Eurographics Symposium on Geometry Processing (2007)
Alexander Belyaev, Michael Garland (Editors)

Linear Angle Based Parameterization

Rhaleb Zayer[1], Bruno Lévy[2], and Hans-Peter Seidel [1]

[1]Max-Planck-Institut für Informatik, Saarbrücken, Germany
[2]INRIA, project ALICE, France

Abstract

In the field of mesh parameterization, the impact of angular and boundary distortion on parameterization quality have brought forward the need for robust and efficient free boundary angle preserving methods. One of the most prominent approaches in this direction is the Angle Based Flattening (ABF) which directly formulates the problem as a constrained nonlinear optimization in terms of angles. Since the original formulation of the ABF, a steady research effort has been dedicated to improving its efficiency. As for any well posed numerical problem, the solution is generally an approximation of the underlying mathematical equations. The economy and accuracy of the solution are to a great extent affected by the kind of approximation used. In this work we reformulate the problem based on the notion of error of estimation. A careful manipulation of the resulting equations yields for the first time a linear version of angle based parameterization. The error induced by this linearization is quadratic in terms of the error in angles and the validity of the approximation is further supported by numerical results. Besides performance speedup, the simplicity of the current setup makes re-implementation and reproduction of our results straightforward.

1. Introduction

With the ever increasing computational power delivered by modern processors, it is possible to address a wide range of nonlinear problems in a reasonable time. This sheer power still has to deal with the increased size of data dictated by the strive for more detailed problem representations. This brings forward the need for efficient and reliable numerical tools capable of redesigning or reformulating these problems in a more tractable way. Among the long list of mesh parameterization methods [FH05, SPR06], some of the non-linear methods have the interesting property of computing natural boundaries and well balancing the deformations. In this paper, we focus on ABF (Angle Based Flattening) [SdS01], one of these non-linear methods. The numerical experiments conducted in the original paper and derivative works [SdS01, LdSS*01, SdS02, ZRS04, ZRS05, Sie06] suggest that it remains a challenging problem. Recently, a combination of hierarchical structure with an intelligent matrix decoupling approach was proposed in [SLMB05] allowing for increased performance. Nevertheless most of the proposed approaches so far address mainly the numerical issues arising at the optimization level and do not touch upon the setup of the original problem itself.

Within the ABF framework [SdS01], a set of linear and nonlinear constraints on the planar angles guarantees the validity of the embedding. The angles of the parametric representation are obtained as an approximate stationnary point of a Lagrangian function which punishes the deviation from a set of optimal angles and enforces the constraints. The standard iterative Newton scheme is commonly adopted for carrying out the minimization.

In this paper, a reformulation of the angle based flattening problem is laid out. Instead of working directly with angles, we address the problem in terms of the error in angle estimation, more specifically the angle difference between the optimal solution and an initial guess. Using these variables, we first follow the idea of Zayer *et. al* [ZRS05] of applying a log-transform to the equation. In this setup, a careful analysis of the nonlinear constraints reveals that they can be approximated by linear constraints. This approximation is well justified as the error induced by the linearization is quadratic in terms of the error in angles. In other words, this means than an error of the order of 10^{-3} in the angles induces an

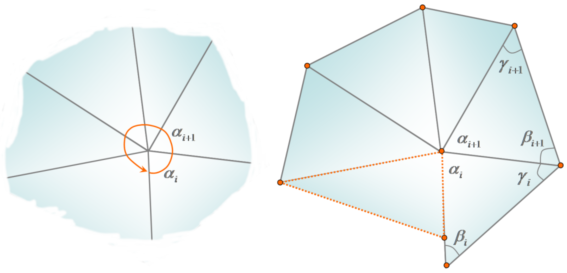


Figure 1: *Illustration of angle validity conditions. Vertex consistency (left) guarantees planarity, and wheel consistency guarantees closed vertex rings.*

error of the order of 10^{-6} in the constraints. Interestingly, this simple remark leads to a completely different and much simpler solution mechanism. The windfall of this new formulation is that the problem need not be addressed as a constrained optimization but as an underdetermined system of linear equations. We show that the latter is equivalent to a weighted least norm problem and can be solved using the normal equation. The resulting algorithm is up to 27 × faster than the algebraic ABF++ and 4 × faster than the hierarchical+algebraic ABF++ of [SLMB05]. More interestingly, it is also much simpler to implement than these latter methods. The statistics displayed in the results section show that we obtain nearly the same result as the original, non-linear ABF(++).

2. Planar angle constraints

In order to keep the exposition self-contained, we briefly summarize the angle constraints at the heart of the original ABF setup. Sheffer and de Sturler [SdS01] addressed the problem of the validity of the planar embedding by requiring the following consistency condition on the set of positive angles of the planar mesh:

- *Vertex consistency*
  For each internal vertex v, with central angles $\alpha_1^*,\ldots,\alpha_d^*$:
  $$\sum_{i=1}^{d} \alpha_i^* = 2\pi \qquad (1)$$
- *Triangle consistency*
  For each triangular face with angles α^*, β^*, γ^* the face consistency:
  $$\alpha^* + \beta^* + \gamma^* = \pi \qquad (2)$$
- *Wheel consistency*
  For each internal vertex v with left angles β_1^*,..,β_d^* and right angles $\gamma_1^*,\ldots,\gamma_d^*$:
  $$\prod_{i=1}^{d} \frac{\sin(\beta_i^*)}{\sin(\gamma_i^*)} = 1 \qquad (3)$$

Condition (1) enforces the planarity of vertex rings whereas condition (3) enforces the triangle sine rule over a vertex ring and guarantees the closedness of the ring. Failure to satisfy this condition yields the situation illustrated in figure (1).

3. Reformulation and linearization

In this section, we propose an alternative formulation of the problem, that leads to a linearization of the constraints. Linearization was already used in previous methods (e.g. ABF++). However, in our case, before linearizing the constraints, we carefully reformulate the problem in terms of alternative variables, that will make this linearization so accurate that solving single linear system will converge to the solution directly without requiring multiple Newton steps used in previous work. In more details, our approach is based on the notion of *error adjustement*, i.e. it uses the relative error of estimation of the angles rather than their absolute values.

Let us denote the ideal angles which solve the parametrization problem by α^* and the initial guess as α , the estimation error is then given by

$$\alpha^* \; = \; \alpha \, + \, e_\alpha \qquad (4)$$

The variables α_i represent an initial estimation of the angles of the flat mesh and will be discussed later in the paper.

In this setup the constraints on the planar angles read :

- *Vertex consistency*
  For each internal vertex v, with central angles $\alpha_1,\ldots,\alpha_d$:
  $$\sum_{i=1}^{d} e_i = 2\pi \, - \, \sum_{i=1}^{d} \alpha_i \qquad (5)$$
- *Triangle consistency*
  For each triangular face with angles α, β, γ the face consistency:
  $$e_\alpha + e_\beta + e_\gamma = \pi \, - \, (\alpha \, + \, \beta \, + \, \gamma) \qquad (6)$$
- *Wheel consistency*
  Based on the logarithmic modification introduced in [ZRS05], we have for each internal vertex v with left angles β_1,..,β_d and right angles $\gamma_1,\ldots,\gamma_d$:
  $$\sum_{i=1}^{d} \log\left(\sin\beta_i \, + \, e_{\beta_i}\right) - \log\left(\sin\gamma_i \, + \, e_{\gamma_i}\right) = 0. \qquad (7)$$

The changes introduced so far do not affect the nature of the linear conditions. On the other hand, the nonlinear expression in (7) looks as if it just got more complicated. However, the Taylor expansion of $\log(\sin(\alpha \, + \, e))$ can be written as

$$\begin{aligned} \log(\sin(\alpha \, + \, e)) \; = \; & \log(\sin(\alpha)) + \cot(\alpha)\, e \\ & - \frac{1}{2}(1 \, + \, \cot^2(\alpha))\, e^2 \qquad (8) \\ & + \cdots \end{aligned}$$

Inspection of this series reveals that we can safely use the approximation

$$\log(\sin(\alpha + e)) \backsimeq \log(\sin(\alpha)) + \cot(\alpha)\, e. \qquad (9)$$

The error induced by this approximation depends quadratically on the error in angle e. In other words, for a small error e in the angle estimation, the error in the consistency constraint is even smaller (e.g. an angle error of the order of 10^{-3} induces an error of order 10^{-6}). This point is of utter importance to the method and it is in fact similar to the linear approximation generally used in finite elements for approximating potential energy as discussed in e.g. [Bra01]. Considering that even in the most general case enforcing equality constraints amount to a minimization up to a certain reasonable accuracy, our approximation is then well justified. We will also backup this claim with numerical experiments in the results section.

In the light of this new approximation, the nonlinear equation (7) can be then replaced by following linear variant

$$\sum_{i=1}^{d} \cot(\beta_i)\, e_{\beta_i} - \cot(\gamma_i)\, e_{\gamma_i} = \sum_{i=1}^{d} \log(\sin\gamma_i) - \log(\sin\beta_i) \qquad (10)$$

The term on the right hand side measures the error in the wheel consistency condition induced by the initial estimation. Similarly the right hand sides of equations (5), (6) measure triangle consistency and the angular deficit respectively. In this way, given an angle estimation α, we describe the error induced on the constraints as linear function of the estimation error e.

4. Numerical solution

At this stage a least norm solution to the resulting underdetermined system of linear equations can be readily obtained through the normal equation. This setup however is even-handed as it treat all angles in the same way and at times, this may cause instabilities for very small and very large angles. One straightforward approach consists of using additional bounds on the error e and solving the system using standard techniques e.g. Matlab$^{\text{TM}}$Optimization Toolbox. However, as this work is geared towards simple implementation, it is more interesting to maintain the new gains from the linearization of the constraints and associate an objective function with the constraints which allows for introducing additional weights to control the errors in similar fashion to the original ABF [SdS01]. The weighted objective function described in the following subsection allows for a balanced treatment of angles by penalizing large angles and enforcing smaller ones.

4.1. Normal equation setup

In this subsection, We aim at minimizing a weighted error objective function while enforcing the equality constraint.

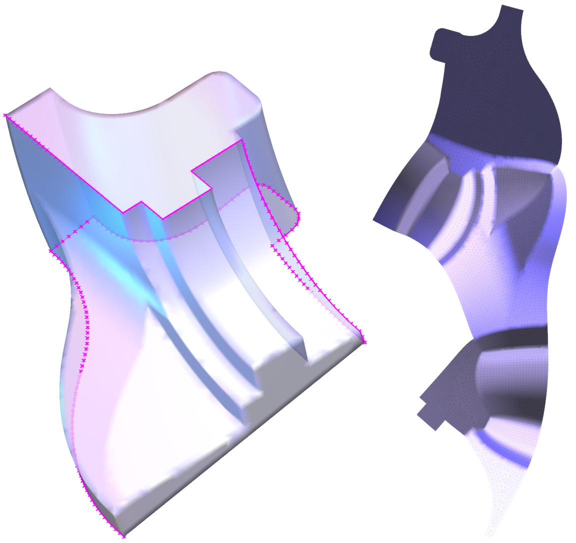


Figure 2: *Parameterization of the fan disk model (*$13K\Delta$*). Solution runtime (*$0.15s$*).*

In the light of the new representation in terms of the error, the objective function

$$F(\alpha) = \sum_{i=1}^{N} \frac{1}{\alpha_i^2}\, (\alpha_i^* - \alpha_i)^2, \qquad (11)$$

can be stated as:

$$\text{minimize} \sum_{i=1}^{N} \frac{1}{\alpha_i^2}\, e_i^2 \quad \text{subject to } Ae = b \qquad (12)$$

using a simple change of variables

$$r_i = \frac{e_i}{\alpha_i} \qquad (13)$$

In matrix notation this change of variables gives $e = D_\alpha\, r$, where $D_\alpha = \text{diag}(\alpha_i)$ is the diagonal matrix with the angles as entries. Thus the angle parameterization problem reads as simple as

$$\text{minimize } ||r||^2 \quad \text{subject to } Cr = b \qquad (14)$$

This is now clearly a least-norm problem. The size of the matrix $C = A\, D_\alpha$ is $(n_t + 2 \cdot n_i) \times (3 \cdot n_t)$, where n_t is the number of triangles and n_i is the number of internal vertices. As the equality constraints are independent, the matrix C has full rank, and it ensures that the least-norm problem has a unique solution (see e.g. [Lue69]) :

$$r = C^T\, (C\, C^T)^{-1} b \qquad (15)$$

Thus, the problem can be solved by finding a solution to the normal equation

$$(C\, C^T)x = b \qquad (16)$$

After solving this equation, r can be obtained as $r = C^T\, x$.

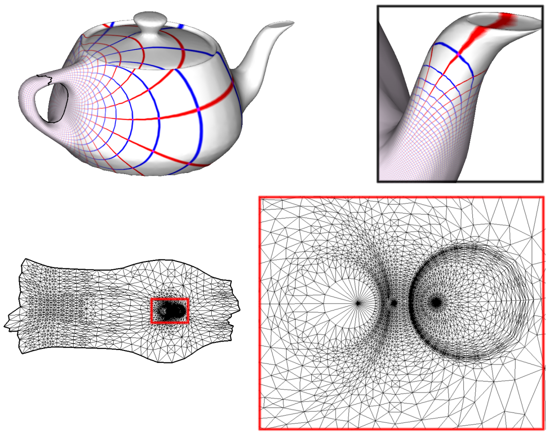


Figure 3: *This teapot, with high Gauss curvature, is a numerical challenge for parameterization methods. As can be seen, our linearization satisfies the constraints and produces a valid parameterization, even in this difficult configuration.*

The angles of the mesh in the parametric domain can be obtained by substituting back in equations (13) and (4).

4.2. Choice of initial estimation

In order to reduce error in the above presented method, the choice of the initial estimation is very important as it directly affect the global error. For this purpose it is imperative that very large and very small angles do not force the solution out of the $(0,\pi)$ domain. Setting α equal to the original angles of the mesh yields valid parameterizations in most cases. However, it is not difficult to tailor cases which yield invalid angles. In order to enforce a valid solution for general cases, we set a threshold on the error of fan angles of vertices and also on angles at the vicinity of 0 and π. When the error associated with a fan is large (e.g. spikes), we replace the original angles by the angles obtained from an exponential map of the vertex one-ring. They are given in terms of the original angles original angles α_i^o as

$$\alpha_i = \begin{cases} \alpha_i^o \frac{2\pi}{\sum_{i=1}^{d} \alpha_i^0} & \text{around an interior vertex} \\ \alpha_i^o & \text{around a boundary vertex} \end{cases}$$

In practice for vertex rings with an angular deficit larger than 1 it is recommendable to switch to the angles obtained from the exponential map for that specific ring. For example, in the obtuse case illustrated in figure (4), the angular deficit (4.56) is very large and triggers the threshold switch.

5. Algorithmic outline

The algorithmic approach outlined in this paper is simple and easy to implement. The whole algorithm for setting up the normal equation system and solving for the angles spans around 30 lines of vectorized Matlab$^{\text{TM}}$code. The algorithmic flow can be summarized in the following steps:

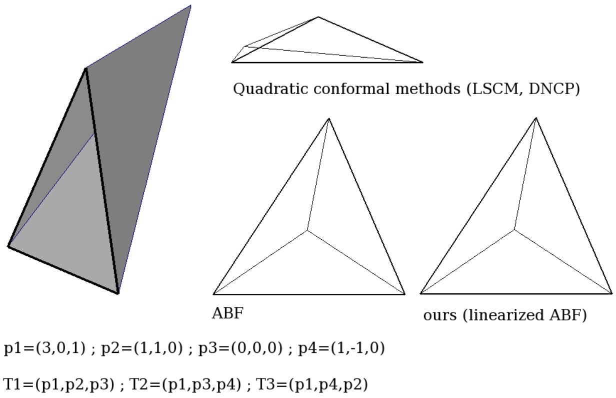

Quadratic conformal methods (LSCM, DNCP)
ABF
ours (linearized ABF)
p1=(3,0,1) ; p2=(1,1,0) ; p3=(0,0,0) ; p4=(1,-1,0)
T1=(p1,p2,p3) ; T2=(p1,p3,p4) ; T3=(p1,p4,p2)

Figure 4: *The simple example shown here is known to make linear conformal parameterization methods (LSCM [LPRM02], DNCP[DMA02]) generate an invalid parameterization. As shown here, our linearized method generates the same (valid) result as ABF.*

a. Establish the initial angle estimation α as explained in subsection (4.2)

b. Setup the constraints, i.e. vertex consistency (equations 5), triangle consistency (equation 6), and linearized wheel consistency (equation 10), as a linear system, $A\, e_\alpha \;=\; b$

c. Compute $C \;= AD_\alpha$, where $D_\alpha = \mathrm{diag}(\alpha_i)$ is the diagonal matrix described in subsection(4.1)

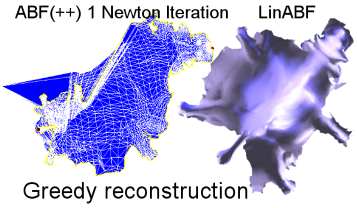
d. Solve for x in $(CC^T)x = b$ (equation 16)

e. Compute the estimation error $e_\alpha = D_\alpha C^T x$ (equations 13 and 4)

f. Get the angle solution as $\alpha^* \;=\; \alpha \,+\, e_\alpha$

One can use two approaches for obtaining the uv coordinates from the angles. The first one is a greedy reconstruction, which constructs the triangles one by one using a depth-first traversal. The second one is an angle based least squares formulation which solves a set of linear equations relating angles to coordinates [SLMB05].

In the greedy approach, the last triangle of a fan is never explicitly constructed as two of its edges must have already been constructed. When reconstructing with non-optimal angles, error accumulation may lead to degenerate meshes (see figure). Although our method behaves generally well with the greedy approach (see figure), we recommend using the least squares reconstruction, since it better balances the cumulative error, especially for large meshes. We used that approach for all our experiments.



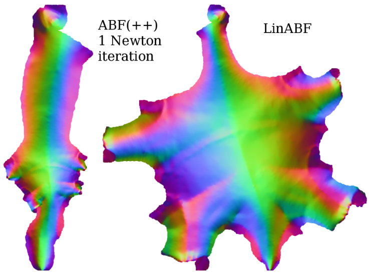
6. Results

The current method was tested on a benchmark of nontrivial meshes. Table (1) shows typical values of the error in angles induced by the original ABF++ and the current linearized version for the models depicted in the paper. The angular distortion measure we use is $\|e_\alpha\|^2/3nt$, where nt denotes the number of triangles. Most of these meshes were made homeomorphic to a disk by using Seamster [SH02]. To make sure that convergence comparison is accurate, we only used the algebraic transform of ABF++ (and did not use the hierarchical+algebraic HABF++). No difference is visually noticeable in the results. Timings are up to 27 × faster than the algebraic ABF++ (or up to 4 × faster than HABF++).

The numerical examples confirm the validity of the approximation used in equation (9). This is further illustrated in figures (2), (6), (7), and (8). Besides speed, the main advantage of our approach is that the setup of the problem is simplified to a great extent in comparison to previous work, that require both complex sparse matrix manipulation and a hierarchical mesh data structure. In contrast, reproduction of our results is straightforward. This way the performance of the angle based parameterization becomes comparable to the well established discrete versions of conformal maps (see [FH05, SPR06]), while keeping the much better balance of deformations achieved by ABF.

We experimented the method with a large number of complicated test cases, including surfaces with high curvature (see Figure 3), and the example known to make linear conformal parameterization methods fail ('obtuse' entry in Table 1 and Figure 4). Although this is not guaranteed, for all these test cases, a valid parameterization was obtained. A failure case is shown in Figure 5. In such a (very unlikely) configuration, one can use multiple iterations. The angles computed at one iteration are constrained in $[0,\pi]$ and used to define the α's for the next iteration. In other words, we use a constrained Newton method with an active set approach.

In terms of memory consumption, all the tests of our linearized method were conducted on a computer with 1Gb of system RAM, whereas ABF++ required more than 2Gb for some meshes.



For comparison, we have also experimented how a single iteration of ABF++ performs. For several meshes, it gives a result similar to ours (at the expense of a much more complex implementation). However, for some meshes, as shown in the small figure, this gives a result inbetween LSCM and ABF. In contrast, our approach better balances deformation[SSGH01], which is also reflected by the following statistics :

	$F/3nt$ ABF	Stretch L2 ABF	$F/3nt$ linABF	Stretch L2 linABF
Horse	8.847e-4	1.89	2.906e-4	1.11
Dino	2.794e-3	2.937	1.184e-3	1.22

model	$\sharp\Delta$	timing ABF++	timing linABF	$F/3nt$ ABF++	$F/3nt$ linABF
obtuse	3	0.1s	0.05s	1.267	1.267
cow	5.8K	0.45 s	0.1 s	5.041e-3	5.256e-3
fandisk	13 K	0.85 s	0.15 s	5.041e-3	5.256e-3
teapot	14K	3 s	0.3 s	2.154e-3	2.282e-3
foot	20K	2 s	0.4 s	1.867e-4	1.921e-4
gargo	20K	2.5 s	0.4 s	1.603e-3	1.604e-3
bull	34K	4 s	0.8 s	5.323e-4	5.331e-4
bunny1	40K	5.5	0.9 s	2.597e-4	2.593e-4
dino	48K	15 s	1 s	1.363e-3	1.184e-3
kiss	48K	8 s	1 s	7.092e-4	7.109e-4
tweety	54K	8.6s	1.5 s	1.5671e-4	1.5672e-4
bunny2	70K	13s	2 s	2.232e-4	2.243e-4
hand	73K	13 s	2 s	1.191e-4	1.213e-4
camel	78K	23 s	2.5 s	5.896e-4	6.202e-4
horse	97K	34 s	3 s	2.746e-4	2.906e-4
man	120K	36 s	2.7 s	5.293e-4	5.602e-4
head	128K	87 s	3.5 s	1.243e-4	1.240e-4
male	293K	272 s	9.5 s	3.577e-4	3.841e-4
isis	374K	250 s	11.5 s	4.834e-5	4.7941e-5
david	505K	355 s	12 s	5.776e-4	5.844e-4

Table 1: *Timings and angular deviation.*

This can be explained as follows : a single iteration of ABF++ computes angles that do not satisfy the constraints, then the LSCM-based reconstruction transforms them into a valid parameterization, but fails balancing deformations. Therefore all our comparisons were made to the fully converged ABF++.

Discussion

As in previous constrained optimization based methods, in this work, the constraints are satisfied up to a certain precision. In Newton-based approaches, each step improves the approximation by linearizing the gradient of the (non-linear) Lagrangian. In our work, we perform a Taylor expansion at the level of the non-linear constraints, thus avoiding non-linear optimizations in the first place.

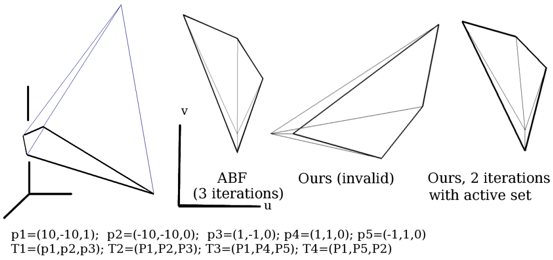

v
u
ABF
(3 iterations)
Ours (invalid)
Ours, 2 iterations
with active set
p1=(10,-10,1); p2=(-10,-10,0); p3=(1,-1,0); p4=(1,1,0); p5=(-1,1,0)
T1=(p1,p2,p3); T2=(P1,P2,P3); T3=(P1,P4,P5); T4=(P1,P5,P2)

Figure 5: *An example that makes our 1-iteration method fail. An additional iteration fixes the problem.*

We tested our method on a representative benchmark of meshes. No triangle flips were detected on the results. As shown in Figure 5, it is possible however to engineer specific situation where a single iteration may fail. This would happen for example when a single one ring is brought to have an obtuse solid angle and sheared triangles. However, such a situation is seldom encountered in practice, and can be fixed by using our method with an active set approach.

Conclusion

We presented a complete reformulation of the angle based parameterization problem. Working directly with the approximation error instead of angles we developed a linearized version of the challenging nonlinear constraints associated with this type of parameterization. In the light of this new representation, the planar angles can be obtained as a solution to a least norm problem. The approximations used in our framework are well justified and lead to easier implementation and faster solution in comparison to previous nonlinear formulations.

Furthermore, an even faster method may be obtained by combining our linearization with the hierarchical acceleration technique used by ABF++. We plan also to investigate incorporating global non-intersection boundary constraints in our framework as well as properly handling meshes with holes.

Acknowledgment

We would like to thank the reviewers for their constructive comments. This work is supported by aim@Shape (European network of excellence), georep (Inria grant) and geometric intelligence (Microsoft Research grant). We thank Alla Sheffer for providing the seamster-ized meshes.

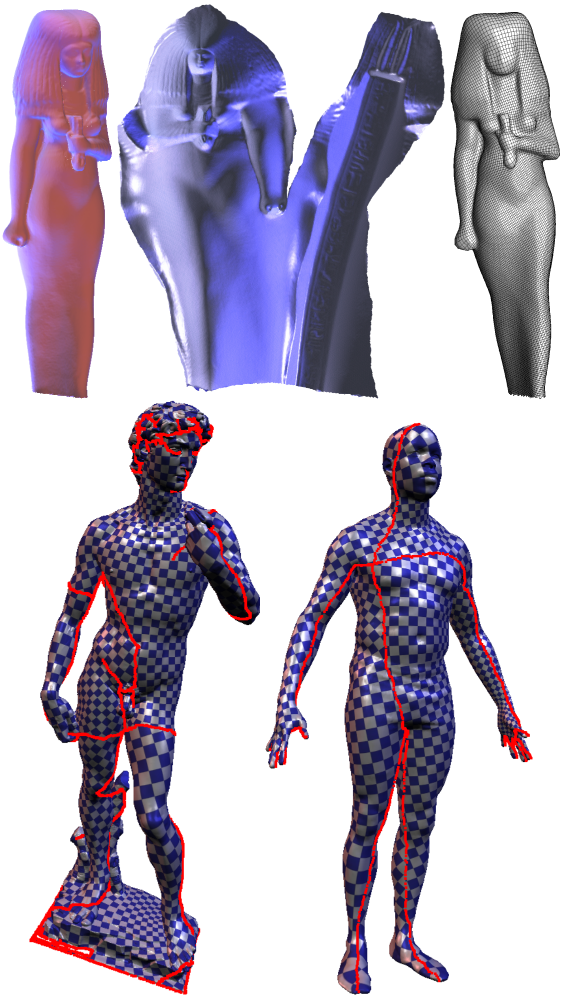


Figure 6: *Parameterization of the Isis ($374K\Delta$), David ($505K\Delta$), and man ($120K\Delta$) models. Quadrangular remeshing and texture mapping reflect the quality of the parameterization. Respective runtimes are (11.5s), (12s), and (2.7s).*

References

[Bra01] BRAESS D.: *Finite elements*, second ed. Cambridge University Press, Cambridge, 2001. 3

[DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. *Computer Graphics Forum (Proc. Eurographics) 21*, 3 (2002), 209–218. 4

[FH05] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization. Springer, 2005, pp. 157–186. 1, 5

[LdSS*01] LIESEN J., DE STURLER E., SHEFFER A., AYDIN Y., SIEFERT C.: Preconditioners for indefinite linear systems arising in surface parameterization. In *Proc. of the 10th Intl. Meshing Round Table* (2001), pp. 71–82. 1

[LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 21, 3* (2002), 362–371. 4

[Lue69] LUENBERGER D. G.: *Optimization by vector-space methods*. Wiley-Interscience, 1969. 3

[SdS01] SHEFFER A., DE STURLER E.: Parameterization of faceted surfaces for meshing using angle based flattening. *Engineering with Computers 17*, 3 (2001), 326–337. 1, 2, 3

[SdS02] SHEFFER A., DE STURLER E.: Smoothing an overlay grid to minimize linear distortion in texture mapping. *ACM Transactions on Graphics 21*, 4 (2002), 874–890. 1

[SH02] SHEFFER A., HART J.: Seamster: Inconspicuous low-distortion texture seam layout. In *Proceedings of IEEE Visualization* (2002). 5

[Sie06] SIEFERT C.: *Preconditioners for Generalized Saddle-Point Problems*. PhD thesis, University of Illinois at Urbana-Champaign, 2006. 1

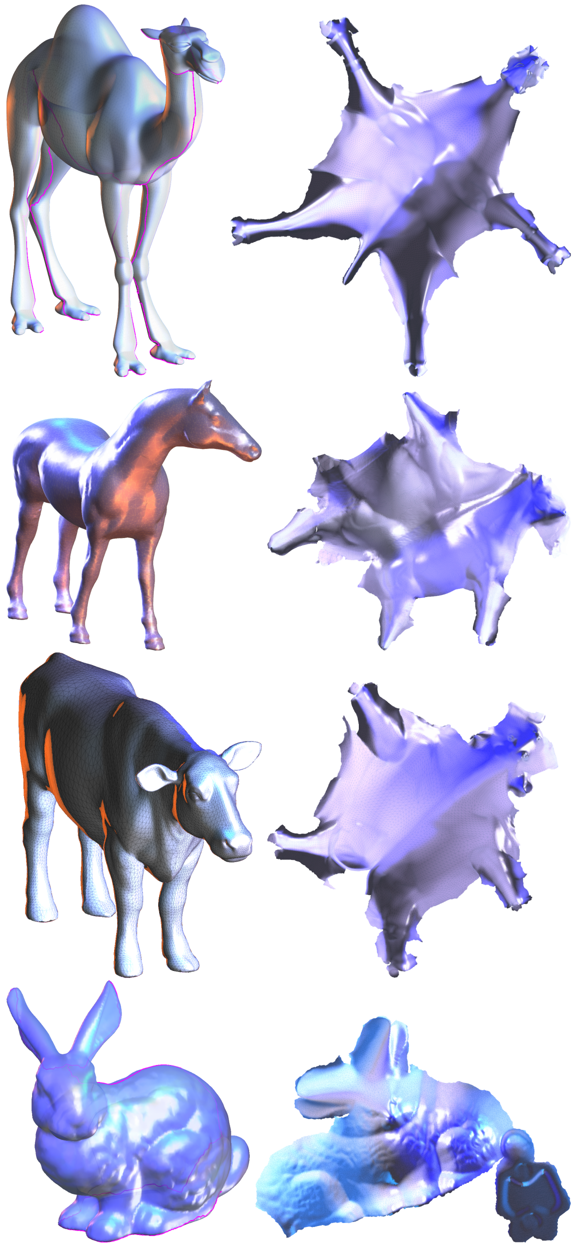


Figure 7: *Flattening of several animal models. Model sizes and runtime are given in table(1).*

[SLMB05] Sheffer A., Lévy B., Mogilnitsky M., Bogomyakov A.: ABF++: fast and robust angle based flattening. *ACM Trans. Graph. 24*, 2 (2005), 311–330. 1, 2, 4

[SPR06] Sheffer A., Praun E., Rose K.: Mesh parameterization methods and their applications. *Foundation and Trends in Computer Graphics and Vision 2*, 2 (2006), 105–171. 1, 5

[SSGH01] Sander P. V., Snyder J., Gortler S. J., Hoppe H.: Texture mapping progressive meshes. In *Proc. SIGGRAPH ’01* (2001), ACM Press, pp. 409–416. 5

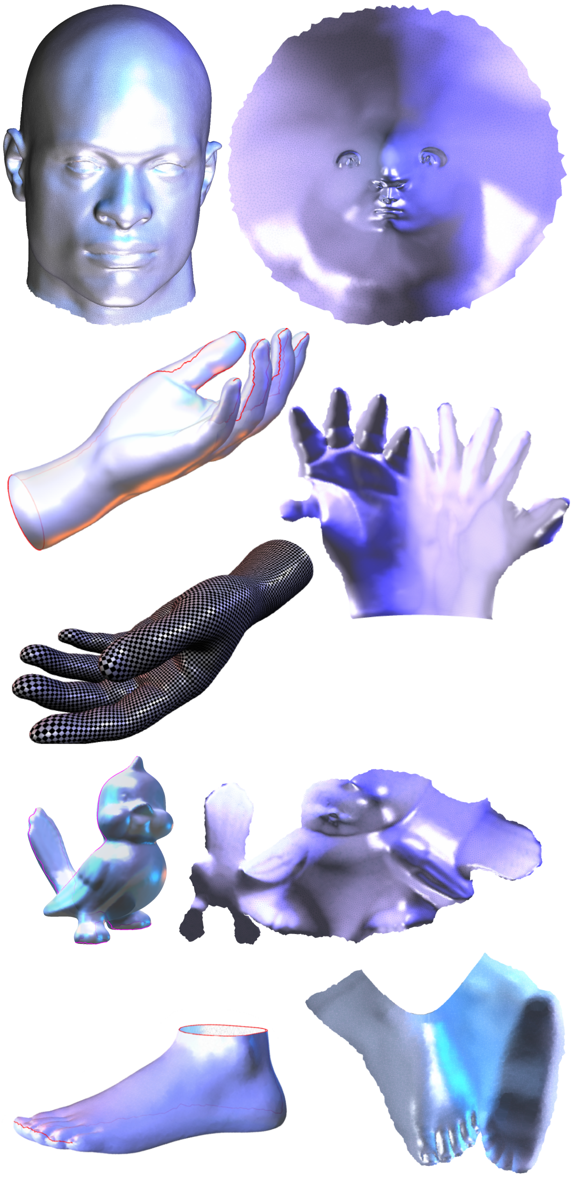


Figure 8: *Parameterization results of the head, hand, tweety, and foot models.*

[ZRS04] Zayer R., Rössl C., Seidel H.-P.: Efficient iterative solvers for angle based flattening. In *Vision, modeling, and visualization* (Stanford, USA, 2004), pp. 347–354. 1

[ZRS05] Zayer R., Rössl C., Seidel H.-P.: Variations of angle based flattening. In *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization. Springer, 2005, pp. 187–199. 1, 2