

Linear-Time Estimation of Smooth Rotations in ARAP Surface Deformation

Mauricio Cele Lopez Belon

Abstract. In recent years the As-Rigid-As-Possible with Smooth Rotations (SR-ARAP [5]) technique has gained popularity in applications where an isometric-type of surface mapping is needed. The advantage of SR-ARAP is that quality of deformation results is comparable to more costly volumetric techniques operating on tetrahedral meshes. The SR-ARAP relies on local/global optimization approach to minimize the non-linear least squares energy. The power of this technique resides on the local step. The local step estimates the local rotation of a small surface region, or cell, with respect of its neighboring cells, so a local change in one cell's rotation affect the neighboring cell's rotations and viceversa. The main drawback of this technique is that the local step requires a global convergence of rotation changes. Currently the local step is solved in an iterative fashion, where the number of iterations needed to reach convergence can be prohibitively large and so, in practice, only a fixed number of iterations is possible. This trade-off is, in some sense, defeating the goal of SR-ARAP. We propose a linear-time closed-form solution for estimating the codependent rotations of the local step by solving a sparse linear system of equations. Our method is more efficient than state-of-the-art since no iterations are needed and optimized sparse linear solvers can be leveraged to solve this step in linear time. It is also more accurate since this is a closed-form solution. We apply our method to generate interactive surface deformation, we also show how a multiresolution optimization can be applied to achieve real-time animation of large surfaces.

Mathematics Subject Classification (2010). Parallel algorithms 68W10; Clifford algebras, spinors 15A66.

Keywords. Geometric Algebra, Quaternion Estimation, Mesh Deformation.

1. Introduction

Quality of surface deformation aims to preserve local properties of shapes as much as possible, removing distortion in the form of shear and stretch. As-Rigid-As-Possible with Smooth Rotations (SR-ARAP) [5] aims to create a surface mapping that minimize the deviation from rigid behavior on the local scale. It also makes rigid transformations smoothly vary on local neighborhoods distributing the distortions uniformly over the surface. The resulting deformations not only preserves the shape at small scale but also at large scale significantly increasing the quality of the final result.

There are two main disadvantages on the current SR-ARAP method. First is the local step, which looks for best rotations of corresponding surface cells while keeping neighbor rotations similar to each other. It needs to reach global convergence for the whole mesh. Currently the local step is solved with a relaxation method i.e., optimizing rotations for individual cells keeping cell's neighbor rotations fixed and also optimizing neighbor rotations in the same manner repeating that process until convergence. Although the cost per relaxation iteration can be reduced considerably by avoiding SVD calculations ([11]) usually a fixed number of iterations is used in order to bound the solver time. That compromises the quality of results since a fixed number of iterations (usually two or three) is not enough to reach optimal rotations across the surface. Second is the high computational cost, that increases drastically when the number of vertices grows. The SR-ARAP is a non-linear technique that requires several iterations of local/global optimization to converge.

In this paper we address both drawbacks of SR-ARAP, first we propose a method to solve the local step as a single linear system of equations, avoiding the need of iteratively solve a series of SVD problems which are codependant to each other. Second we propose a simple multiresolution method based on solving the non-linear system on a simplified surface mesh first and then use harmonic interpolation of rotations to transfer the optimized rotations to the full resolution mesh, leveraging the Laplacian matrix needed to solve the global step of the optimization.

2. Related Work

Since the focus of this paper is on As-Rigid-As-Possible (ARAP) surface deformation methods we will only review previous works on ARAP deformation. The ARAP Surface Modeling was introduced by Sorkine and Alexa [10] to produce robust and physically plausible deformation by minimizing the local rigid transformation of the laplacian coordinates associated to every vertex. Although results on low resolution meshes looks close to physical deformations the method fails to achieve satisfactory results on high resolution meshes. The reason is that the ARAP energy is minimized by preserving rigidity on large parts of the mesh at the expense of concentrating all the distortion on small parts of the mesh. This problem has been solved in the

Smooth Rotation enhanced ARAP method (SR-ARAP) of Levi and Gotsman [5] where the deformation energy is extended with a further factor that penalizes too different rotations of close vertices, significantly increasing the quality of the final result. Chao *et al* [1] derives the ARAP energy from the elastic energy of continuum mechanics showing a consistent discretization for volumetric ARAP with tetrahedron cells and for 2D ARAP with triangle cells in 2D. However the ARAP energy doesn't have a consistent discretization for triangle surfaces in 3D. The SR-ARAP method [5] also address this issue providing an energy with a consistent discretization for surfaces in 3D. Further enhancements have been proposed to the SR-ARAP deformation technique addressing performance issues on large meshes [8, 11], making it able to adjust physical stiffness [2, 4] and Introducing local scaling to the rotation i.e., turning rigid transforms into a similarity transforms [3].

3. As-Rigid-As-Possible Shape Deformation with Smooth Rotations

Given two meshes Q and P consisting of vertices q_i and p_i respectively, and directed edges $p_{ij} = p_j - p_i$ and $q_{ij} = q_j - q_i$, the discrete As-Rigid-As-Possible with Smooth Rotations (SR-ARAP) energy is defined as:

$$E(Q, P) = \sum_{i=1}^m \left(\sum_{j \in N(i)} c_{ij} \|R_i(q_j - q_i)\tilde{R}_i - (p_j - p_i)\|^2 + \alpha A \sum_{j \in N(i)} w_{ij} \|R_j - R_i\|^2 \right)$$

where $R_1, \dots, R_m \in \mathbb{H}$ are local quaternions, $N(i)$ denote the set of 1-ring neighbors of vertex at position i , c_{ij} are weighting coefficients such that $\sum_j^n c_{ij} = 1$, typically the familiar cotangent weights, w_{ij} are positive weighting coefficients such that $\sum_j^n w_{ij} = 1$, A is the mesh area used to make the energy scale invariant and α is a positive scalar parameter.

The main idea of this method is breaking the surface into overlapping cells and seek for keeping the cells transformations as rigid as possible in the least squares sense. Overlap of the cells is necessary to avoid surface stretching or shearing at the boundary of the cells. The first term is a membrane energy which penalizes stretching and shearing of a cell and the second term is the bending energy which penalizes the difference between a cell's rotation and the rotations of its neighboring cells. The objective of the membrane term is to lower the distortion of a cell by keeping the map differential close to rigid. The objective of the bending term is to keep the variation in the rotations in a cell neighborhood low, such that the neighborhood would transform as a unit, as much as possible.

The vertices of mesh Q are in original position while vertices of mesh P are the deformed vertices and the quaternion R_i is the best rigid transformation, in the least squares sense, relating the original and the deformed vertices. This is a non-linear optimization problem that is typically solved by a iterative local/global method that solves two linear sub-problems on each iteration.

The first step, so called local step, is to consider the vertices of P constant and obtain the best rigid transformation R_i for each cell. The second step, so called global step, is to consider the rotations R_i constant and computing the optimal deformed vertices p_i in the least squares sense.

4. Global Step

The global step is computing the optimal vertices $\{p_i\} \in P$.

$$E(P) = \min_{p_1, \dots, p_m \in \mathbb{R}^3} \sum_{i=1}^m \left(\sum_{j \in N(i)} c_{ij} \|R_i q_{ij} \tilde{R}_i - p_{ij}\|^2 + \alpha A \sum_{j \in N(i)} w_{ij} \|R_i - R_j\|^2 \right)$$

Taking the partial derivatives of $E(P)$ w.r.t. p_i and equating the result to zero lead us to obtain the linear system of Equation (4.1) ([6]):

$$\sum_{j \in N(i)} c_{ij} (p_j - p_i) = \sum_{j \in N(i)} \frac{c_{ij}}{2} (R_i q_{ij} \tilde{R}_i + R_j q_{ij} \tilde{R}_j) \quad (4.1)$$

which can be expressed in matrix form as $L P = C$, where L is the discrete Laplace-Beltrami operator, P is the column of target positions and C a column vector whose i th row is the right hand side of equation (4.1). Constraints of the form $p_i = p_i^{const}$ are incorporated into the system by substituting the corresponding variables i.e., erasing respective rows and columns from L and updating the right-hand side of equation (4.1) with the values p_i^{const} . The system is then solved in the least squares sense:

$$(L^T L) P = L^T C \quad (4.2)$$

5. Local Step

As described in a previous sections, the rotations matching cell's are codependent to each other over the whole mesh. The current approach is to optimize the rotations with a relaxation method in which the rotation of each cell is independently computed, while keeping the neighbor rotations fixed, repeating it until global convergence is reached. At least two relaxation iterations must be done per each global iteration. In this section we show how the local step can be solved in closed form as a single sparse linear system eliminating entirely the need of relaxation iterations.

5.1. Closed Form of Local Step

We attempt to minimize the SR-ARAP energy function:

$$E(R) = \min_{R_1, \dots, R_m \in \mathbb{H}} \sum_{i=1}^m \left(\sum_{j \in N(i)} c_{ij} \|R_i q_{ij} \tilde{R}_i - p_{ij}\|^2 \right) \quad (5.1)$$

$$+ \alpha A \sum_{j \in N(i)} w_{ij} \|R_i - R_j\|^2 \quad (5.2)$$

Ignoring temporarily the sum over all vertices, the energy to minimize in the neighborhood of a point p_i is $E_i(R_i)$:

$$E_i(R_i) = \sum_{j \in N(i)} c_{ij} \|R_i q_{ij} \tilde{R}_i - p_{ij}\|^2 + \alpha A \sum_{j \in N(i)} w_{ij} \|R_j - R_i\|^2$$

Notice that the first term $\sum_{j \in N(i)} c_{ij} \|R_i q_{ij} \tilde{R}_i - p_{ij}\|^2$ can be written in matrix language as the quadratic form $R_i^T M_i R_i$ where M_i is a 4×4 matrix constructed from vectors q_{ij} and p_{ji} (see Section 7):

$$E_i(R_i) = R_i^T M_i R_i + \alpha A \sum_{j \in N(i)} w_{ij} \|R_j - R_i\|^2 \quad (5.3)$$

Differentiating with respect to R_i we get:

$$\frac{\partial E_i(R_i)}{\partial R_i} = 2M_i R_i + 2\alpha A \sum_{j \in N(i)} w_{ij} (R_i - R_j) \quad (5.4)$$

$$= M_i R_i + \alpha A R_i - \alpha A \sum_{j \in N(i)} w_{ij} R_j \quad (5.5)$$

Setting the partial derivatives to zero $\frac{\partial E_i(R_i)}{\partial R_i} = 0$

$$(M_i + \alpha A I) R_i = \alpha A \sum_{j \in N(i)} w_{ij} R_j \quad (5.6)$$

Which is a linear system of equations. The system (5.6) can be solved for all quaternions in closed form. Writing (5.6) in matrix form we get:

$$M R = W R \quad (5.7)$$

$$(M - W) R = 0 \quad (5.8)$$

Where M is a symmetric sparse matrix with dimensions $4n \times 4n$ which is stacking M_i at its diagonal, W is a discrete Laplacian matrix with dimensions $4n \times 4n$ derived from RHS of (5.6) i.e., $\sum_{j \in N(i)} \alpha A w_{ij} R_j$ holding the coefficients $\alpha A w_{ij}$. R is column matrix of dimensions $1 \times 4n$ stacking the coefficients of quaternions R_i .

The sparse linear system of (5.7) can be solved by imposing quaternion constraints R_i^{const} which corresponds to best rotation of constrained points p_i^{const} :

$$\begin{aligned} (M - W)R &= 0 \\ \text{s.t. } R_i &= R_i^{const} \end{aligned} \quad (5.9)$$

Constraints of the form $R_i = R_i^{const}$ are incorporated into the system by substituting the corresponding variables i.e., erasing respective rows and columns from $(M - W)$ and updating the right-hand side of (5.9) with the values R_i^{const} . The system is then solved in the least squares sense.

Since the constraint $R_i^T R_i = 1$ is not honored by the linear system the quaternion R_i given as solution must be normalized. As shown in [11] the solution of (5.6) gives a linear approximate solution to the optimal quaternion in the least squares sense. Our results confirm that the proposed linearization is accurate for the SR-ARAP rotations and it is not affecting accuracy in any significant way.

5.2. Rotation constraints R_i^{const}

The rotation constraints R_i^{const} in (5.9) have to be computed directly from equation (5.6) for each constrained point p_i^{const} :

$$R_i^{const} = (M_i + \alpha AI)^{-1} \sum_{j \in N(i)} \alpha A w_{ij} R_j^{prev}$$

where neighbor rotations R_j^{prev} are known from previous iteration. So computing constraint quaternions amounts to solve a small 4×4 linear system. The resulting quaternion R_i^{const} must be normalized.

5.3. Rotation's Feedback

The linear system of (5.9) find quaternions from scratch i.e., it doesn't take into account the previous state of the quaternions i.e., the temporal coherence. That might lead to undesired results in an animation sequence. In particular, the sense of the rotations found by (5.9) for some animation step are not always respecting the sense of rotations from previous time. The abrupt change in the sense of rotation cause animation artifacts specially for large rotations.

The equation (5.9) allow us to introduce *feedback* quaternions to the RHS which acts as *hints* to find quaternions close to the ones in previous iteration. Given the quaternion R_i^{prev} and a positive small scalar value ϵ_i we augment equation (5.6) in the following way:

$$(M_i + \alpha AI)R_i = \alpha A \sum_{j \in N(i)} w_{ij} R_j + \epsilon R_i^{prev} \quad (5.10)$$

where $\sum_{j \in N(i)} w_{ij} = 1 - \epsilon$. Our intention is to make R_i a neighbor of itself, in some sense. The strength of feedback is given by ϵ . So the global system

is changed as follows:

$$\begin{aligned} (M - W)R &= \epsilon R^{prev} \\ \text{s.t. } R_i &= R_i^{const} \end{aligned} \quad (5.11)$$

where R^{prev} is a $1 \times 4n$ column vector stacking all the unconstrained quaternions R_i^{prev} from previous iteration.

6. Multiresolution Optimization

For achieving real-time performance we optimize the SR-ARAP energy in a low resolution version of the input mesh and then we transfer that solution to the full resolution mesh. To obtain the low resolution mesh we simplify the mesh using *half edge collapses* (i.e., the simplified mesh is a triangulation of a subset of the original vertices) while minimizing the Quadrics error metric. After we obtained the optimal deformed shape on the simplified mesh we transfer the optimized rotations to the full resolution mesh using Harmonic Interpolation of quaternions (see Section 6.1) and then solve the following linear system using the optimized vertices of the low resolution mesh as positional constraints:

$$\sum_{j \in N(i)} c_{ij}(p_j - p_i) = \sum_{j \in N(i)} \frac{c_{ij}}{2}(R_i q_{ij} \tilde{R}_i + R_j q_{ij} \tilde{R}_j)$$

6.1. Harmonic Interpolation of Quaternions

The seminal work of Pinkall and Polthier [9] shows how to interpolate given data over a discrete domain using harmonic maps. Harmonic maps are critical points of the Dirichlet energy (stretching energy) $\int_{\Omega} |\nabla f|^2 dA$, which generates minimal surfaces. The *discrete harmonic energy* of a map f defined on mesh vertices $\{p_i\}$ has the form:

$$E(f) = \sum_{i,j} c_{ij} \|f(p_j) - f(p_i)\|^2 \quad (6.1)$$

where w_{ij} are the (symmetric) cotangent weights [9] defined on triangle edges going from p_i to p_j . The discrete Laplacian operator can be identified in that energy as:

$$\Delta f = \sum_{j \in N_i} c_{ij}(f(p_j) - f(p_i)) \quad (6.2)$$

where N_i is the set of indices of the neighbors of vertex p_i . It is known that harmonic energy minimizes angular distortions. That means that harmonic functions smoothly blend boundary conditions over the domain. Harmonic functions are intrinsic to surfaces and independent of the discretization used to produce meshes. In spirit similar to [12], we propose the interpolation of quaternions over a mesh using harmonic functions. The harmonic quaternion

interpolation over a surface mesh can be formulated as the solution to the following discrete harmonic equation:

$$\sum_{j \in N(i)} c_{ij} (R_j - R_i) = 0 \quad (6.3)$$

subject to Dirichlet boundary conditions $R_k = R_k^{const}$, where $\{R_k^{const}\}$ are the optimized quaternions from the lower resolution mesh and $\{c_{ij}\}$ are the cotangent weights. This leads to the solution of a sparse linear matrix system. This interpolation produces a smooth field of quaternions by “averaging” the boundary conditions gradually over the surface. This is in effect equivalent to a linear interpolation of boundary conditions across the surface. It can be written in matrix form as:

$$\begin{aligned} L R &= 0 \\ \text{s.t. } R_k &= R_k^{const} \end{aligned} \quad (6.4)$$

where L is discrete Laplacian operator used to solve the global step. Note that interpolated quaternions are not of unit length and must be normalized afterwards. Despite the simplicity of this approach, it works surprisingly well, it is extremely efficient, and it provides a natural propagation at no extra cost.

7. The form of M_i

Notice that the membrane term $\sum_j \|R_i q_{ij} \tilde{R}_i - p_{ij}\|^2$ is equivalent to $\sum_j \|R_i q_{ij} - p_{ij} R_i\|^2$ under the L2 norm. Also notice that $R_i q_{ij} - p_{ij} R_i$ can be rewritten as $(w_i + B_i) q_{ij} - p_{ij} (w_i + B_i)$ for some a scalar w_i and pure quaternion B_i such that $R_i = w_i + B_i$. Expanding the quaternion product in terms of the inner product and the cross product we get:

$$(q_{ij} - p_{ij}) w_i - (q_{ij} - p_{ij}) \cdot B_i - (q_{ij} + p_{ij}) \times B_i \quad (7.1)$$

The expression above can be written in matrix form to get the matrix system $M_{ij} R_i$:

$$\begin{aligned} M_{ij} R_i &= \begin{bmatrix} 0 & -d_{ij}^T \\ d_{ij} & [s_{ij}]_{\times}^T \end{bmatrix} \begin{bmatrix} w_i \\ B_i \end{bmatrix} = \begin{bmatrix} -d_{ij}^T B_i \\ w_i d_{ij} - s_{ij} \times B_i \end{bmatrix} \\ d_{ij} &= q_{ij} - p_{ij} \quad s_{ij} = p_{ij} + q_{ij} \end{aligned} \quad (7.2)$$

where d_{ij} and s_{ij} are 3×1 column vectors holding pure quaternion coefficients, M_{ij} is a skew-symmetric 4×4 real matrix, so that $M_{ij}^T = -M_{ij}$. The quaternion R_i is represented as 4×1 column vector made of the scalar w_i and the 3×1 column vector B_i holding the pure quaternion components. The 3×3 matrix $[s_{ij}]_{\times}$ is representing the skew-symmetric cross-product matrix as usually defined for vectors in \mathbb{R}^3 .

We can express $\sum_j \|R_i q_{ij} - p_{ij} R_i\|^2$ as the quadratic form $R_i^T M_i R_i$ where $M_i = \sum_j^n c_{ij} M_{ij}^T M_{ij}$. Note that since M_{ij} is skew-symmetric, the product $M_{ij}^T M_{ij}$ is symmetric and positive semi-definite. Consequently the matrix

M_i is also symmetric positive semi-definite. It follows that all eigenvalues of M_i are real and $\lambda_i \geq 0$.

$$M_{ij}^T M_{ij} = \begin{bmatrix} \|d_{ij}\|^2 & (s_{ij} \times d_{ij})^T \\ s_{ij} \times d_{ij} & d_{ij} d_{ij}^T - [s_{ij}]_{\times}^2 \end{bmatrix} \quad (7.3)$$

$$d_{ij} = q_{ij} - p_{ij} \quad s_{ij} = p_{ij} + q_{ij}$$

7.1. Efficient Computation of M_i

The symmetric matrix $M_{ij}^T M_{ij}$ has a simple form:

$$M_{ij}^T M_{ij} = \begin{bmatrix} \|d_{ij}\|^2 & (s_{ij} \times d_{ij})^T \\ s_{ij} \times d_{ij} & d_{ij} d_{ij}^T - s_{ij} s_{ij}^T + \|s_{ij}\|^2 I_{3 \times 3} \end{bmatrix}$$

$$d_{ij} = q_{ij} - p_{ij} \quad s_{ij} = p_{ij} + q_{ij}$$

Writing it in terms of p_{ij} and q_{ij} we get:

$$M_{ij}^T M_{ij} = 2 \begin{bmatrix} q_{ij}^T p_{ij} & (q_{ij} \times p_{ij})^T \\ q_{ij} \times p_{ij} & q_{ij} p_{ij}^T + p_{ij} q_{ij}^T - q_{ij}^T p_{ij} I_{3 \times 3} \end{bmatrix} \quad (7.4)$$

$$+ (\|p_{ij}\|^2 + \|q_{ij}\|^2) I_{4 \times 4}$$

All terms of 7.4 can be derived from the dyadic tensor $q_{ij} p_{ij}^T$ plus the quantity $\|p_{ij}\|^2 + \|q_{ij}\|^2$. Since matrix $q_{ij} p_{ij}^T$ is of 3×3 its computation is efficient.

8. Implementation

The author's source code is publicly available at github [7].

9. Results

10. Conclusion

References

- [1] I. Chao, U. Pinkall, P. Sanan, and P. Schröder. A simple geometric model for elastic deformations. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, New York, NY, USA, 2010. Association for Computing Machinery.
- [2] S.-Y. Chen, L. Gao, Y.-K. Lai, and S. Xia. Rigidity controllable as-rigid-as-possible shape deformation. *Graphical Models*, 91:13 – 21, 2017.
- [3] T. Jiang, K. Qian, S. Liu, J. Wang, X. Yang, and J. Zhang. Consistent as-similar-as-possible non-isometric surface registration. *The Visual Computer*, 33:891–901, 2017.
- [4] Y. Le Vaou, J.-C. Léon, S. Hahmann, S. Masfrand, and M. Mika. As-stiff-as-needed surface deformation combining arap energy with an anisotropic material. *Computer-Aided Design*, 121:102803, 2020.
- [5] Z. Levi and C. Gotsman. Smooth rotation enhanced as-rigid-as-possible mesh animation. *IEEE transactions on visualization and computer graphics*, 21(2):264–277, February 2015.

- [6] M. C. Lopez Belon. Applications of conformal geometric algebra in mesh deformation. In *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI - Conference on*, pages 39–46, Aug 2013.
- [7] M. C. Lopez Belon. Project on github. Available at <https://github.com/mauriciocele/arap-sr-linearized>, 2020.
- [8] A. Morsucci, M. Centin, and A. Signoroni. Fast Centroidal Deformation for Large Mesh Models. In M. Livesu, G. Pintore, and A. Signoroni, editors, *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association, 2018.
- [9] U. Pinkall, S. D. Juni, and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2:15–36, 1993.
- [10] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the fifth Eurographics symposium on Geometry processing, SGP '07*, pages 109–116, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [11] J. Wu, M. Lopez, M. Liu, and Y. Zhu. Linear geometric algebra rotor estimator for efficient mesh deformation. *IET Cyber-systems and Robotics*, 2(2):88–95, 2020.
- [12] R. Zayer, C. Rössl, Z. Karni, and H. peter Seidel. Harmonic guidance for surface deformation. In *In Proc. of Eurographics 05*, pages 601–609, 2005.

Mauricio Cele Lopez Belon
Madrid, Spain
e-mail: mclopez@outlook.com