New Geometric Constraint Solving Formulation: Application to the 3D Pentahedron

Hichem Barki[1], Jean-Marc Cane[2], Dominique Michelucci[2], and Sebti Foufou[1,2]

[1] CSE Dep., College of Engineering, Qatar University, PO BOX 2713, Doha, Qatar
[2] LE2I, UMR CNRS 6306, University of Burgundy, 21000 Dijon, France
{hbarki,sfoufou}@qu.edu.qa,
{jean-marc.cane,dominique.michelucci}@u-bourgogne.fr

Abstract. Geometric Constraint Solving Problems (GCSP) are nowadays routinely investigated in geometric modeling. The 3D Pentahedron problem is a GCSP defined by the lengths of its edges and the planarity of its quadrilateral faces, yielding to an under-constrained system of twelve equations in eighteen unknowns. In this work, we focus on solving the 3D Pentahedron problem in a more robust and efficient way, through a new formulation that reduces the underlying algebraic formulation to a well-constrained system of three equations in three unknowns, and avoids at the same time the use of placement rules that resolve the under-constrained original formulation. We show that geometric constraints can be specified in many ways and that some formulations are much better than others, because they are much smaller and they avoid spurious degenerate solutions. Several experimentations showing a considerable performance enhancement ($\times 42$) are reported in this paper to consolidate our theoretical findings.

Keywords: Geometric Constraint Solving Problems, Parametrization, 3D Pentahedron.

1 Introduction

GCSPs have retained much of the researchers attention since several decades [6,5,13]. This attention may be justified by the advances in computing systems, in terms of both hardware capabilities and software facilities, which translated into a growing need for new CAD/CAM techniques and opened new perspectives for the implementation of researchers ideas. Despite the large number of existing works, expressing and solving geometric constraint systems is still an active research topic and much more effort has to be done in this direction.

This paper considers a particular GCSP problem: the 3D pentahedron. In this work, we focus on the convex pentahedron, so the term pentahedron implicitly refers to the convex version. To the best of our knowledge, no work has been done in the literature to study this problem and this is the first work that deals with the pentahedron problem. A resembling geometric problem is the octahedron one, also called the Stewart platform. This problem is similar to the pentahedron

A. Elmoataz et al. (Eds.): ICISP 2014, LNCS 8509, pp. 594–601, 2014.

in the fact that both of them are composed of six vertices in $\mathbb{E}^3$. In a pioneering work, Michelucci et al. [12] proposed a method that reduces the octahedron problem into a non-linear system in two unknowns and two equations, through the use of Cayley-Menger determinants.

In this work, we show that naive formulations of geometric constraint systems result in spurious and degenerate solutions. Such irrelevant and parasite solutions hinder the solving process, as they may form manifolds that slow down interval solvers [4]. These solvers handle the spurious manifolds with small residual boxes. However, in such boxes, it is not possible to prove the uniqueness of one regular root, say for example with Newton-Kantorovich theorem [7].

The main contribution of our work consists in a new formulation of the 3D pentahedron GCSP, that yields to a considerable reduction in the underlying algebraic system complexity, and discards spurious roots inherent to the classical formulation of the problem. This formulation does not only improve the solving performance as our experimentations prove, but it also broadens the range of interval solvers that can be used to solve the reduced system, compared to the impossible usage cases of many solvers when it comes to solve the more complex classical system formulation.

The rest of this paper is organized as follows: we first discuss the classical pentahedron problem in section 2. Then, we present in detail our new formulation in section 3. In section 4, we expose some relevant hints about our implementation, provide a performance benchmark, and a comparative study of the results of solving the pentahedron problem with the two formulations. Finally, we discuss our future work directions.

2 The Classical 3D Pentahedron GCSP

A GCSP is composed of a set of geometric objects, whose placement must fulfill a set of geometric constraints. The 3D pentahedron problem is composed of six points: p_1, p_2, p_3, q_1, q_2, and q_3. Triples of points (p_1, p_2, p_3) and (q_1, q_2, q_3) constitute the vertices of the two triangular facets of the pentahedron, while the remaining three quadrilateral facets denoted as F_1, F_2, and F_3 have respective vertices (p_2, p_3, q_3, q_2), (p_3, p_1, q_1, q_3), and (p_1, p_2, q_2, q_1), cf. Fig. 1(a).

The classical formulation of the pentahedron problem defines twelve constraints: nine distances between all the pairs of adjacent points: $d_1 = d(p_1, p_2)$, $d_2 = d(p_1, p_3)$, $d_3 = d(p_2, p_3)$, $d_4 = d(q_1, q_2)$, $d_5 = d(q_1, q_3)$, $d_6 = d(q_2, q_3)$, $d_7 = d(p_1, q_1)$, $d_8 = d(p_2, q_2)$, $d_9 = d(p_3, q_3)$ and three coplanarities of the quadrilateral facets: $copl(F_1)$, $copl(F_2)$, and $copl(F_3)$.

In the Euclidean three-dimensional space $\mathbb{E}^3$, if we put $p_1(x_1, x_2, x_3)$, $p_2(x_4, x_5, x_6)$, $p_3(x_7, x_8, x_9)$, $q_1(x_{10}, x_{11}, x_{12})$, $q_2(x_{13}, x_{14}, x_{15})$, and $q_3(x_{16}, x_{17}, x_{18})$, then even if the classical formulation leads to a structurally well-defined system, at the algebraic level, it implies an under-constrained system of twelve equations (constraints) in eighteen unknowns (the points Cartesian coordinates), with an infinite number of solutions. In GCSP literature, a common way to deal with this situation (whenever possible) is to use placement rules that constraint the

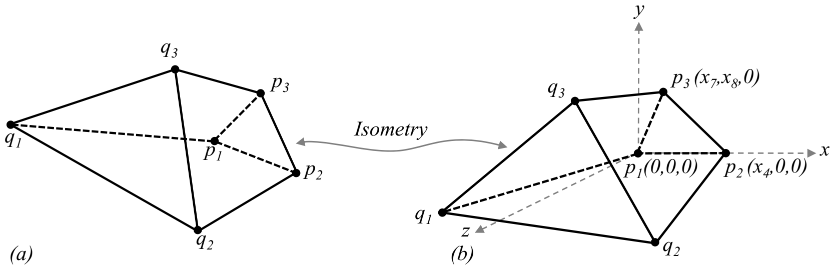

Fig. 1. (a) The general 3D pentahedron GCSP. (b) Adopted placement rules for a well-constrained pentahedron system.

placement of a particular subset of the original system and transform it into a well-constrained algebraic system without affecting the set of possible solutions, thus guaranteeing that the later system is consistent [3]. The finitely many solutions of the reduced system allow to obtain the infinitely many solutions of the original under-constrained system, up to isometries (composition of rotations, translations, and symmetries).

For the pentahedron, we adopt the three points placement rule illustrated in Fig. 1(b) for fixing the coordinates of the three points p_1, p_2, and p_3. Point $p_1(0,0,0)$ is placed at the coordinates origin, point $p_2(x_4,0,0)$ is placed at the positive x-axis at a distance d_1 from p_1, and point $p_3(x_7,x_8,0)$ is placed in the xy-plane with positive y coordinate ($x_8 > 0$), at respective distances d_2 and d_3 from point p_1, thus implying that $x_1 = x_2 = x_3 = x_5 = x_6 = x_9 = 0$. If we denote by C_1 to C_{12} the well-constrained pentahedron problem of twelve equations in twelve unknowns is algebraically expressed as follows:

$$\left\{\begin{array}{lll}
C_1: & x_4 - d_1 & = 0 \\
C_2: & x_7^2 + x_8^2 - d_2^2 & = 0 \\
C_3: & (x_4 - x_7)^2 + x_8^2 - d_3^2 & = 0 \\
C_4: & (x_{10} - x_{13})^2 + (x_{11} - x_{14})^2 + (x_{12} - x_{15})^2 - d_4^2 & = 0 \\
C_5: & (x_{10} - x_{16})^2 + (x_{11} - x_{17})^2 + (x_{12} - x_{18})^2 - d_5^2 & = 0 \\
C_6: & (x_{13} - x_{16})^2 + (x_{14} - x_{17})^2 + (x_{15} - x_{18})^2 - d_6^2 & = 0 \\
C_7: & x_{10}^2 + x_{11}^2 + x_{12}^2 - d_7^2 & = 0 \\
C_8: & (x_4 - x_{13})^2 + x_{14}^2 + x_{15}^2 - d_8^2 & = 0 \\
C_9: & (x_7 - x_{16})^2 + (x_8 - x_{17})^2 + x_{18}^2 - d_9^2 & = 0 \\
C_{10}: & x_4(x_8(x_{18} - x_{15}) + x_{15}x_{17} - x_{14}x_{18}) & \\
 & \quad - x_7(x_{15}x_{17} - x_{14}x_{18}) + x_8(x_{15}x_{16} - x_{13}x_{18}) & = 0 \\
C_{11}: & x_7(x_{11}x_{18} - x_{12}x_{17}) - x_8(x_{10}x_{18} - x_{12}x_{16}) & = 0 \\
C_{12}: & -x_4(x_{12}x_{14} - x_{11}x_{15}) & = 0
\end{array}\right. \tag{1}$$

where the coplanarity constraints C_{10} to C_{12} are computed as 4×4 determinants, which translate into null volumes of the tetrahedra corresponding to the four vertices of each planar facet.

$$C_{10}: \begin{vmatrix} x_4 & 0 & 0 & 1 \\ x_7 & x_8 & 0 & 1 \\ x_{16} & x_{17} & x_{18} & 1 \\ x_{13} & x_{14} & x_{15} & 1 \end{vmatrix} = 0,\ C_{11}: \begin{vmatrix} x_7 & x_8 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ x_{10} & x_{11} & x_{12} & 1 \\ x_{16} & x_{17} & x_{18} & 1 \end{vmatrix} = 0,\ C_{12}: \begin{vmatrix} 0 & 0 & 0 & 1 \\ x_4 & 0 & 0 & 1 \\ x_{13} & x_{14} & x_{15} & 1 \\ x_{10} & x_{11} & x_{12} & 1 \end{vmatrix} = 0 \qquad (2)$$

The obtained system is well-constrained and has a finite number of solutions – under mild assumptions. Though correct, the reduced system is awkward. First, it may have spurious roots, where all vertices are coplanar. Indeed, consider this problem in 2D (planar pentahedron). In this case, the planarity constraints disappear and only nine 2D point-point distance constraints remain. It turns out that this system is well-constrained: it is well known from rigidity theory and Laman's theorem that n 2D vertices are well-constrained by $c = 2n - 3$ distances [9], and no sub-system is over-constrained (e.g., no four vertices are involved in more than 3 constraints). In our case, $n = 6$ and $c = 2n - 3 = 9$. In consequence, this 2D problem is well-constrained: spurious roots are of finite number. To get rid of such spurious system roots and even considerably reduce the system complexity, we propose in the next section a new formulation for the pentahedron problem.

3 New Formulation of the 3D Pentahedron Problem

One main observation about the well-constrained pentahedron system of twelve unknowns given by the classical formulation is that it misses an essential property, which is specific to non-degenerate solutions. This property consists in the fact that the three supporting lines of the pentahedron edges $[p_j q_j], j = 1, 2, 3$ must be either concurrent or parallel. Indeed, the supporting planes P_1, P_2, and P_3 of the respective three quadrilateral facets F_1, F_2, and F_3 meet at a common point named i, which may be located at infinity if the three intersection lines of these supporting planes are parallel. Clearly, these intersection lines $l_1 = P_2 \cap P_3$, $l_2 = P_3 \cap P_1$, and $l_3 = P_1 \cap P_2$ pass through point $i = P_1 \cap P_2 \cap P_3$ (Fig. 2(a)).

The aforementioned property of lines l_1, l_2, and l_3 inspires our new formulation, and suggests another way of expressing the constraints of the pentahedron problem. Let us suppose that lines l_1, l_2, and l_3 are concurrent in point i. The "theorem of Al-Kashi", also known as the "law of cosines", states that given a triangle (a, b, c) in $\mathbb{E}^2$, if we denote by α, β, and γ the angles corresponding to its respective vertices a, b, and c, and by A, B, and C the lengths of the sides respectively opposite to these angles (cf. Fig. 2(b)), then the length of any side of the triangle, say A, can be given in terms of the lengths of the two other triangle sides and the cosine of the opposite angle as follows:

$$A^2 = B^2 + C^2 - 2BC\cos\alpha. \qquad (3)$$

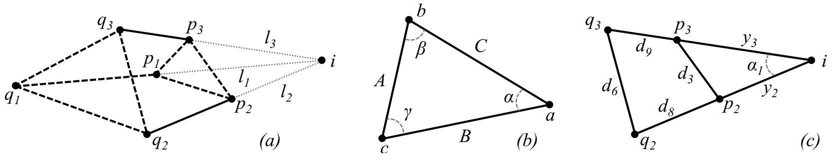

Fig. 2. New formulation of the pentahedron problem. (a) Concurrent lines l_1, l_2, and l_3 intersect in point i. (b) Illustration of Al-Kashi theorem for triangles. (c) A 2D view illustrating the application of Al-Kashi theorem for the new formulation of the pentahedron.

The theorem of Al-Kashi generalizes the Pythagorean theorem for non-right triangles. By considering the quadrangular facet F_1 of the pentahedron, and applying the Al-Kashi theorem in its supporting plane P_1 on triangles (i, p_2, p_3) and (i, q_2, q_3), cf. Fig. 2(c), we obtain:

$$d_3^2 = y_2^2 + y_3^2 - 2y_2y_3cos\alpha_1 \quad (4)$$

$$d_6^2 = (y_2 + d_8)^2 + (y_3 + d_9)^2 - 2(y_2 + d_8)(y_3 + d_9)cos\alpha_1 \quad (5)$$

where $y_2 = d(i, p_2)$ and $y_3 = d(i, p3)$ represent the lengths of the two sides of triangle (i, p_2, p_3) that are incident to point i, and α_1 denotes the angle formed by these sides in the plane P_1. α_1 also corresponds to the angle formed by the two sides of triangle (i, q_2, q_3) that are incident to point i. In consequence, by substituting the expression of $\cos\alpha_{23}$ from Eq. 4 into Eq. 5, we get a nonlinear equation in two unknowns y_2, and y_3, where d_3 and d_6 are constants. By proceeding analogously for the pairs of triangles (i, p_3, p_1) and (i, q_3, q_1) in the plane P_2 of facet F_2, and (i, p_1, p_2) and (i, q_1, q_2) in the plane P_3 of facet F_3, we finally obtain our new formulation of the pentahedron, as a system of only three equations in three unknowns y_1, y_2, y_3 as follows:

$$\begin{cases} C_1' : (y_1^2 + y_2^2 - d_1^2)(y_1 + d_7)(y_2 + d_8) - y_1y_2((y_1 + d_7)^2 + (y_2 + d_8)^2 - d_4^2) = 0 \\ C_2' : (y_1^2 + y_3^2 - d_2^2)(y_1 + d_7)(y_3 + d_9) - y_1y_3((y_1 + d_7)^2 + (y_3 + d_9)^2 - d_5^2) = 0 \\ C_3' : (y_2^2 + y_3^2 - d_3^2)(y_2 + d_8)(y_3 + d_9) - y_2y_3((y_2 + d_8)^2 + (y_3 + d_9)^2 - d_6^2) = 0 \end{cases} \quad (6)$$

where $y_1 = d(i, p_1)$. It is clear that the new formulation, by means of the theorem of Al-Kashi, led to a new system that is much simpler than the classical one. Moreover, this new system has no spurious root. Another advantage of our formulation consists in the avoidance of placement rules which are necessary in the original formulation to make the system well-constrained.

Finally, the solutions of the original system, i.e., coordinates $x_k, k = 10, \ldots, 18$ of points q_1, q_2, and q_3 (coordinates of points p_1, p_2, p_3 are determined in the classical formulation by placement rules) can be easily computed from the solutions $y_j, j = 1, 2, 3$ of the new formulation as follows: (1) the three distance constraints $y_j = d(i, p_j)$ constitute a system of three quadratic equations whose solution gives the coordinates of point i, and (2) the three proportionality formulas $\vec{iq_j} = t_j\,\vec{ip_j}, j = 1, 2, 3$ imply that $\vec{ip_j}\vec{iq_j} = \|\vec{ip_j}\|^2 t_j$, the later three equations give the values of the three parameters t_j, which when substituted back in the proportionality equations, give the coordinates of points $q_j, j = 1, 2, 3$ as $q_j = i + t_j(p_j + i)$. The detailed developments are omitted for the sake of brevity.

4 Experiments and Results

We implemented the two pentahedron formulations in C++. We used *ALIAS-C++* interval analysis library [11] for solving the underlying algebraic systems of non-linear equations. Due to space limitations, we present only a subset of our experimentations, by providing a summary of our performance comparisons, without detailing other aspects. For the same reason, we also omit the presentation of other benchmarks performed with other interval solvers. Our results have been obtained on a 2.4 GHz Intel Core i7 computer, equipped with 16 GB of RAM, and running a 32 bits linux version, with `g++ 4.8.1`.

In the current experiments, we used the general purpose interval solver of ALIAS-C++, which is implemented in the function `Solve_General_int()`. Other solving techniques that make use of the Jacobi and Hessian of the equations system are provided by ALIAS-C++ [1].

First of all, we shall note that on a sample of randomly generated 3D pentahedra systems, the general purpose solver of ALIAS-C++ failed to solve the twelve equations of the classical formulation, because of the high memory requirements of the default full bisection strategy combined with the number of unknowns that exceeds ten. When using a single bisection strategy, the average running time is 353.32 seconds. With the same systems sample and considering our new formulation of three equations, the general solver successfully computed all the solutions in an average time of 11.44 seconds with the full bisection method, which shows an advantage of our formulation that makes it more practical because it is smaller. When using single bisection, the running time dropped to 8.43 seconds, which represents a performance gain of $\times 41.91$ over classical formulation. Several experiments revealed that when decreasing the number of maximal boxes or nD intervals to be used with ALIAS-C++, our formulation is still solvable until a reasonable number, while the classical formulation becomes quickly unfeasible for the same number of boxes, thus revealing the memory footprint improvements of our formulation.

Our new formulation is limited in two aspects. First, it does not handle pentahedra for which the lines l_1, l_2, and l_3 are parallel. In such a case, the intersection point i is located at infinity and such a system cannot be solved by ALIAS-C++. Second, our current formulation supposes that intersection point i of concurrent lines l_1, l_2, and l_3 is reached towards the positive direction of vector $q_j\vec{p}_j$ (Fig. 2(c)). However, the opposite case may happen, as point i may be located when moving along the negative direction of vector $q_j\vec{p}_j$. In such a case, the correct formulation can be derived from the current one just by swapping d_4 and d_1, d_5 and d_2, and d_6 and d_3 in Eq. 6. When using ALIAS-C++

with the formulation of Eq. 6, the opposite formulation can be easily detected as ALIAS-C++ computed negative values for distances y_1, y_2, and y_3, which implies to recompute them using the opposite formulation to get correct values. Potential solutions for such concerns are given in the next section, in addition to some ongoing and future work directions.

5 Conclusion and Future Work

In this work, we have presented a new formulation, based on the "theorem of Al-Kashi", for the reduction of the classical under-constrained 3D pentahedron problem of twelve equations in eighteen unknowns, to an equivalent well-constrained problem of only three equations in three unknowns. Our new formulation has the advantage that it is more robust since the underlying system of equations has no spurious roots, compared to the classical formulation of the pentahedron. It also avoids the use of placement rules that reduce the classical problem into a well-constrained system. Our experimentations revealed that the new formulation is more efficiently handled by some interval solvers. In addition, the classical formulation was impractical with some implementations of interval solvers, due to the imposed limit on the maximum number of unknowns, which reduces the range of usable solvers, contrary to our formulation which can be handled by practically any solver, thanks to the drastically reduced number of unknowns. The later statement implies that more efficient solvers can even improve our running times.

As future work, we are addressing the two aforementioned limitations of our work. Concerning the parallel lines configuration, we are investigating a technique whose principle consists in solving this problem in two steps: solving a 3D triangle problem, and then using the result for solving a pyramid problem having a quadrilateral base. The later result gives the solution of the parallel lines configuration through simple translations. We are also working to find a unified formulation of the relative position of intersection point i w.r.t. pentahedron vertices p_i and q_i. We started investigating the use of Cayley-Manger determinants [10,14] to develop a unique formulation that is independent from the relative position of point i.

A second direction concerns the use of another property that may lead to an interesting formulation of the pentahedron problem. This property states that the supporting lines of the opposite edges $[p_j p_k]$ and $[q_j q_k]$ of each quadrilateral facet, where $j, k = 1, 2, 3, j \neq k$, meet in three points i_1, i_2, and i_3 which are necessarily collinear, because each of the aforementioned points is the intersection of a line lying on the supporting plane of points p_1, p_2, and p_3, with a line lying in the supporting plane of q_1, q_2, and q_3, i.e., points i_1, i_2, and i_3 lie on the intersection line between the aforementioned two planes. This property is known as the "Desargues' theorem" [8,2], which holds both in 2D and in 3D.

Acknowledgments. This publication was made possible by NPRP grant #09-906-1-137 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. ALIAS-C++, A C++ Algorithms Library of Interval Analysis for equation Systems, http://www-sop.inria.fr/coprin/logiciels/ALIAS/ALIAS-C++/ALIAS-C++.html
2. Coxeter, H.: The Beauty of Geometry: Twelve Essays. Dover books on mathematics. Dover Publications (1999)
3. Durand, C.: Symbolic and numerical techniques for constraint solving. Ph.D. thesis, West Lafayette, IN, USA (1998)
4. Füenfzig, C., Michelucci, D., Foufou, S.: Polytope-based computation of polynomial ranges. Computer Aided Geometric Design 29(1), 18–29 (2012)
5. Hoffmann, C., Joan-Arinyo, R.: A brief on constraint solving. Computer-Aided Design and Applications 2(5), 655–663 (2005)
6. Hoffmann, C., Peters, J.: Geometric constraints for CAGD. In: Proceedings of the 3rd International Conference on Mathematical Methods for CAGD, Ulvic., pp. 237–254. Vanderbilt University Press (1995)
7. Hubbard, J., Hubbard, B.: Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach. Matrix Editions (2007)
8. Johnson, R., Young, J.: Modern Geometry: An Elementary Treatise on the Geometry of the Triangle and the Circle. Houghton, Mifflin Company (1929)
9. Laman, G.: On graphs and rigidity of plane skeletal structures. Journal of Engineering Mathematics 4(4), 331–340 (1970)
10. MacDuffee, C.: Theory of Equations. John Wiley (1954)
11. Merlet, J.P.: ALIAS: an interval analysis based library for solving and analyzing system of equations. In: SEA, pp. 14–16 (2000)
12. Michelucci, D., Foufou, S.: Using Cayley-Menger determinants for geometric constraint solving. In: Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications, SM 2004, pp. 285–290. Eurographics Association (2004)
13. Michelucci, D., Foufou, S., Lamarque, L., Schreck, P.: Geometric constraints solving: Some tracks. In: Proceedings of the 2006 ACM Symposium on Solid and Physical Modeling, SPM 2006, pp. 185–196. ACM (2006)
14. Sommerville, D.: An Introduction to the Geometry of N Dimensions (1958)