

ORIGINAL ARTICLE

Open Access

# Efficient object location determination and error analysis based on barycentric coordinates



Andrea Bodonyi<sup>1,2</sup> and Roland Kunkli<sup>1\*</sup>

## Abstract

In this paper, we propose an efficient computational method for converting local coordinates to world coordinates using specially structured coordinate data. The problem in question is the computation of world coordinates of an object throughout a motion, assuming that we only know the changing coordinates of some fixed surrounding reference points in the local coordinate system of the object. The proposed method is based on barycentric coordinates; by taking the aforementioned static positions as the vertices of a polyhedron, we can specify the coordinates of the object in each step with the help of barycentric coordinates. This approach can significantly help us to achieve more accurate results than by using other possible methods. In the paper, we describe the problem and barycentric coordinate-based solution in detail. We then compare the barycentric method with a technique based on transformation matrices, which we also tested for solving our problem. We also present various diagrams that demonstrate the efficiency of our proposed approach in terms of precision and performance.

**Keywords:** Barycentric coordinates, Conversion, Transformation, Coordinate systems, Location

## Introduction

Conversion between coordinate systems and applying coordinate transformations are common techniques for solving different scientific problems. This general method is efficiently used in various fields, such as computer graphics, image processing [1], geometric processing [2], physics [3], and geoinformatics [4, 5]. In this paper, we propose a method for dealing with a similar problem, where conducting conversions between different coordinate systems is also necessary.

To illustrate our problem, let us assume we have a motion sequence described by each of its steps. What we are interested in is the global position of the moving object in each step. Object determination is a common problem in different scientific fields as well; however, there may be cases where the available information is limited or special in some way [6–9]. In our case, we

assume that the given data is special in that only a few reference points (and their changing coordinates), which are located in the surroundings of the moving object and defined in its local coordinate system, are known. To obtain the desired object positioning, we must determine the coordinates of the main object for every step by converting from the object's local coordinate system to the global one using the information at our disposal. There is also a need for analyzing the provided precision, which is essential in different object locating applications [10, 11].

To this end, we propose a computational method that is capable of obtaining the required motion data using the information outlined above. The essence of the method lies in the use of barycentric coordinates related to the surrounding reference points. Barycentric coordinates are widely used in the field of computer graphics. They have many applications, including interpolation and deformation [12–14], character articulation [15], and mesh parameterization [16, 17]. Their utility has been proved in other fields besides computer graphics,

\* Correspondence: [kunkli.roland@inf.unideb.hu](mailto:kunkli.roland@inf.unideb.hu)

<sup>1</sup>Faculty of Informatics, University of Debrecen, Debrecen H-4028, Hungary  
Full list of author information is available at the end of the article

such as sensor networks [8] and robot localization, as a part of the trilateration method [6, 7].

In the next section, we present the problem in detail by describing the data concept. Then, we continue with a brief overview of barycentric coordinates along with a thorough demonstration of our proposed method. After that, we evaluate our method by comparing it with other possible methods, and the same section shows the corresponding error metrics for their precision and performance as well. Lastly, the paper is concluded with a summary of our results.

## Methods

### Problem definition

Our main problem is given by the concept of data creation and what kind of information it provides us. As we mentioned above, it contains a motion series that is described by the environment of the moving object. This supposition implies the task of locating this object in the global coordinate system based on its environment.

In the data sequence, each step contains information that is described only by itself. At every step of the motion, we only know the changed coordinates of the surrounding reference points with respect to the local coordinate system of the main object. Nevertheless, our goal is to compute the (changing) world coordinates of the main object, assuming that the reference points remain fixed in the global system. We also know that the main object starts its motion at the origin of the global coordinate system. Of course, if we consider its local coordinates, the object plays the role of the origin in every step.

In other words, the motion is defined by some base points in the surroundings of the object. Knowing that these reference points are stationary, we can rely on them to retrieve the position of our observed object.

The structure of the data for a motion step in the sequence is built up as follows: it contains a step identifier and an object array that holds four objects. For each object, the array stores the corresponding relative position in the current step. There is no additional information contained in the data.

### Barycentric approach

#### *Barycentric coordinates*

Barycentric coordinates have the capability of giving the position of an arbitrary point based on some reference points. We can get this mentioned position by taking an adequately weighted sum of the said references. In the following, we present an example case for this in the plane.

Based on ref. [18], we would like to highlight the main characteristics of barycentric coordinates in the three-dimensional case.

Let us consider an arbitrary given tetrahedron with vertices  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$ , and  $\mathbf{d}$ . Any point on the edge between  $\mathbf{a}$  and  $\mathbf{b}$  can be obtained as  $(1 - k)\mathbf{a} + k\mathbf{b}$ , where  $0 \leq k \leq 1$ . Let us assume that point  $\mathbf{r}$  is considered in the mentioned form.

Then—employed the previously used technique—any point  $\mathbf{q}$  of the segment of  $\mathbf{r}$  and  $\mathbf{c}$  can be determined as  $(1 - l)\mathbf{r} + l\mathbf{c}$ , where  $0 \leq l \leq 1$ .

Similarly, any point  $\mathbf{p}$  of the segment of  $\mathbf{q}$  and  $\mathbf{d}$  can be calculated as  $(1 - m)\mathbf{q} + m\mathbf{d}$ , where  $0 \leq m \leq 1$ . Now, let us express  $\mathbf{p}$  with  $k$ ,  $l$ ,  $m$ , and the points of the tetrahedron. From the previously mentioned equations, we get

$$\begin{aligned} \mathbf{p} &= (1 - m)\mathbf{q} + m\mathbf{d} \\ &= (1 - m)((1 - l)\mathbf{r} + l\mathbf{c}) + m\mathbf{d} \\ &= (1 - m)(1 - l)\mathbf{r} + (1 - m)l\mathbf{c} + m\mathbf{d} \\ &= (1 - m)(1 - l)\mathbf{r} + (l - lm)\mathbf{c} + m\mathbf{d} \\ &= (1 - m - l + lm)((1 - k)\mathbf{a} + k\mathbf{b}) + (l - lm)\mathbf{c} + m\mathbf{d} \\ &= (1 - m - l + lm - k + km + kl - klm)\mathbf{a} \\ &\quad + (k - km - kl + klm)\mathbf{b} + (l - lm)\mathbf{c} + m\mathbf{d} \end{aligned}$$

Let us introduce the following notations:

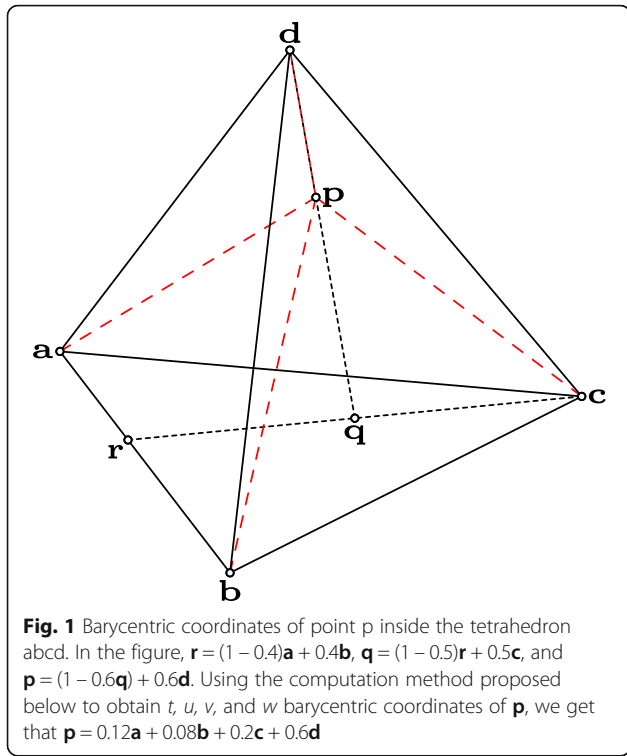
$$\begin{aligned} t &:= 1 - k - l - m + lm + km + kl - klm, \\ u &:= k - km - kl + klm, \\ v &:= l - lm, \\ w &:= m. \end{aligned}$$

Using the notations introduced above,  $\mathbf{p}$  can be written as  $\mathbf{p} = t\mathbf{a} + u\mathbf{b} + v\mathbf{c} + w\mathbf{d}$ , and it is easy to see that  $t + u + v + w = 1$ . The  $t$ ,  $u$ ,  $v$ , and  $w$  numbers are called the barycentric coordinates of point  $\mathbf{p}$ , and by using them, the point can be calculated as a weighted sum of  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$ , and  $\mathbf{d}$ . As an example, see Fig. 1. With the proposed calculation based on the intervals between 0 and 1, we can describe only the inner part of the tetrahedron, but this idea can be generalized to any positions in the three-dimensional space.

### *Determining global coordinates using barycentric coordinates*

As we pointed out earlier, our computation is based on the use of barycentric coordinates. To use the barycentric approach as a solution to our problem, it is essential to be able to calculate the barycentric coordinates of our moving object with respect to the four fixed reference positions—which surround our moving one—in its local coordinate system.

First, we introduce some notations. Let us assume that  $x_{i,k}$ ,  $y_{i,k}$ , and  $z_{i,k}$  ( $i \in \{1, 2, 3, 4\}$ ) are the  $x$ ,  $y$ , and  $z$  coordinates, respectively, of the  $i$ -th surrounding reference point in the  $k$ -th step. Furthermore,  $w_{i,k}$  ( $i \in \{1, 2, 3, 4\}$ ) will denote the barycentric coordinates of the main object also in the  $k$ -th step regarding this reference system.



We also know that in this part of our computation, everything is described in the local coordinate system of the main object. Thus, the main object is always located at the origin and has  $(0, 0, 0)$  coordinates.

Based on the method suggested on page 46 in ref. [19] for the two-dimensional case, we can reach the desired barycentric coordinates as follows.

Considering that  $\sum_{i=1}^4 w_{i,k} = 1$ ,  $w_{4,k}$  can be expressed using the other weights as

$$w_{4,k} = 1 - \sum_{i=1}^3 w_{i,k} \tag{1}$$

Using this—and also the fact that the observed object’s local coordinates are  $(0, 0, 0)$  in every step—we can get the following system of equations for the  $k$ -th step:

$$\begin{aligned} \sum_{i=1}^3 w_{i,k} \cdot x_{i,k} + (1 - \sum_{i=1}^3 w_{i,k}) \cdot x_{4,k} &= 0 \\ \sum_{i=1}^3 w_{i,k} \cdot y_{i,k} + (1 - \sum_{i=1}^3 w_{i,k}) \cdot y_{4,k} &= 0 \\ \sum_{i=1}^3 w_{i,k} \cdot z_{i,k} + (1 - \sum_{i=1}^3 w_{i,k}) \cdot z_{4,k} &= 0 \end{aligned} \tag{2}$$

The equation system (2) trivially leads us to the following:

$$\begin{aligned} \sum_{i=1}^3 w_{i,k} \cdot (x_{i,k} - x_{4,k}) &= -x_{4,k} \\ \sum_{i=1}^3 w_{i,k} \cdot (y_{i,k} - y_{4,k}) &= -y_{4,k} \\ \sum_{i=1}^3 w_{i,k} \cdot (z_{i,k} - z_{4,k}) &= -z_{4,k} \end{aligned} \tag{3}$$

which can also be written in matrix form as

$$A_k \cdot \mathbf{w}_k = \mathbf{b}_k \tag{4}$$

where

$$A_k := \begin{pmatrix} x_{1,k} - x_{4,k} & x_{2,k} - x_{4,k} & x_{3,k} - x_{4,k} \\ y_{1,k} - y_{4,k} & y_{2,k} - y_{4,k} & y_{3,k} - y_{4,k} \\ z_{1,k} - z_{4,k} & z_{2,k} - z_{4,k} & z_{3,k} - z_{4,k} \end{pmatrix}$$

$$\mathbf{w}_k := \begin{pmatrix} w_{1,k} \\ w_{2,k} \\ w_{3,k} \end{pmatrix}, \mathbf{b}_k := - \begin{pmatrix} x_{4,k} \\ y_{4,k} \\ z_{4,k} \end{pmatrix}$$

From the matrix form above, we can compute the  $w_{i,k}$  weights as

$$\mathbf{w}_k = A_k^{-1} \cdot \mathbf{b}_k \tag{5}$$

Based on Eq. (1), we can easily calculate  $w_{4,k}$  with vector operations as

$$w_{4,k} = 1 - \langle \mathbf{w}_k, \mathbf{1} \rangle \tag{6}$$

where  $\langle \cdot \rangle$  denotes the dot product, and  $\mathbf{1}$  is the  $(1 \ 1 \ 1)^T$  vector.

So far, we have only found the barycentric coordinates of  $\mathbf{0}$  in the local coordinate system of the actual step. Now, we apply these barycentric coordinates to obtain the position of the moving object for the current step in the global coordinate system.

To get the global coordinates of our object, we need to form a static reference basis in the global coordinate system. We form this basis by the position vectors of the four references from the very first step. Considering them as the four vertices of a tetrahedron, they can be used with the barycentric coordinates derived in the actual step to compute the world coordinates of the searched position.

We assumed that the moving object starts its path at the origin of the global coordinate system. Therefore, at the start of the motion, the coordinates of the four reference points are the same as in the local one. Using this fact, we can consider the calculated barycentric coordinates from every step and use them as weights for the starting reference positions.

Let us collect the barycentric coordinates from the  $k$ -th step into vector  $\hat{\mathbf{w}}_k$ , i.e.,

$$\hat{\mathbf{w}}_k := (w_{1,k} \ w_{2,k} \ w_{3,k} \ w_{4,k})^T$$

Let us assume that  $G_i$  is the matrix containing the local coordinates of the reference positions in the following form:

$$G_i := \begin{pmatrix} x_{1,i} & x_{2,i} & x_{3,i} & x_{4,i} \\ y_{1,i} & y_{2,i} & y_{3,i} & y_{4,i} \\ z_{1,i} & z_{2,i} & z_{3,i} & z_{4,i} \end{pmatrix} \quad (7)$$

If we use this notation for the start of the movement, the global coordinates of the reference positions can be collected in  $G_1$ , and they are the same as their local coordinates at the beginning.

Now we have all necessary data to obtain the desired solution—the global coordinates in the actual step—as the weighted sum of the reference positions. If these coordinates are denoted by  $x_k$ ,  $y_k$ , and  $z_k$ , they can be calculated as follows:

$$\begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} = G_1 \cdot \hat{\mathbf{w}}_k \quad (8)$$

### Transformation matrix-based alternative

When trying to determine a suitable way to retrieve the required coordinates, we ran some tests with a few other methods. However, the huge number of motion steps resulted in computational errors with increasing tendencies throughout the entire simulation in most cases.

The most promising basic method could be using matrix transformations to get the proper coordinates. In each step, we considered the four reference positions from two consecutive steps from our motion data (the previous and current ones) and calculated the matrix that transforms the first set of positions into the second one. We obtained the final transformation by multiplying the previous matrix (computed using the same algorithm) with the current one and used the result to retrieve the desired object location for the current step.

Using the notation introduced in Eq. (7), we would like to find a matrix for which the following is true:

$$T_{k+1} \cdot G_k = G_{k+1} \quad (9)$$

i.e.,  $T_{k+1}$  is the transformation matrix that transforms the reference positions from the  $k$ -th step to the  $k+1$ -th. Therefore, we can compute the transformation matrices in the following way:

$$T_{k+1} = G_{k+1} \cdot G_k^{-1} \quad (10)$$

Based on this formulation and given our assumption that the object is initially located at the origin, we can obtain the position of the observed object in the  $i$ -th motion step by accumulating all of the transformation matrices until that step. Formally written, the desired position of the object is

$$\left( \prod_{k=1}^i T_k^{-1} \right) \cdot (0 \ 0 \ 0 \ 1)^T \quad (11)$$

Equation (11) can be rearranged to have only one inverse matrix calculation in the following way:

$$\left( \prod_{k=0}^{i-1} T_{i-k} \right)^{-1} \cdot (0 \ 0 \ 0 \ 1)^T \quad (12)$$

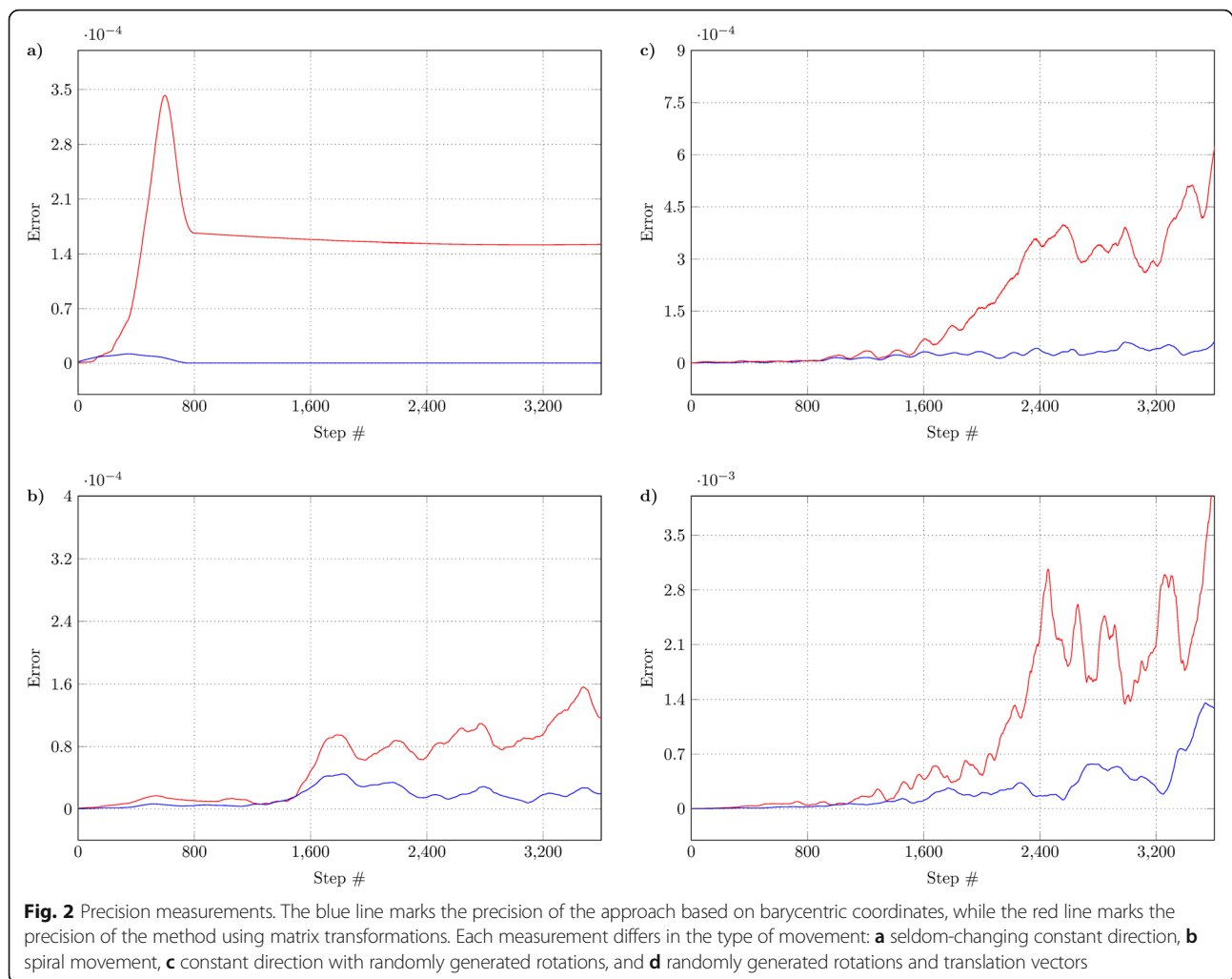
## Results

To have ground-truth reference information to compare our results with and to have a measure for obtaining the accuracy of the two methods, we generated some test data sequences describing different movement types of the object. These synthetic sequences differ in the movement alignment of the object for analyzing the impact of each motion type on the approximation errors.

Here, we present four unique test cases with various translation and rotation configurations. In the first case, the object moves in a seldom-changing direction with no additional rotation. Following this, the moving object follows a spiral path introduced along the  $x$ -axis. For the third case, we further varied the first test case having a nearly constant moving direction. We introduced per-step random rotations along the  $x$  and  $y$  axes. Finally, we kept the random rotations from the previous test case but randomized the translation vectors in each step too. For each test case, we collected the absolute errors of the matrix method and barycentric approach as the distance between the results provided by each method and the ground-truth information. We used these numbers as performance indicators of the individual test approaches. We summarize the resulting error metrics in Fig. 2.

### Precision

As demonstrated by our results, the error produced by the matrix transformation method is several orders of magnitude larger than that of our proposed algorithm. Because barycentric coordinates yield an error of approximately  $10^{-6}$ , it is difficult to visualize the error on the same scale as the matrix transformation error. For this reason, the error of the barycentric method may appear as zero in some places, but the magnitude difference between the two methods is clearly noticeable.



In the case of periodic movement, where the object had no rotation and seldom changed its moving direction, we experienced a nearly constant error amount with a magnitude of  $10^{-4}$  (Fig. 2a). In the case when the object was moving on a spiral path with and without random rotations (Fig. 2b and c), the error was first relatively unified at a magnitude of  $10^{-5}$  with little growth, eventually reaching a magnitude of  $10^{-4}$ . This means that the matrix method produced acceptable results. The next test case, however, shows a different view. In this case, the object moves in randomly generated directions with per-step random rotations (Fig. 2d). Such a complicated scenario resulted in larger-scale errors having a firmly rising tendency, showing a growing precision difference between the barycentric method and matrix transformations.

### Performance

We also conducted performance measurements for the two methods. The results show a clear gap between the barycentric approach and the method

using matrix transformations. We had three runs for the runtime measurement, containing 10000, 100000, and 200000 steps. We measured the times for the process of calculating the position of the main object for all motion steps.

The results of different cases show a rising tendency for the difference between the runtime of the two methods. With 10000 and 200000 steps, the program run gives us a nearly six millisecond and 240 millisecond difference, respectively. In our case, this means more than 200% runtime growth (see the exact values in Table 1). We ran the performance tests on a dual-core Intel Core i5-3230M 2.60GHz processor.

**Table 1** Performance measurements of the barycentric approach and matrix method

Step count	Barycentric approach	Matrix method
10000	14.5 ms	20.7 ms
100000	54.9 ms	85.9 ms
200000	101.4 ms	343.5 ms



## Discussion and future work

In this paper, we proposed a method to obtain the actual position of an object using some reference points from its surroundings. Our computation builds upon barycentric coordinates and uses the available points as references to obtain the movement of the observed object in the global coordinate system.

The strength of the barycentric method lies in the fact that information from earlier motion steps is seldom used, leading to reduced error propagation in the system. On the other hand, as mentioned before, in three dimensions, four reference points are required to retrieve the barycentric coordinates of a point. We had four guaranteed fixed points that we could use as references for the barycentric coordinates, which gave a perfect basis for designing the proposed method.

As the self-localization problem is always dependent on the actual environment, it always requires special solutions that account for the available information or that are measurable with disposable devices. Thomas outlined the problem of robot localization in their work [6], from which a possible use case can be derived. As a robot moves, its sensors may collect coordinate information about the surrounding static objects relative to itself; this results in a data format from which the movement-dependent localization problem can be easily solved using the method proposed in this paper.

Additionally, the suggested barycentric method can also solve the inverse of the problem presented in ref. [20]. If we can find static anchors in the surroundings of a moving pedestrian such that we can determine the positions of the anchors relative to them, one could determine the global position of the pedestrian using our proposed algorithm.

Lastly, the field of scientific visualization is another great use case for the proposed barycentric position determination algorithm. We can mainly imagine the algorithm helping with the visualization and animation of object motion data originating from specialized, domain-specific simulation systems.

Considering the results presented in this paper, we plan to work on the applications of the barycentric method in motion data from simulations of an ongoing biological study. We would also like to examine and resolve special cases, such as possible changes in the current fixed environment. We also plan to evaluate the computations in more detail by applying new special cases and a homogeneous coordinates-based computation approach [21] and making further precision measurements of them.

## Acknowledgments

The authors wish to thank Győző Kurucz and Gábor Holló for highlighting the problem considered in the paper.

## Authors' contributions

RK developed the theoretical background of the proposed method, while both of the authors had contributions in implementing the method; AB implemented and ran the tests for obtaining precision and performance; All authors wrote, read, and approved the final manuscript.

## Funding

This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

## Availability of data and materials

The datasets used in this study are available from the corresponding author on reasonable request.

## Competing interests

The authors declare that they have no competing interests.

## Author details

<sup>1</sup>Faculty of Informatics, University of Debrecen, Debrecen H-4028, Hungary.

<sup>2</sup>Doctoral School of Informatics, University of Debrecen, Debrecen H-4028, Hungary.

Received: 30 October 2019 Accepted: 29 June 2020

Published online: 31 July 2020

## References

1. Davis MH, Khotanzad A, Flamig DP, Harms SE (1997) A physics-based coordinate transformation for 3-D image matching. *IEEE Trans Med Imaging* 16(3):317–328 <https://doi.org/10.1109/42.585766>
2. D'Azevedo EF (1991) Optimal triangular mesh generation by coordinate transformation. *SIAM J Sci Stat Comput* 12(4):755–786 <https://doi.org/10.1137/0912040>
3. Broer LJF, Kobussen JA (1974) Conversion from material to local coordinates as a canonical transformation. *Appl Sci Res* 29(1):419–429 <https://doi.org/10.1007/BF00384163>
4. Zhu J (1994) Conversion of earth-centered earth-fixed coordinates to geodetic coordinates. *IEEE Trans Aerosp Electron Syst* 30(3):957–961 <https://doi.org/10.1109/7.303772>
5. Ligas M (2012) Cartesian to geodetic coordinates conversion on a triaxial ellipsoid. *J Geod* 86(4):249–256 <https://doi.org/10.1007/s00190-011-0514-7>
6. Thomas F, Ros L (2005) Revisiting trilateration for robot localization. *IEEE Trans Robot* 21(1):93–101 <https://doi.org/10.1109/TRO.2004.833793>
7. Manolakis DE (1996) Efficient solution and performance analysis of 3-D position estimation by trilateration. *IEEE Trans Aerosp Electron Syst* 32(4):1239–1248 <https://doi.org/10.1109/7.543845>
8. Diao YF, Lin ZY, Fu MY (2014) A barycentric coordinate based distributed localization algorithm for sensor networks. *IEEE Trans Signal Process* 62(18):4760–4771 <https://doi.org/10.1109/TSP.2014.2339797>
9. Wu YH, Tang FL, Li HP (2018) Image-based camera localization: an overview. *Vis Comput Ind Biomed Art* 1(1):8 <https://doi.org/10.1186/s42492-018-0008-z>
10. Tian W (2007) The GPS coordinates conversion and error analysis in radar precision test. *Radar ECM* 2:54–56
11. Park CS, Cho DJ, Cha EJ, Hwang DH, Lee SJ (2006) Error analysis of 3-dimensional GPS attitude determination system. *Int J Control Autom Syst* 4(4):480–485
12. Weber O, Ben-Chen M, Gotsman C (2009) Complex barycentric coordinates with applications to planar shape deformation. *Comput Graph Forum* 28(2):587–597 <https://doi.org/10.1111/j.1467-8659.2009.01399.x>
13. Zhang JY, Deng BL, Liu ZS, Patané G, Bouaziz S, Hornmann K et al (2007) Local barycentric coordinates. *ACM Trans Graph* 33(6):188 <https://doi.org/10.1145/2661229.2661255>
14. Torsten L (2008) On generalized barycentric coordinates and their applications in geometric modeling. Dissertation, MPI Informatik
15. Joshi P, Meyer M, DeRose T, Green B, Sanocki T (2014) Harmonic coordinates for character articulation. *ACM Trans Graph* 26(3):71 <https://doi.org/10.1145/1276377.1276466>
16. Floater MS (1997) Parametrization and smooth approximation of surface triangulations. *Comput Aided Geom Des* 14(3):231–250 [https://doi.org/10.1016/S0167-8396\(96\)00031-3](https://doi.org/10.1016/S0167-8396(96)00031-3)

17. Desbrun M, Meyer M, Alliez P (2002) Intrinsic parameterizations of surface meshes. *Comput Graph Forum* 21(3):209–218 <https://doi.org/10.1111/1467-8659.00580>
18. Hughes JF, van Dam A, McGuire M, Sklar DF, Foley JD, Feiner SK et al (2013) *Computer graphics: principles and practice*, 3rd edn. Addison-Wesley Professional, New Jersey
19. Marschner S, Shirley P (2015) *Fundamentals of computer graphics*, 4th edn. A K Peters/CRC Press, Boca Raton
20. Kang GD, Pérennou T, Diaz M (2008) Barycentric location estimation for indoors localization in opportunistic wireless networks. In: *Proceedings of the 2nd international conference on future generation communication and networking*, IEEE, Hainan, 13–15 December 2008 <https://doi.org/10.1109/FGCN.2008.85>
21. Skala V (2008) Barycentric coordinates computation in homogeneous coordinates. *Comput Graph* 32(1):120–127 <https://doi.org/10.1016/j.cag.2007.09.007>

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)

---