A Singularity-Free Rotation Estimator in Geometric Algebra

Mauricio Cele Lopez Belon

Abstract. Robust methods for finding the best rotation aligning two sets of corresponding vectors are formulated in the linear algebra framework, using tools like the SVD for polar decomposition or QR for finding eigenvectors. Those are well established numerical algorithms which on the other hand are iterative and computationally expensive. Recently, closed form solutions has been proposed in the quaternion's framework, those methods are fast but they have singularities i.e., they completely fail on certain input data. In this paper we propose a singularity-free estimator of the optimal quaternion based on a formulation of the problem in the Geometric Algebra framework. The proposed method's performance is competitive with closed form solutions reported in literature and at the same time is accurate and simple to implement.

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

Keywords. Geometric Algebra, Rotation Estimation, Wahba Problem.

1. Introduction

Rotation estimation problem has been studied for over half a century [15]. The problem looks for the optimal rotation between two sets of corresponding vectors. Many effective methods have been developed to solving the problem [1, 8, 12, 14, 20]. Formulations based on quaternion's framework leads to solve a max-eigenvalue problem while formulations based on the linear algebra framework relies on the Singular Value Decomposition (SVD). In recent years formulations based on geometric algebra framework [4, 13] appeared but they also rely on the linear algebra framework and SVD due to the lack of native numerical algorithms. Closed form solutions for finding the optimal quaternion has been proposed [16, 17, 19, 21] based on analytic formulas for solving the roots of the quartic polynomial associated with eigenvalue problem.

Accuracy and speed of prominent methods have been compared in previous works [5,11,19] where the trade-off between performance and robustness has also been assessed. In particular SVD based methods exhibit the best accuracy but low performance and quaternion based methods are in the other end depending on the implementation. Regarding the later methods, the closed form solutions exhibit the best trade-off so far but they have singularities i.e., they completely fail on certain input data.

In this paper we propose a singularity-free estimator of the best quaternion aligning two sets of corresponding vectors. It is based on maximizing a convex quadratic energy functional using the geometric algebra framework $\mathbb{G}_3$ which allow us to find a robust solution without resorting to linear algebra numerical algorithms. Geometric algebra rotors are isomorphic to quaternions, we find geometric algebra to be a more natural choice for studying this problem since it integrates rotations with a larger set of subspaces inside the Euclidean vector space $\mathbb{R}^3$, where original data is defined. In contrast to previous works we work primarily with bivectos instead of vectors for the sake of mathematical convenience. Due to mathematical duality of vectors and bivectors in $\mathbb{G}_3$ our formulation is also valid for vectors.

2. Geometric Algebra $\mathbb{G}_3$

A geometric algebra $\mathbb{G}_3$ is constructed over a real vector space $\mathbb{R}^3$, with basis vectors $\{e_1, e_2, e_3\}$. The associative geometric product is defined so that the square of any vector is a scalar $aa = a^2 \in \mathbb{R}$. From the vector space $\mathbb{R}^3$, the geometric product generates the geometric algebra $\mathbb{G}_3$ with elements $\{X, R, A...\}$ called multivectors.

For a pair of vectors, a symmetric inner product $a \cdot b = b \cdot a$ and antisymmetric outer product $a \wedge b = -b \wedge a$ can be defined implicitly by the geometric product $ab = a \cdot b + a \wedge b$ and $ba = b \cdot a + b \wedge a$. It is easy to prove that $a \cdot b = \frac{1}{2}(ab + ba)$ is scalar, while the quantity $a \wedge b = \frac{1}{2}(ab - ba)$, called a bivector or 2-vector, is a new algebraic entity that can be visualized as the two-dimensional analogue of a direction, that is, a planar direction. Similar to vectors, bivectors can be decomposed in a bivector basis $\{e_{12}, e_{13}, e_{23}\}$ where $e_{ij} = e_i \wedge e_j$.

The outer product of three vectors $a \wedge b \wedge c$ generates a 3-vector also known as the pseudoscalar, because the trivector basis consist of single element $e_{123} = e_1 \wedge e_2 \wedge e_3$. Similarly, the scalars are regarded as 0-vectors whose basis is the number 1. It follows that the outer product of k-vectors is the completely antisymmetric part of their geometric product: $a_1 \wedge a_2 \wedge ... \wedge a_k = \langle a_1 a_2 ... a_k \rangle_k$ where the angle bracket means k-vector part, and k is its grade. The term grade is used to refer to the number of vectors in any exterior product. This product vanishes if and only if the vectors are linearly dependent. Consequently, the maximal grade for nonzero k-vectors is 3. It follows that every multivector X can be expanded into its k-vector parts and

the entire algebra can be decomposed into k-vector subspaces:

$$\mathbb{G}_3 = \sum_{k=0}^{n} \mathbb{G}_3^k = \{X = \sum_{k=0}^{n} \langle X \rangle_k\}$$

This is called a *grading* of the algebra.

Reversing the order of multiplication is called reversion, as expressed by $(a_1 a_2 ... a_k)\tilde{} = a_k ... a_2 a_1$ and $(a_1 \wedge a_2 \wedge ... \wedge a_k)\tilde{} = a_k \wedge ... \wedge a_2 \wedge a_1$, and the reverse of an arbitrary multivector is defined by $\tilde{X} = \sum_{k=0}^{n} \langle \tilde{X} \rangle_k$.

Rotations are even grade multivectors known as rotors. We denote the subalgebra of rotors as $\mathbb{G}_3^+$. A rotor R can be generated as the geometric product of an even number of vectors. A reflection of any k-vector X in a plane with normal n is expressed as the sandwitch product $(-1)^k nXn$. The most basic rotor R is defined as the product of two unit vectors a and b with angle of $\frac{\theta}{2}$. The rotation plane is the bivector $B = \frac{a \wedge b}{\|a \wedge b\|}$.

$$ab = a \cdot b + a \wedge b = \cos\left(\frac{\theta}{2}\right) + B \sin\left(\frac{\theta}{2}\right). \tag{2.1}$$

Rotors act on all k-vectors using the sandwitch product $X' = RX\tilde{R}$, where $\tilde{R}$ is the reverse of R and can be obtained by reversing the order of all the products of vectors.

We define the commutator product of two bivectors p_j and q_j as $p_j \times q_j = \frac{1}{2}(p_j q_j - q_j p_j)$. The commutator product of bivectors in $\mathbb{G}_3$ can be interpreted as a cross-product of bivectors in the sense that the resulting bivector $B = p_j \times q_j$ is orthogonal to both p_j and q_j. The commutator product allow us to define the geometric product of two bivectors as $AB = A \cdot B + A \times B$. The inner product of bivectors differs from the inner product of vectors on the sign, since the square of bivectors is negative, the inner product of bivectors is a negative scalar e.g., $(ae_{12} + be_{13} + ce_{23}) \cdot (de_{12} + ee_{13} + fe_{23}) = -ad - be - cf$.

For a pair of unit bivectors, A and B, the rotor R aligning them can be defined as $R = A\frac{(A+B)}{\|A+B\|}$. Defining the unit bivector $C = \frac{A+B}{\|A+B\|}$ the rotor R is also $R = AC = w + L$ where $w = A \cdot C = \cos(\theta/2)$ and $L = A \times C = \sin(\theta/2)\frac{A \times B}{\|A \times B\|}$.

3. Geometric Algebra Rotor Estimation

Given two sets of n corresponding bivectors $P = \{p_j\}_{j=1}^n$ and $Q = \{q_j\}_{j=1}^n$, we attempt to maximize the following energy function:

$$E(R) = \max_{R \in \mathbb{G}_3^+} \sum_j c_j \|p_j + \tilde{R} q_i R\|^2 \tag{3.1}$$

$$s.t.\ R\tilde{R} = 1$$

where $\{c_j\}_{j=1}^n$ are scalar weights such that $\sum_j^n c_j = 1$. It is a quadratic maximization problem with a non-linear constraint in R. Notice that the term $\|p_j + \tilde{R} q_i R\|^2$ is dual to the traditional least squares error $\|q_j - R p_j \tilde{R}\|^2$. Duality in the sense that the optimal R is a critical point of both energies.

Notice also that $p_j + \tilde{R}q_iR$ is equivalent to $Rp_j + q_jR$ by multiplying by R on the left and using the fact that $R\tilde{R} = 1$. The equivalent problem is:

$$E(R) = \max_{R\in\mathbb{G}_3^+} \sum_j c_j \|Rp_j + q_jR\|^2 \tag{3.2}$$
$$s.t.\ R\tilde{R} = 1$$

which exposes the that R is only quadratic in 3.2.

The constraint $Rp_j + q_jR$ can be rewriten as $(w + L)p_j + q_j(w + L)$. For some a scalar w and bivector L. Expanding the geometric product of bivectors in terms of the inner product and the communtator product we get:

$$w(p_j + q_j) + L \cdot (p_j + q_j) + (q_j - p_j) \times L \tag{3.3}$$

In matrix language we can define the following matrix system $M_jR = 0$:

$$M_jR = \begin{bmatrix} 0 & -s_j^T \\ s_j & [d_j]_\times \end{bmatrix} \begin{bmatrix} w \\ L \end{bmatrix} = \begin{bmatrix} -s_j^T L \\ ws_j + d_j \times L \end{bmatrix} \tag{3.4}$$
$$d_j = q_j - p_j \quad s_j = p_j + q_j$$

where d_j and s_j are 3×1 column vectors holding bivector's coefficients, M_j is a skew-symmetric 4×4 real matrix, so that $M_j^T = -M_j$. The rotor R is represented as 4×1 column vector made of the scalar w and the 3×1 column vector L holding the bivector's components. The 3×3 matrix $[d_j]_\times$ is representing the skew-symmetric cross-product matrix as usually defined for vectors in $\mathbb{R}^3$.

We can express $E(R)$ as the following quadratic form:

$$E(R) = \max_R R^T MR \tag{3.5}$$
$$s.t.\ R^T R = 1$$

where $M = \sum_j^n c_j M_j^T M_j$. Note that since M_j is skew-symmetric, the product $M_j^T M_j$ is symmetric and positive semi-definite. Consequently the matrix M is also symmetric positive semi-definite. It follows that all eigenvalues of M are real and $\lambda_i \geq 0$.

$$M_j^T M_j = \begin{bmatrix} \|s_j\|^2 & (s_j \times d_j)^T \\ s_j \times d_j & s_j s_j^T - [d_j]_\times^2 \end{bmatrix} \tag{3.6}$$
$$d_j = q_j - p_j \quad s_j = p_j + q_j$$

By the spectral theorem the maximizer of $E(R)$ is the eigenvector of M associated with the largest eigenvalue which is a positive number.

4. Convexity

The convexity of the energy $E(R)$ can be proof by showing that its Hessian matrix of second partial derivatives is positive semi-definite. The Hessian matrix of $E(R)$ is $\frac{\partial^2 E(R)}{\partial R} = \sum_j^n c_j M_j^T M_j$ which is symmetric, moreover

since M_j is skew-symmetric matrix, the product $M_j^T M_j$ is symmetric positive semi-definite. Then follows that $\sum_j^n c_j M_j^T M_j$ is positive semi-definite and therefore convex, provided that the sum of weights is convex i.e., $\sum_j^n c_j = 1$.

5. Optimal Quaternion using 4D Geometric Algebra

The most time consuming task of the estimation is to compute the eigenvector of M associated with the greatest eigenvalue. In this section we show how to find the largest eigenvalue and its corresponding eigenvector i.e., the required quaternion, using the 4D Geometric Algebra $\mathbb{G}_4$ which is robust, efficient and accurate.

Let us define four vectors $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$, $\mathbf{m}_4$ corresponding to the columns of matrix M:

$$M = \sum_j^n c_j M_j^T M_j = \begin{bmatrix} \mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3 & \mathbf{m}_4 \end{bmatrix} \tag{5.1}$$

The matrix system that we want to solve is $MR = \lambda R$ for λ corresponding to the largest eigenvalue of M. We can write the system in its homogeneous form:

$$\begin{bmatrix} \mathbf{m}_1 - \lambda e_1 & \mathbf{m}_2 - \lambda e_2 & \mathbf{m}_3 - \lambda e_3 & \mathbf{m}_4 - \lambda e_4 \end{bmatrix} \begin{bmatrix} w \\ L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.2}$$

The matrix in equation 5.2, called *characteristic matrix*, is of rank 3 and thus singular i.e., the vectors $(\mathbf{m}_1 - \lambda e_1)$, $(\mathbf{m}_2 - \lambda e_2)$, $(\mathbf{m}_3 - \lambda e_3)$ and $()\mathbf{m}_4 - \lambda e_4)$ are linearly dependent. It follows that its outer product must be zero:

$$(\mathbf{m}_1 - \lambda e_1) \wedge (\mathbf{m}_2 - \lambda e_2) \wedge (\mathbf{m}_3 - \lambda e_3) \wedge (\mathbf{m}_4 - \lambda e_4) = 0 \tag{5.3}$$

Which is called the *charactetistic outer porduct* and is equivalent to the characteristic polynomial $P(\lambda) = det(M - \lambda I)$. A simple way to find the largest eigenvalue is using Newton-Raphson method $\lambda_{i+1} = \lambda_i - P(\lambda)/P'(\lambda)$. This method is indeed robust given that all eigenvalues are non-negative real numbers, it wasn't however for methods based on Davenport's matrix [2] which eigenvalues can be negative. We found that $Trace(M)$ is a robust guess for Newton-Raphson iteration because it is larger than the largest eigenvalue and is also close enough to converge in few iterations. For the sake of completeness we show the first derivative of 5.3:

$$\begin{aligned} P'(\lambda) = &-e_1 \wedge (\mathbf{m}_2 - \lambda e_2) \wedge (\mathbf{m}_3 - \lambda e_3) \wedge (\mathbf{m}_4 - \lambda e_4) \\ &-(\mathbf{m}_1 - \lambda e_1) \wedge e_2 \wedge (\mathbf{m}_3 - \lambda e_3) \wedge (\mathbf{m}_4 - \lambda e_4) \\ &-(\mathbf{m}_1 - \lambda e_1) \wedge (\mathbf{m}_2 - \lambda e_2) \wedge e_3 \wedge (\mathbf{m}_4 - \lambda e_4) \\ &-(\mathbf{m}_1 - \lambda e_1) \wedge (\mathbf{m}_2 - \lambda e_2) \wedge (\mathbf{m}_3 - \lambda e_3) \wedge e_4 \end{aligned}$$

Following [3] we solve the system 5.2 using outer products. For a linear system $A\mathbf{x} = b$ the authors in [3] defines N linear equations of the form $A_j^T\mathbf{x} = b_j$ each of which corresponds to a dual hyper-plane of the form $\mathbf{a}_j \equiv A_j - b_j e_0$ where the solution $\mathbf{x}$ must lie on. The e_0 is an *homogeneous* basis vector needed for enabling projective geometry, enlarging the base space to $N+1$, which interpretation is to be the offset of the hyper-plane. So solution of the linear system is the intersection of N dual hyper-planes, which is given by its outer product.

$$\alpha(\mathbf{x} + e_0)^* = \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge ... \wedge \mathbf{a}_N \tag{5.4}$$

Where the symbol * is the dual operator of the $N+1$ space and α is a weight factor. After taking the dual of $\alpha(\mathbf{x} + e_0)^*$ and divide by the coefficient of e_0 (which is α), the solution $\mathbf{x}$ can be read off the coefficients of the 1-vector.

We know that the null space of $(M - \lambda I)$ is of rank one. Algebraically this means that one of its column vectors is redundant i.e., it can be written in term of the others. In linear algebra this means that the system has infinitely many solutions. The geometric interpretation is that all hyper-planes intersect in a line passing through the origin. Since all solutions lie on the same line and they only differ by a scaling term, a particular solution can be found by fixing the scale. Actually, it is enough to constrain the scale of a single hyper-plane. The homogeneous component e_0 can be interpreted as scale of solution (instead of hyper-plane's offset as in [3]) since it affects only that aspect of the solution. In linear algebra language it is equivalent to set one value at the right hand side of 5.2, however that system can't be solved in linear algebra because it requires to invert a singular matrix.

We define the dual hyper-planes passing through the origin as:

$$\mathbf{a}_i \equiv \mathbf{m}_i - \lambda e_i \tag{5.5}$$

Although in most cases it is enough to set the scale of a single hyper-plane go get a solution it is inconvenient to do so, as will be explained in Section 6. It is more robust to set the scale of all hyper-planes to some $\gamma \neq 0$ which is a scalar value. Intersection can then be found by taking the outer product as:

$$\alpha(\mathbf{x} + 1)^* = (\mathbf{a}_1 + \gamma) \wedge (\mathbf{a}_2 + \gamma) \wedge (\mathbf{a}_3 + \gamma) \wedge (\mathbf{a}_4 + \gamma) \tag{5.6}$$

Distributing the outer product and keeping the terms of grade-3 we get:

$$\alpha(\mathbf{x} + 1)^* = \gamma(\mathbf{a}_1 \wedge \mathbf{a}_3 \wedge \mathbf{a}_4 + \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \mathbf{a}_4 + \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \mathbf{a}_3 + \mathbf{a}_2 \wedge \mathbf{a}_3 \wedge \mathbf{a}_4) \tag{5.7}$$

Here the symbol * is the dual operator of the 4D space (not 5D as in [3] which allow us to be more efficient) and α is a weight factor. After taking the dual of $\alpha(\mathbf{x} + 1)^*$ the eigenvector $\mathbf{x}$ can be read off the coefficients of the 1-vector. Notice that solution needs to be normalized.

6. Robustness and Singularities

As stated before, setting the scale of a single hyperplane is enough to get a valid eigenvector in most cases. However, the choice of which hyper-plane to constraint is problematic. For instance, assuming input vectors without noise, constraining the hyperplane corresponding to the w component of the rotor won't work because when the angle of rotation is π the $w = \cos(\pi/2) = 0$ and so its scale cannot be constrained. Similarly, constraining one of the hyper-planes corresponding to L_1, L_2 or L_3 won't work because when the angle of rotation is 0 or 4π the $\sin(2\pi) = 0$ and so on. Then, it is complex task to avoid all problematic situations. By constraining all hyper-planes as in 5.6 our geometric algebra method does not suffer from any of those singularities.

Some of the above mentioned singularities are present in classic methods such as QUEST [14] and FOAM [10] but also on methods derived from those, including recent descendants based on analytic formulas [16–19,21]. All those methods are based on finding the eigenvector corresponding to largest eigenvalue of Davenport's matrix [2]. Since that is an indefinite matrix, some eigenvalues are positive and some negative, the Newton-Raphson can fail to find the max eigenvalue (which can be negative). So significant effort has been put on finding fast and robust analytic solutions to the quartic polynomial but no advances has been made on improving robustness on finding the associated eigen-vector.

7. Optimal Computation of M

The symmetric matrix $M_j^T M_j$ has a simple form:

$$M_j^T M_j = \begin{bmatrix} \|s_j\|^2 & (s_j \times d_j)^T \\ s_j \times d_j & s_j s_j^T - d_j d_j^T + \|d_j\|^2 I \end{bmatrix}$$

$$d_j = q_j - p_j \quad s_j = p_j + q_j$$

Writing it in terms of p_j and q_j we get:

$$M_j^T M_j = 2 \begin{bmatrix} p_j^T q_j & (p_j \times q_j)^T \\ p_j \times q_j & p_j q_j^T + q_j p_j^T - p_j^T q_j I_{3\times 3} \end{bmatrix} + (\|p_j\|^2 + \|q_j\|^2) I_{4\times 4} \quad (7.1)$$

All terms of 7.1 can be derived from the covariance matrix $B = p_j q_j^T$ plus the quantity $\|p_j\|^2 + \|q_j\|^2$. Since matrix B is of 3×3 its computation is more efficient than the whole 7.1. Details can be found in Section 8.

8. Algorithms

The pseudo-code of proposed method is shown in Algorithm 1.

An optimized C++ code using the Eigen library [6] and GAALOP [7] is publicly available on GitHub [9]

Algorithm 1 Fast Rotor Estimation

Require: $P = \{p_j\}_{j=1}^{n}, Q = \{q_j\}_{j=1}^{n}, C = \{c_j\}_{j=1}^{n}$
$S = B = 0, \gamma = 1$
for $j = 1$ **to** n **do**
  $S = S + c_j(p_j \cdot p_j + q_j \cdot q_j)$
  $B = B + c_j p_j q_j^T$
end for
$\mathbf{m}_1 = (S + Tr(B))e_1 + (B_{12} - B_{21})e_2 + (B_{20} - B_{02})e_3 + (B_{01} - B_{10})e_4$
$\mathbf{m}_2 = (B_{12} - B_{21})e_1 + 2(B_{00} + S - Tr(B))e_2 + (B_{01} + B_{10})e_3 + (B_{20} + B_{02})e_4$
$\mathbf{m}_3 = (B_{20} - B_{02})e_1 + (B_{01} + B_{10})e_2 + 2(B_{11} + S - Tr(B))e_3 + (B_{12} + B_{21})e_4$
$\mathbf{m}_3 = (B_{01} - B_{10})e_1 + (B_{20} + B_{02})e_2 + (B_{12} + B_{21})e_3 + 2(B_{22} + S - Tr(B))e_4$
{Newton-Raphson}
$\lambda_0 = 7S - 3Tr(B)$
repeat
  $\lambda_{i+1} = \lambda_i - P(\lambda_i)/P'(\lambda_i)$
until $\|\lambda_{i+1} - \lambda_i\| < \epsilon$
$\mathbf{a}_1 = \mathbf{m}_1 - \lambda e_1$
$\mathbf{a}_2 = \mathbf{m}_2 - \lambda e_2$
$\mathbf{a}_3 = \mathbf{m}_3 - \lambda e_3$
$\mathbf{a}_4 = \mathbf{m}_4 - \lambda e_4$
$\mathbf{X} = \gamma(\mathbf{a}_1 \wedge \mathbf{a}_3 \wedge \mathbf{a}_4 + \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \mathbf{a}_4 + \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \mathbf{a}_3 + \mathbf{a}_2 \wedge \mathbf{a}_3 \wedge \mathbf{a}_4)$
$R = normalize(\langle \mathbf{X}^* \rangle_1)$
return $R(0) + R(1)e_{12} + R(2)e_{13} + R(3)e_{23}$

9. Comparisons

We selected several representative methods e.g. FLAE [19], SVD [1] and QUEST [14] for comparison. The Eigen library [6] was employed to implement all methods. The tests were ran on a MacBook Pro laptop with Intel Core i7 CPU running at 2,5 GHz. The Clang C++ compiler was used with -Ofast option enabled. Results are listed below:

10. Conclusion

In this paper, we presented a novel method for estimating the best rotation aligning two sets of corresponding 3D bivectors and vectors. It is based on maximizing a quadratic energy functional formulated in geometric algebra language. Our method is fast, robust and accurate. Experimental validation of the proposed algorithm was presented. Results shows that our method present increased robustness while keeping accuracy and competitive speed.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, Sep. 1987.

[2] P. B. Davenport. A vector approach to the algebra of rotations with applications. 1968.

[3] S. De Keninck and L. Dorst. Geometric algebra levenberg-marquardt. In M. Gavrilova, J. Chang, N. M. Thalmann, E. Hitzer, and H. Ishikawa, editors, *Advances in Computer Graphics*, pages 511–522, Cham, 2019. Springer International Publishing.

[4] L. Dorst and J. Lasenby, editors. *Guide to Geometric Algebra in Practice*. Springer, 2011.

[5] D. Eggert, A. Lorusso, and R. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5):272–290, Mar 1997.

[6] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[7] D. Hildenbrand, P. Charrier, C. Steinmetz, and J. Pitt. Gaalop home page. Available at http://www.gaalop.de, 2014.

[8] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[9] M. C. Lopez Belon. Project on github. Available at https://github.com/mauriciocele/fast-rotor-estimation, 2019.

[10] L. Markley. Attitude determination from vector observations: A fast optimal matrix algorithm. *Journal of the Astronautical Sciences*, 41, 07 1993.

[11] L. Markley. 30 years of wahba's problem. 02 1999.

[12] D. Mortari. EULER-q algorithm for attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, 21(2):328–334, 1998.

[13] C. B. U. Perwass. *Geometric algebra with applications in engineering*, volume 4 of *Geometry and Computing*. Springer, Berlin; Heidelberg, 2009.

[14] M. D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, 4(1):70–77, 1981.

[15] G. Wahba. A Least Squares Estimate of Satellite Attitude, 1965.

[16] J. Wu, M. Liu, Z. Zhou, and R. Li. Fast rigid 3d registration solution: A simple method free of svd and eigen-decomposition, 2018.

[17] J. Wu, M. Liu, Z. Zhou, and R. Li. Fast symbolic 3d registration solution, 2018.

[18] J. Wu, Z. Zhou, J. Chen, H. Fourati, and R. Li. Fast Complementary Filter for Attitude Estimation Using Low-Cost MARG Sensors. *IEEE Sensors Journal*, 16(18):6997–7007, 2016.

[19] J. Wu, Z. Zhou, B. Gao, R. Li, Y. Cheng, and H. Fourati. Fast Linear Quaternion Attitude Estimator Using Vector Observations. *IEEE Transactions on Automation Science and Engineering*, X(X):1–13, 2017.

[20] Y. Yang. Attitude determination using Newton's method on Riemannian manifold. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 229(14):2737–2742, 2015.

[21] Y. Yang and Z. Zhou. An analytic solution to wahba's problem. *Aerospace Science and Technology*, 30(1):46–49, Oct 2013.

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