j^2 - 2\alpha_i\alpha_j + \pi\alpha_i p_{ij,1} - \pi\alpha_j p_{ij,1} + 2\pi\alpha_i p_{ij,2} - 2\pi\alpha_j p_{ij,2} \\ & + \left(\frac{\pi}{2} \right)^2 \underline{p_{ij,1}^2} + \pi^2 p_{ij,1} p_{ij,2} + \pi^2 p_{ij,2}^2. \end{aligned}$$

Here again we may replace $p_{ij,1}^2$ with $p_{ij,1}$, replacing a quadratic monomial with a linear term.

So, we have our final form:

$$\begin{aligned} E_{\text{smooth}}^{ij} := & 2 \left[\alpha_i^2 + \alpha_j^2 - 2\alpha_i\alpha_j + \pi\alpha_i p_{ij,1} - \pi\alpha_j p_{ij,1} + 2\pi\alpha_i p_{ij,2} \right. \\ & - 2\pi\alpha_j p_{ij,2} + \left(\frac{\pi}{2} \right)^2 p_{ij,1} + \pi^2 p_{ij,1} p_{ij,2} + \pi^2 p_{ij,2}^2 \\ & \left. + \beta_i^2 + \beta_j^2 - 2\beta_i\beta_j + 4\beta_i\beta_j p_{ij,1} \right], \end{aligned} \quad (9)$$

which we use in (3).

To be completely clear, we state the full model. We minimize

$$E_{\text{angle}} := E_{\text{smooth}} + w_{\text{strokes}} E_{\text{strokes}} + w_{\beta} E_{\beta},$$

where $w_{\text{strokes}} E_{\text{strokes}} + w_{\beta} E_{\beta}$ has been defined, and we use (9) for the summands of E_{smooth} . Our variables are (1),(2),(7).

To compute, because we have products of two and three variables, we need a nonconvex MINLO (mixed-integer nonlinear-optimization) solver like **Baron** (see [20,18]). Also see [19] for recent guidance on handling triple products.

3 A related formulation with more linearity

Another possibility is to use a 1-norm definition of E^{ij} . This could well make the optimization problem more tractable/scalable. But this *might* lead to some anomalies in the solution, due to its well-known sparsity-inducing effect.

For this, we return to (8), and we instead consider

$$E_{\text{smooth}}^{ij} := 2 \left[\left| \alpha_i - \alpha_j + \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}) \right| + |(1 - 2p_{ij,1})\beta_i - \beta_j| \right]. \quad (10)$$

This buys us a lot of linearity, at the cost of some additional variables and constraints. As is standard, we let variable

$$s_{ij} \in \mathbb{R}, \quad i \in I, \quad j \in \mathcal{N}_i, \quad (11)$$

stand in for

$$\left| \alpha_i - \alpha_j + \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}) \right|,$$

and variable

$$t_{ij} \in \mathbb{R}, \quad i \in I, \quad j \in \mathcal{N}_i, \quad (12)$$

stand in for

$$|(1 - 2p_{ij,1})\beta_i - \beta_j|.$$

Then we have

$$E_{\text{smooth}}^{ij} := 2 [s_{ij} + t_{ij}], \quad (13)$$

with now the additional side constraints

$$s_{ij} \geq \alpha_i - \alpha_j + \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}); \quad (14)$$

$$s_{ij} \geq -\alpha_i + \alpha_j - \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}); \quad (15)$$

$$t_{ij} \geq (1 - 2p_{ij,1})\beta_i - \beta_j; \quad (16)$$

$$t_{ij} \geq (-1 + 2p_{ij,1})\beta_i + \beta_j. \quad (17)$$

This has drastically less nonlinearity than (9), with only the bilinear products $p_{ij,1}\beta_i$ of (16) and (17). Of course there is also the purely continuous convex nonlinearity of (5) and (6), but this is unlikely to be very troublesome for solvers.

To be completely clear, we state the full model. We minimize

$$E_{\text{angle}} := E_{\text{smooth}} + w_{\text{strokes}} E_{\text{strokes}} + w_{\beta} E_{\beta},$$

where $w_{\text{strokes}} E_{\text{strokes}} + w_{\beta} E_{\beta}$ has been defined, and we use (13) for the summands of E_{smooth} . Our variables are (1),(2),(7),(11),(12), and we have the side constraints (14),(15),(16),(17).

Finally, in a similar manner, we could also make 1-norm versions of E_{strokes} and/or E_{β} . That is, we let

$$\gamma_i \in \mathbb{R}, \quad i \in I, \quad (18)$$

stand in for $|\alpha_i + \beta_i - \theta_i|$ and

$$\psi_i \in \mathbb{R}, \quad i \in I, \quad (19)$$

stand in for $|\beta_i|$, leading to the use of the 1-norm versions

$$E_{\text{strokes}} := \sum_{i \in \mathcal{S}_c} w_i \gamma_i \quad (20)$$

and

$$E_\beta := \sum_{i \in I} \psi_i, \quad (21)$$

with the additional side constraints

$$\gamma_i \geq \alpha_i + \beta_i - \theta_i; \quad (22)$$

$$\gamma_i \geq -\alpha_i - \beta_i + \theta_i; \quad (23)$$

$$\psi_i \geq \beta_i; \quad (24)$$

$$\psi_i \geq -\beta_i. \quad (25)$$

4 And even more linearity

Internally, nonconvex MINLO solvers (like **Baron**) attack the terms $p_{ij,1}\beta_i$ of (16) and (17) via the well-known McCormick relaxation (see [15]). This would replace occurrences of the product $p_{ij,1}\beta_i$ by new variables z_{ij} . Then, they would (initially) convexify by employing the McCormick (linear) inequalities, which manifest here as

$$z_{ij} \geq -\frac{\pi}{4}; \quad (26)$$

$$z_{ij} \leq \beta_i - \frac{\pi}{4}p_{ij,1} + \frac{\pi}{4}; \quad (27)$$

$$z_{ij} \leq \frac{\pi}{4}p_{ij,1}; \quad (28)$$

$$z_{ij} \geq \beta_i + \frac{\pi}{4}p_{ij,1} - \frac{\pi}{4}. \quad (29)$$

These inequalities capture the tetrahedron that is the convex hull of the four extremal points that satisfy the product equation. That is $(\beta_i, p_{ij,1}, z_{ij}) = (-\pi/4, 0, 0), (\pi/4, 0, 0), (-\pi/4, 1, -\pi/4),$ and $(\pi/4, 1, \pi/4)$.

We have a lot of moving parts now, and it is worth highlighting some good options. We would apply this McCormick relaxation to the 1-norm version of E_{smooth} , enforcing still $p_{ij,1} \in \{0, 1\}$ and $p_{ij,2} \in \mathbb{Z}$. In doing so, we would get a *relaxation* that is a *convex* MINLO problem, which should scale better than a nonconvex one; here, software like **Bonmin** (see [4]) could well be more appropriate than **Baron**. Further, if we also employed 1-norm versions of E_{strokes} and E_β , then we would get a *relaxation* that is a MILO (mixed-integer linear-optimization) problem, which should scale even better; here, software like **Cplex** (see [11]) or **Gurobi** (see [8]) may be more appropriate. In both of these scenarios, we are only solving a relaxation, so it remains to be seen if that produces good solutions or whether a more demanding approach is needed.

5 Toward computing

Practical application of our ideas must be sensitive to scaling. Therefore, we recommend initially staying in the realm of MILO and NLO (nonlinear optimization) as this has the best immediate chance of scaling to the size needed for practical use.

Our MILO model is then as follows.

$$\begin{aligned}
& \text{minimize} \quad \sum_{i \in I} \sum_{j \in \mathcal{N}_i} 2(s_{ij} + t_{ij}) + w_{\text{strokes}} \sum_{i \in S_c} w_i \gamma_i + w_\beta \sum_{i \in I} \psi_i, \\
& \text{subject to:} \\
& s_{ij} \geq \alpha_i - \alpha_j + \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}), \text{ for } i \in I, j \in \mathcal{N}_i; \\
& s_{ij} \geq -\alpha_i + \alpha_j - \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}), \text{ for } i \in I, j \in \mathcal{N}_i; \\
& t_{ij} \geq \beta_i - 2z_{ij} - \beta_j, \text{ for } i \in I, j \in \mathcal{N}_i; \\
& t_{ij} \geq -\beta_i + 2z_{ij} + \beta_j, \text{ for } i \in I, j \in \mathcal{N}_i; \\
& \gamma_i \geq \alpha_i + \beta_i - \theta_i, \text{ for } i \in I; \\
& \gamma_i \geq -\alpha_i - \beta_i + \theta_i, \text{ for } i \in I; \\
& \psi_i \geq \beta_i, \text{ for } i \in I; \\
& \psi_i \geq -\beta_i, \text{ for } i \in I; \\
& z_{ij} \geq -\frac{\pi}{4}, \text{ for } i \in I, j \in \mathcal{N}_i; \\
& z_{ij} \leq \beta_i - \frac{\pi}{4} p_{ij,1} + \frac{\pi}{4}, \text{ for } i \in I, j \in \mathcal{N}_i; \\
& z_{ij} \leq \frac{\pi}{4} p_{ij,1}, \text{ for } i \in I, j \in \mathcal{N}_i; \\
& z_{ij} \geq \beta_i + \frac{\pi}{4} p_{ij,1} - \frac{\pi}{4}, \text{ for } i \in I, j \in \mathcal{N}_i; \\
& 0 \leq \alpha_i \leq \frac{\pi}{2}, \text{ for } i \in I; \\
& -\frac{\pi}{4} \leq \beta_i \leq \frac{\pi}{4}, \text{ for } i \in I; \\
& p_{ij,1} \in \{0, 1\} \text{ and } p_{ij,2} \in \mathbb{Z}, \text{ for } i \in I, j \in \mathcal{N}_i.
\end{aligned}$$

We wish to emphasize that we have relaxed the nonlinear constraints $z_{ij} = p_{ij,1}\beta_i$. Therefore, a prudent approach might be to solve this MILO relaxation (using `Cplex` or `Gurobi`) for the purpose of getting good values for the discrete variables $p_{ij,1}$, $p_{ij,2}$. Then, with those variables *fixed*, solve the following box-constrained convex-quadratic optimization problem:

$$\begin{aligned}
& \text{minimize} \quad \sum_i \sum_{j \in \mathcal{N}_i} 2 \left[\left(\alpha_i - \alpha_j + \frac{\pi}{2} (p_{ij,1} + 2p_{ij,2}) \right)^2 + \left((1 - 2p_{ij,1})\beta_i - \beta_j \right)^2 \right] \\
& + w_{\text{strokes}} \sum_{i \in S_c} w_i (\alpha_i + \beta_i - \theta_i)^2 + w_\beta \sum_{i \in I} \beta_i^2,
\end{aligned}$$

subject to:

$$0 \leq \alpha_i \leq \frac{\pi}{2}, \text{ for } i \in I ;$$

$$-\frac{\pi}{4} \leq \beta_i \leq \frac{\pi}{4}, \text{ for } i \in I .$$

There are many very-efficient approaches for this; see [7], for example. Alternatively, we could utilize a general-purpose NLO solver like Ipopt (see [22]).

6 Conclusions and outlook

We have presented a variety of principled optimization models for the problem of extrapolating curvature lines in rough concept sketches. It remains to be seen if any of these can find practical use. Our goal in this paper has been to expose our ideas with the hope that some of the ideas may find their way into computer-graphics software tools.

We believe that similar ideas to the ones we presented can be developed for other problems in computer graphics. We are pursuing this direction ourselves, and we hope that our work will inspire others as well.

Acknowledgments

This paper was supported in part by Horizon 2020, a European Union Research and Innovation programme, under the Marie Skłodowska-Curie grant agreement No. 764759. J. Lee was supported in part by ONR grant N00014-17-1-2296 and LIX, École Polytechnique.

References

1. D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin, *Quad-mesh generation and processing: A survey*, Computer Graphics Forum, vol. 32, Wiley Online Library, 2013, pp. 51–76.
2. D. Bommes, H. Zimmer, and L. Kobbelt, *Mixed-integer quadrangulation*, ACM Transactions On Graphics (TOG), vol. 28, ACM, 2009, p. 77.
3. ———, *Practical mixed-integer optimization for geometry processing*, International Conference on Curves and Surfaces, Springer, 2010, pp. 193–206.
4. P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, *An algorithmic framework for convex mixed integer nonlinear programs*, Discrete Optimization 5 (2008), no. 2, 186–204.
5. K. Crane, M. Desbrun, and P. Schröder, *Trivial connections on discrete surfaces*, Computer Graphics Forum, vol. 29, Wiley Online Library, 2010, pp. 1525–1533.
6. M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe, *Design of tangent vector fields*, ACM transactions on graphics (TOG), vol. 26, ACM, 2007, p. 56.
7. A. Friedlander, J.M. Martínez, and M. Raydon, *A new method for large-scale box constrained convex quadratic minimization problems*, Optimization Methods and Software 5 (1995), no. 1, 57–74.
8. Gurobi Optimization, Inc., *Gurobi optimizer reference manual*, 2016.
9. E. Iarussi, *Computer drawing tools for assisting learners, hobbyists, and professionals*, Ph.D. thesis, University Nice Sophia Antipolis, 2015.

10. E. Iarussi, D. Bommes, and A. Bousseau, *Bendfields: Regularized curvature fields from rough concept sketches*, ACM Transactions on Graphics **34** (2015), no. 3, 24:1–24:16.
11. IBM, *ILOG CPLEX 12.6 user's manual*, IBM, 2014.
12. F. Knöppel, K. Crane, U. Pinkall, and P. Schröder, *Globally optimal direction fields*, ACM Transactions on Graphics (TOG) **32** (2013), no. 4, 59.
13. J. Lee, J. Leung, and S. de Vries, *Separating type-I odd-cycle inequalities for a binary-encoded edge-coloring formulation*, Journal of Combinatorial Optimization **9** (2005), no. 1, 59–67.
14. J. Lee and F. Margot, *On a binary-encoded ILP coloring formulation*, INFORMS Journal on Computing **19** (2007), no. 3, 406–415.
15. G.P. McCormick, *Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems*, Mathematical Programming **10** (1976), no. 1, 147–175.
16. D.K. Molzahn and I.A. Hiskens, *Convex relaxations of optimal power flow problems: An illustrative example*, IEEE Transactions on Circuits and Systems I: Regular Papers **63** (2016), no. 5, 650–660.
17. N. Ray, W.C. Li, B. Lévy, A. Sheffer, and P. Alliez, *Periodic global parameterization*, ACM Transactions on Graphics (TOG) **25** (2006), no. 4, 1460–1485.
18. N.V. Sahinidis, *BARON 14.3.1: Global Optimization of Mixed-Integer Nonlinear Programs*, User's Manual, 2014.
19. E. Speakman and J. Lee, *Quantifying double McCormick*, Mathematics of Operations Research **42** (2017), no. 4, 1230–1253.
20. M. Tawarmalani and N.V. Sahinidis, *A polyhedral branch-and-cut approach to global optimization*, Mathematical Programming **103** (2005), 225–249.
21. A. Vaxman, M. Campen, O. Diamanti, D. Panozzo, D. Bommes, K. Hildebrandt, and M. Ben-Chen, *Directional field synthesis, design, and processing*, Computer Graphics Forum, vol. 35, Wiley Online Library, 2016, pp. 545–572.
22. A. Wächter and L.T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming **106** (2006), no. 1, 25–57.