Journal for Geometry and Graphics
Volume 21 (2017), No. 1, 119–130.

A Dual Quaternion Based Pipe Alignment Algorithm for Constrained Pipe Systems

Daniel Klawitter, Christian Tonn, Oliver Bringmann

FARO 3D Software GmbH
Tiergartenstrasse 79, D-01219 Dresden, Germany
email: daniel.klawitter@faro.com

Abstract. Faro 3D Software GmbH develops software for interpreting point clouds generated by 3D laser scanners. A main application is to extract complete pipe systems that may be found in industrial plants. The focus of our contribution is directed at an iterative algorithm to align these pipe systems. Patent Pending.

Key Words: Dual quaternion, laser scanning, constraints, pipe systems
MSC 2010: 53A17, 68U07, 51N20

1. Introduction

In plant industry there is a high demand for precise modelling tools to create complex plant pipe systems. The surveying tool of choice for this task is laser scanning. After scanning and modelling with the measured point cloud, the pipes follow precisely the reality via cylinder fitting methods. The pipe joints (e.g., elbows, tees, reducers and crosses) are usually catalogued part types. They are not defined by incident pipe axis angles. Indeed, they are mostly produced with fixed predefined pipe axis angles (e.g., a 90 or 45 degree elbow). This fact leads to small gaps between the joints and the fitted pipes.

Usually, plant pipe models are made for specialized design packages, which are used for piping isometrics and simulations (e.g., fluid dynamic simulations). For this kind of software the pipe system connection points at each joint have to be collinear with the opposite connection point of the other pipe or joint fixture. Hence, the pipe model does not only have to be precise, but is also constrained to be water-tight without gaps. This step from taking the measured pipe model to the constrained model is presented in the paper.

There are two main objectives: The approach should leave the fitted pipe axis mostly unaltered and should limit the movement of the fittings to a minimum. Additionally, the orientation of the joint connection points should always be perfectly collinear with each other without gaps. The argumentation is that the cylinder axes are well-defined through RANSAC and least-square methods while the other pipe parts might be insufficiently modelled (see [12]). The proposed algorithm deals with minimal movement of the fitted pipe axis, cycles in the

ISSN 1433-8157/$ 2.50

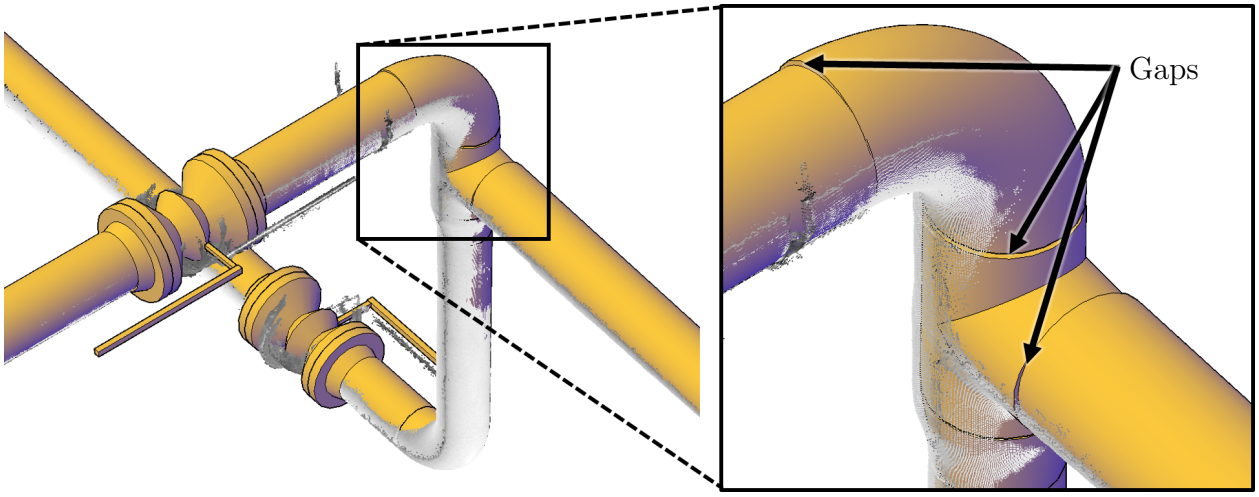

Figure 1: Gaps in the piping system.

piping system and fixed measured tie-in points using a dual quaternion approach. Moreover, the presented algorithm is a significant contribution for auto-routing features in plant design applications.

2. Mathematical background

To align multiple objects to each other a lot of constraints have to be satisfied. In fact the alignment of complicated systems with loops may in general not be solved analytically. These kinds of problems lead to complicated configurations in the kinematic image space that have to be investigated. The more complicated the problem, the more hopeless it will be to find an analytic solution. Loops in the system lead to additional contraints. Therefore, an analytical approach seems not to be suited to our problem. In this contribution, we suggest an iterative adaptive method using dual quaternion calculus to describe displacements.

2.1. Dual quaternions

Dual quaternions are quaternions with dual number entries. Therefore, we introduce quaternions and dual numbers first. Quaternions $\mathbb{H}$, introduced by W.R. Hamilton in 1843, constitute an elegant tool for the representation of rotations in three- and four-dimensional Euclidean case (see [6]). Furthermore, quaternions form a skew field, i.e., multiplication is not commutative. Unit quaternions are a double cover of the group SO(3). A general quaternion has the form

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}, \quad a, b, c, d \in \mathbb{R}, \tag{1}$$

where $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are the *quaternion units* with

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1. \tag{2}$$

Definition 1. Let q be a quaternion as in Eq. (1). Then, a is called the *scalar part* and $(b, c, d)^{\mathrm{T}}$ is called the *vector part*. A quaternion with vanishing scalar part is called a *vectorial quaternion*.

Quaternions form a four-dimensional vector space over the real numbers. Addition is defined component-wise by

$$q_1 + q_2 = (a_1 + a_2) + (b_1 + b_2)\mathbf{i} + (c_1 + c_2)\mathbf{j} + (d_1 + d_2)\mathbf{k}.$$

The multiplication rules for the quaternion units (2) are summarized in the following scheme:

$\cdot$	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$
$\mathbf{i}$	-1	$\mathbf{k}$	$-\mathbf{j}$
$\mathbf{j}$	$-\mathbf{k}$	-1	$\mathbf{i}$
$\mathbf{k}$	$\mathbf{j}$	$-\mathbf{i}$	-1

This scheme allows to extend the concept of quaternion multiplication to general quaternions. The real numbers form the *center* of the quaternions, i.e., real numbers commute with all other quaternions. Later, we see that the quaternions are an example of an $\mathbb{R}$-algebra.

Definition 2. The *anti-involution*

$$^*\colon \mathbb{H} \to \mathbb{H}, \quad a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} = q \;\mapsto\; q^* = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}$$

is called quaternion *conjugation*. The *norm* of a quaternion is defined by

$$\|q\| = \sqrt{qq^*} = \sqrt{a^2 + b^2 + c^2 + d^2}.$$

Every quaternion $q \neq 0$ has an *inverse quaternion* that can be calculated by

$$q^{-1} = \frac{1}{\|q\|^2} q^* = \frac{q^*}{qq^*}.$$

Unit quaternions are quaternions with norm equal to 1. With respect to quaternion multiplication unit quaternions form a group. Every unit quaternion can be represented by

$$q = \cos\varphi + d\sin\varphi \quad \text{with} \quad d = (d_1\mathbf{i} + d_2\mathbf{j} + d_3\mathbf{k}),$$

where d is an unit vector and $\varphi \in \mathbb{R}$. Moreover, unit quaternions can be used to describe rotations in three-dimensional Euclidean space $\mathbb{E}^3$. This can be realized with the so called *sandwich operator* $x \mapsto qxq^*$ where q is a unit quaternion and the coordinate vector x of a point is considered as a vectorial quaternion. Therefore, we follow [4] and apply the sandwich operator to the standard basis vectors of $\mathbb{R}^3$ written as vectorial quaternions. This means the basis vectors e_1 belonging to the x-component is expressed by the quaternion $x = \mathbf{i}$ and the effect of the sandwich operator results in

$$\begin{aligned} q\,\mathbf{i}\,q^* &= (a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k})\mathbf{j}(a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}) \\ &= (a\mathbf{i} - b - c\mathbf{k} + d\mathbf{j})\mathbf{j}(a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}) \\ &= (a^2 + b^2 - c^2 - d^2)\mathbf{i} + 2(bc + ad)\mathbf{j} + 2(bd - ac)\mathbf{k}. \end{aligned}$$

In the same way we see the action of this operator on the vectorial quaternions $\mathbf{j}$ and $\mathbf{k}$

$$\begin{aligned} q\,\mathbf{j}\,q^* &= 2(bc - ad)\mathbf{i} + (a^2 - b^2 + c^2 - d^2)\mathbf{j} + 2(cd + ab)\mathbf{k}, \\ q\,\mathbf{k}\,q^* &= 2(bd + ac)\mathbf{i} + 2(cd - ab)\mathbf{j} + (a^2 - b^2 - c^2 + d^2)\mathbf{k}. \end{aligned}$$

The sandwich operator is linear and the image of a vectorial quaternion is a vectorial quaternion again. Furthermore, the scalar product of two vectors $x, y \in \mathbb{R}^3$ is invariant under the

action of the sandwich operator and it is orientation preserving. When collecting the images of the basis vectors e_1, e_2, e_3 in a matrix A, we get

$$\mathrm{A} = \begin{pmatrix} a_{1,1} & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a_{2,2} & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a_{3,3} \end{pmatrix},$$

with

$$a_{1,1} = a^2+b^2-c^2-d^2, \quad a_{2,2} = a^2-b^2+c^2-d^2, \quad a_{3,3} = a^2-b^2-c^2+d^2.$$

This matrix is the well-known form of a rotation matrix. The components of a unit quaternion are the *Euler parameters* of a rotation.

Dual numbers

Like complex numbers dual numbers are an extension of the real numbers. A dual number has the form $z_\varepsilon = a + \varepsilon b$, where a and b are real numbers and ε is the *dual unit* that squares to zero, $\varepsilon^2 = 0$. Addition is defined component-wise. For two dual numbers the product is defined by

$$(a_1 + \varepsilon b_1)(a_2 + \varepsilon b_2) = a_1 a_2 + \varepsilon(a_1 b_2 + a_2 b_1).$$

The set of dual numbers

$$\mathbb{D} := \left\{ a + \varepsilon b \mid a, b \in \mathbb{R}, \varepsilon^2 = 0 \right\}$$

together with addition and multiplication forms a commutative ring with identity. Moreover, the dual numbers form a two-dimensional commutative unital associative algebra over the real numbers. Dual numbers with vanishing real part are zero divisors

$$(\varepsilon a)(\varepsilon b) = \varepsilon^2(ab) = 0.$$

Definition 3. For a dual number $z_\varepsilon = a + \varepsilon b$ the dual number $\tilde{z}_\varepsilon = a - \varepsilon b$ is called the *conjugate* dual number. The *norm* of a dual number that is no zero divisor then is

$$\|z_\varepsilon\| := \sqrt{z_\varepsilon \tilde{z}_\varepsilon} = \sqrt{(a+\varepsilon b)(a-\varepsilon b)} = |a|.$$

Any dual number without vanishing real part has an inverse dual number

$$z_\varepsilon^{-1} = (a+\varepsilon b)^{-1} := \frac{1}{a^2}\tilde{z}_\varepsilon\,.$$

Analytic functions can be extended to dual functions with help of their formal Taylor expansion. Note that any power of ε that is bigger than one vanishes. Therefore, we get the Taylor expansion

$$f(a+\varepsilon b) = f(a) + \varepsilon b f'(a),$$

which is the dual extension of the analytic function.

Remark 1. It is possible to calculate the inverse of a dual number by the Taylor expansion of z_ε^{-1} to make the definition of the inverse dual number clear

$$z_\varepsilon^{-1} = (a+\varepsilon b)^{-1} = \frac{1}{a} - \varepsilon\frac{b}{a^2} = \frac{1}{a^2}(a-\varepsilon b) = \frac{1}{a^2}\tilde{z}_\varepsilon\,.$$

Dual vectors

Later we will use dual vectors. Therefore, we introduce the n-dimensional module

$$\mathbb{D}^n := \{v_\varepsilon \mid v_\varepsilon = v + \varepsilon\bar{v},\ \varepsilon^2 = 0,\ v, \bar{v} \in \mathbb{R}^n\}.$$

A dual vector is the sum of its real- and dual part

$$v_\varepsilon = v + \varepsilon\bar{v},\ \text{with } v, \bar{v} \in \mathbb{R}^n.$$

We define a standard scalar product on this module by

$$v_\varepsilon w_\varepsilon^{\mathrm{T}} = \langle v_\varepsilon, w_\varepsilon\rangle_\varepsilon = \langle v, w\rangle + \varepsilon(\langle v, \bar{w}\rangle + \langle \bar{v}, w\rangle),$$

where $\langle\cdot,\cdot\rangle$ denotes the standard scalar product of $\mathbb{R}^n$. Especially for the dimension $n = 3$ we are able to define a cross product by

$$v_\varepsilon \times_\varepsilon w_\varepsilon = v \times w + \varepsilon(\bar{v} \times w + v \times \bar{w}).$$

Dual quaternions

Dual quaternions were introduced by E. Study [10]. Nowadays, dual quaternions form a frequently used tool for the description of Euclidean kinematics in three dimensions (see [1, 4, 5] or [7]). In this section we give a brief introduction to dual quaternions. Our intention is to put this concept in a more general context by using Clifford algebras. Quaternions with dual number components are called dual quaternions and are denoted by

$$\mathbb{H}_d := \{a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} + \varepsilon(c_0 + c_1\mathbf{i} + c_2\mathbf{j} + c_3\mathbf{k}) \mid a_0, \ldots, a_3, c_0, \ldots, c_3 \in \mathbb{R}\}.$$

Multiplication is defined with the relations for quaternion. Furthermore, the dual unit ε commutes with the quaternion units $\varepsilon\mathbf{i} = \mathbf{i}\varepsilon$, $\varepsilon\mathbf{j} = \mathbf{j}\varepsilon$, $\varepsilon\mathbf{k} = \mathbf{k}\varepsilon$. Dual quaternions form an eight-dimensional vector space over the real numbers. The basis elements are 1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$, ε, $\varepsilon\mathbf{i}$, $\varepsilon\mathbf{j}$, $\varepsilon\mathbf{k}$.

Displacements

Euclidean displacements can be described by dual unit quaternions. A dual quaternion $q_\varepsilon = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} + \varepsilon(c_0 + c_1\mathbf{i} + c_2\mathbf{j} + c_3\mathbf{k})$ is normed or a dual unit quaternion, if the norm is equal to one

$$N(q_\varepsilon) := q_\varepsilon q_\varepsilon^* = a_0^2 + a_1^2 + a_2^2 + a_3^2 + 2\varepsilon(a_0c_0 + a_1c_1 + a_2c_2 + a_3c_3) = 1,$$

where the conjugation is the quaternion conjugation

$$q_\varepsilon^* = (q_1 + \varepsilon q_2)^* = q_1^* + \varepsilon q_2^*.$$

Therefore, a dual unit quaternion satisfies two relations in the components $a_0, \ldots, c_3$

$$a_0^2 + a_1^2 + a_2^2 + a_3^2 = 1, \qquad a_0c_0 + a_1c_1 + a_2c_2 + a_3c_3 = 0.$$

Dual unit quaternions are denoted by $\mathbb{U}_d$ and form a group with respect to multiplication. Moreover, dual unit quaternions form a double cover of the group of Euclidean displacements

SE(3), details can be found in [4]. Usually, a displacement is described by the sandwich operator. We start with a dual unit quaternion

$$q = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} + \varepsilon(c_0 + c_1\mathbf{i} + c_2\mathbf{j} + c_3\mathbf{k})$$

and a dual quaternion of the form $p = 1 + \varepsilon(x\mathbf{i} + y\mathbf{j} + z\mathbf{k})$, representing the point $P = (x, y, z)^{\mathrm{T}} \in \mathbb{R}^3$. We apply the sandwich operator as

$$\begin{aligned} qpq^* =& 1 + (xa_0^2 + 2za_0a_2 - 2ya_0a_3 + 2c_1a_0 + xa_1^2 + 2ya_1a_2 \\ &+ 2za_1a_3 - 2c_0a_1 - xa_2^2 + 2c_3a_2 - xa_3^2 - 2c_2a_3)\,\varepsilon\mathbf{i} \\ &+ (ya_0^2 - 2za_0a_1 + 2xa_0a_3 + 2c_2a_0 - ya_1^2 + 2xa_1a_2 \\ &- 2c_3a_1 + ya_2^2 + 2za_2a_3 - 2c_0a_2 - ya_3^2 + 2c_1a_3)\,\varepsilon\mathbf{j} \\ &+ (za_0^2 + 2ya_0a_1 - 2xa_0a_2 + 2c_3a_0 - za_1^2 + 2xa_1a_3 \\ &+ 2c_2a_1 - za_2^2 + 2ya_2a_3 - 2c_1a_2 + za_3^2 - 2c_0a_3)\,\varepsilon\mathbf{k}. \end{aligned}$$

If we rewrite the result as s product of a matrix with a vector vector in homogeneous coordinates we arrive at

$$\begin{pmatrix} 1 \\ x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l & a_0^2+a_1^2-a_2^2-a_3^2 & 2a_1a_2-2a_0a_3 & 2a_0a_2+2a_1a_3 \\ m & 2a_0a_3+2a_1a_2 & a_0^2-a_1^2+a_2^2-a_3^2 & 2a_2a_3-2a_0a_1 \\ n & 2a_1a_3-2a_0a_2 & 2a_0a_1+2a_2a_3 & a_0^2-a_1^2-a_2^2+a_3^2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ x \\ y \\ z \end{pmatrix},$$

where

$$\begin{aligned} l &= 2c_1a_0 - 2c_0a_1 + 2c_3a_2 - 2c_2a_3, \\ m &= 2c_2a_0 - 2c_3a_1 - 2c_0a_2 + 2c_1a_3, \\ n &= 2c_3a_0 + 2c_2a_1 - 2c_1a_2 - 2c_0a_3. \end{aligned}$$

This matrix vector product represents an Euclidean displacement (see [4]). A parametrisation of the special Euclidean group with the help of dual unit quaternions is given by

$$\begin{aligned} Q &= a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} + c_0\varepsilon + c_1\varepsilon\mathbf{i} + c_2\varepsilon\mathbf{j} + c_3\varepsilon\mathbf{k} \\ &= \cos\frac{\varphi}{2} - \sin\frac{\varphi}{2}l_0\,\mathbf{i} - \sin\frac{\varphi}{2}l_1\,\mathbf{j} - \sin\frac{\varphi}{2}l_2\,\mathbf{k} - \frac{v}{2}\sin\frac{\varphi}{2}\,\varepsilon - \left(\sin\frac{\varphi}{2}l_3 + \frac{v}{2}\cos\frac{\varphi}{2}l_0\right)\varepsilon\mathbf{i} \\ &- \left(\sin\frac{\varphi}{2}l_4 + \frac{v}{2}\cos\frac{\varphi}{2}l_1\right)\varepsilon\mathbf{j} - \left(\sin\frac{\varphi}{2}l_5 + \frac{v}{2}\cos\frac{\varphi}{2}l_2\right)\varepsilon\mathbf{k}, \end{aligned}$$

where φ is the rotation angle and v is the magnitude of a translation in the direction defined by the Plücker coordinate vector $L = (l_0 : l_1 : l_2 : l_3 : l_4 : l_5)$ (see [2]).

2.2. Study's sphere

A point model for the group of Euclidean displacements is Study's sphere (see [5] or [8, section 2.3]). Each displacement may be represented as unit dual quaternion. Note that there are two unit dual quaternions q and $-q$ describing the same Euclidean displacement.

We consider a dual vector $v_\varepsilon = v + \varepsilon\bar{v} \in \mathbb{D}^4$. The canonical scalar product of this dual vector with itself results in

$$\langle v_\varepsilon, v_\varepsilon\rangle_\varepsilon = \langle v, v\rangle + 2\varepsilon\,\langle v, \bar{v}\rangle\,. \tag{3}$$

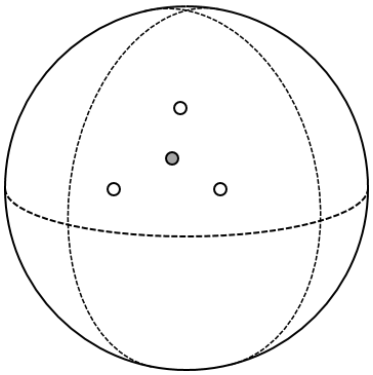

Figure 2: Study's Sphere

If the dual vector is built from a dual unit quaternion corresponding to an Euclidean displacement, the real part contains the *Euler parameters* corresponding to the rotational part of the displacement and the translational part of the displacement may be derived from the dual part. The dual part of Eq. (3) vanishes, and thus, the value of the scalar product is a real number. In this case the scalar product (3) is equal to one and we call a dual vector v_ε with $\langle v_\varepsilon, v_\varepsilon \rangle_\varepsilon = 1$, a *normalized dual vector* or *dual unit vector*. Therefore, we can identify the group of Euclidean displacement SE(3) with points of the dual unit sphere

$$S^3_{\mathbb{D}} := \left\{ x_\varepsilon \in \mathbb{D}^4 \mid \langle x_\varepsilon, x_\varepsilon \rangle_\varepsilon = 1 \right\}.$$

A schematic visualisation is illustrated in Figure 2. Points on Study's sphere correspond to displacements and the grey point corresponds to the mean point on Study's sphere. Moreover, these points may also be interpreted as coordinate systems that result by the application

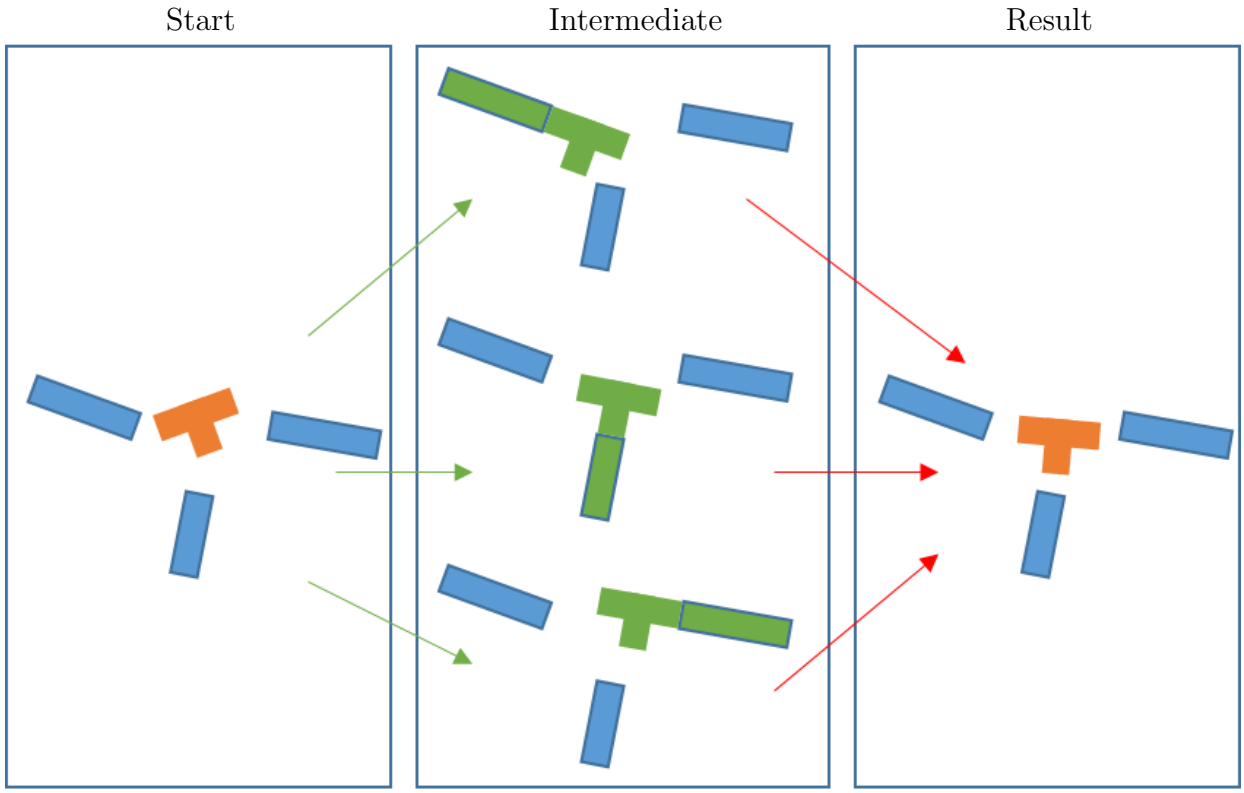

Figure 3: Mean displacement.

of the corresponding transformation to a special coordinate system, i.e., the home or base coordinate system. Great circles on $S^3_{\mathbb{D}}$ correspond to helicoidal motions transforming the coordinate system defined by the application of displacement corresponding to the first point to the coordinate system defined by the second one (see [3]). Ruled surface interpolation with the help of Study's sphere $S^2_{\mathbb{D}}$ as point model for oriented lines was performed in [9]. This model allows an elegant way to find a *mean* displacement. Via slerping it is possible to define great circles on Study's sphere connecting two points and to determine the point corresponding to the midpoint on this curve. The corresponding displacement can be interpreted as mean displacement. Using dual vector calculus this can be done by adding dual vectors and normalizing the result afterwards. Through repeated application of this procedure the mean displacement of multiple displacements can be computed. Figure 3 visualizes the concept of a mean displacement. Firstly, the tee component is transformed to satisfy the constraints of each neighbouring pipe (middle, green). In a second step the three resulting displacements are represented as dual unit quaternions to compute the mean displacement that is also visualized in Figure 3 (right, orange).

Remark 2. For the application in our alignment algorithm the mean transformation is a weighted sum. This means every point on Study's sphere is weighted by the length of the corresponding pipe or cluster where the length of a cluster is defined by the sum of the lengths of all contained objects. Afterwards the weighted dual vectors are summed up. Normalizing the resulting dual vector yields a dual unit quaternion corresponding to the weighted mean displacement. This procedure ensures that a short pipe or a small cluster does not influence the resulting displacement too much.

3. The algorithm

To describe the proposed algorithm we define some terms:

- *ConnectionPoint* is an oriented line-element. Thus it cotains a position, i.e., a point and normalized direction vector. *ConnectionPoints* are attached to every pipe and pipe joint. If two components are connected to each other their *ConnectionPoints* are compatible. This means the position is the same and the normalized direction vector has opposite direction.
- *Pipe* describes a straight pipe with a *ConnectionPoint* at each end. The *ConnectionPoints* are collinear with opposite orientation. The length of the pipe can be altered.
- *Fix* is a rigid set of *ConnectionPoints* that reflect the topology of the catalogued component. In contrast to *Pipes* they cannot change their shape. *Fixes* are components to establish connections between pipes, i.e., elbow, tee, reducer or cross.
- *FixCluster* describes a set of connected *Fixes*. It is used to realize complex systems of *Fixes*.
- *FixGroup* is a subset contained by *FixCluster*. It consists of multiple connected *Fixes*. They should not overlap and are used to optimize the *FixClusters*.
- *PipeRow* is a set of components (pipes and fixes) that posses opposite oriented ConnectionPoints, i.e., opposite oriented line segments *ConnectionPoints*. This means that *fixes* contained by a *PipeRow* also have parallel or collinear *ConnectionPoints*, for example reducers, ex-centric reducers, valves. In certain steps of the algorithm the components of a *PipeRow* are processed as one component.

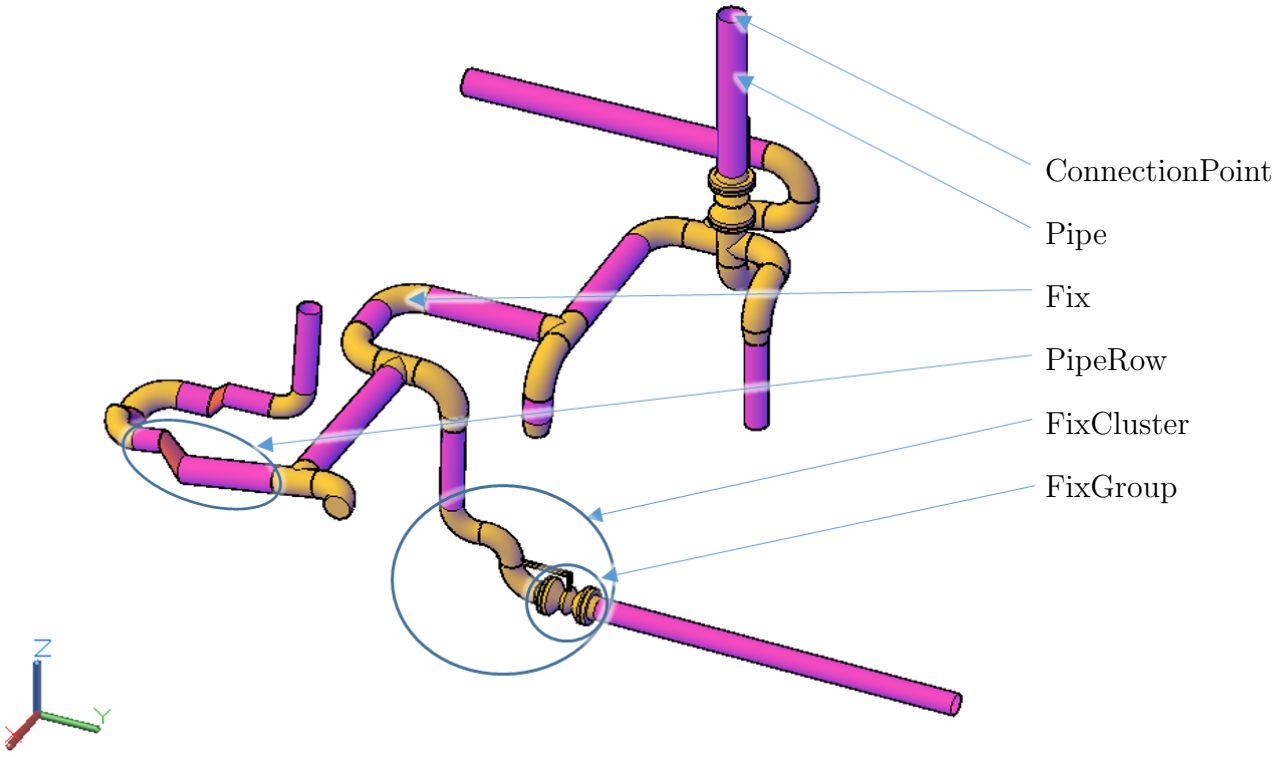

Figure 4: Terms illustrated by an example.

These terms are visualized with the help of an example in Figure 4. In the first step of the algorithm the FixClusters will be optimized in themselves towards the axes of the incident pipes. The pipe axes are left unaltered in this step of the calculation. After this preprocessing step, the main approach is to:

1. Align the pipes and pipe rows towards the FixClusters
2. Only to adjust and optimize the FixClusters afterwards again towards the incident pipe axes.

   Iteratively repeat this process until error is below an epsilon threshold.

This alternating alignment converges towards an improved global result. It is taken advantage of the fact, that there is always exactly one Pipe or PipeRow in between two FixClusters.

The number one stopping condition is the maximum deviation error at the ConnectionPoints in terms of delta movement and delta angle, which means collinear opposite direction vectors of the ConnectionPoints.

For example in *Autodesk AutoCAD Plant 3D 2015* you will need a default precision of an epsilon 10^{-10} in AutoCAD drawing units. Else, you will be shown those imperfections in the model with small images of water drops at the connection point locations. In newer versions of *Autodesk AutoCAD Plant 3D* the user is able to adjust this epsilon threshold to his needs. So the epsilon threshold depends on the drawing size, the used drawing unit and the users needs.

Actually there is also an *explosion* indicator checked after each iteration, which stops and restarts the calculation with a changed net topology. If it should happen, the connection with the highest error impact is cut.

Additionally, there is an iteration counter, which stops the algorithm when a certain iteration count threshold is reached. At the moment its default value is 4000 iterations. It

ensures, that the calculation does not take forever, but quits at some point and shows the result with all errors. So the user is able to identify error sources.

The following concepts are also applied:

1. The order independence during the calculation of the piping components has to be ensured. This means the calculation of the mean dual quaternion transformations from all separate transformations of the ConnectionPoints of a component.

2. A minimal length of the pipes has to be enforced. This neglects pipe *flips* of the ConnectionPoints. The minimum pipe length should just be bigger than zero. We use a tenth of the pipe radius.

3. An additional *up vector* is introduced for fixes with collinear ConnectionPoints and has to be re-established after each step of the algorithm. Thus unwanted rotations of valves around their own axes are cancelled out.

4. After the calculation of the rotation and translation in each iterative step, it is checked weather the resulting movement of the components ConnectionPoints exceed a certain threshold. If the threshold is reached, the complete transformation is limited towards the set threshold. This way, the calculated transformation direction is preserved and the adjacent piping components gain a chance to adjust themselves towards the change. Doing so prevents *explosions* and the convergence is ensured.

5. Pipe movements are stronger limited in their movement than the fixes using the technique of point 4 above. This is because the pipe axes have really been measured in the point cloud and should be better preserved. At the start of our approach we assume the pipes have been fitted to the point cloud precisely. Therefore, the pipe initial axis represent the measured position and orientation in the point cloud data. During the computation of the algorithm the access to the raw point cloud point data is not suitable, because the underlying CAD system takes too long for this.

6. Longer pipes will be more limited in their movement than short pipes.

7. To ensure a high accuracy, we perform the calculation with a much more precise data type than the data that is used to save the result. In the applied system we use the *quadruple* data type for the calculation and *double* to store the result in the CAD system. To increase the accuracy and speed during the calculation of the transformations further, we use dual quaternions. Especially the usage of trigonometric functions like sinus, cosines, tangents and their arcus counterparts are minimized this way. As we already said, the dual quaternions enable an efficient way to calculate mean transformations and replace the commonly used 4×4 matrices.

Another application is to allow the user to fix the position of some pipes and fixes, to adjust the piping system afterwards to the fixed components. This is especially important for working with components, which have been located using tie-in-points. They are exactly placed in the coordinate system and should be unaltered in terms of position and rotation after applying the described algorithm.

If there are inconsistencies in the system because of too much *fixed* components and the conflicts can not be resolved using the present degrees of freedom in the piping system, an automatic splitting of the piping topology and marking of the split is applied. This is done at all connections, which have the highest error impact in the piping system, until a consistent model is achieved. Thus, the sources of errors are isolated and displayed.

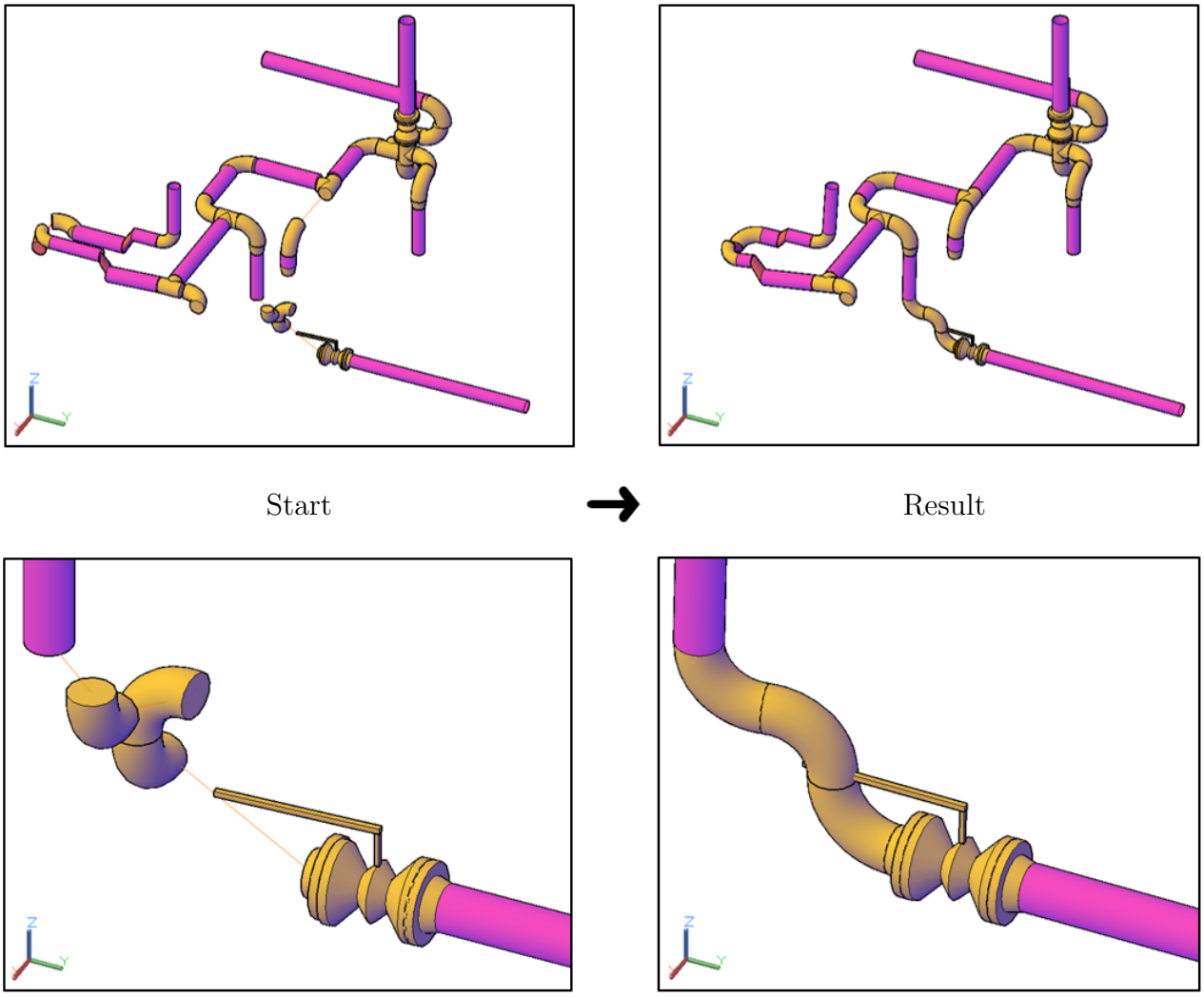

Figure 5: Complex piping system alignment.

4. Conclusion

We proposed an alignment algorithm for constrained pipe systems based on dual quaternion calculus. In practise the procedure has proven to be effective to solve this kind of problems (see Figure 5). Closed cycles and fixed components can be taken into account as well as all kinds of imaginable components, i.e., tees, reducers, elbows and so on. Since we are dealing with minimal displacements the resulting pipe system corresponds to the original point cloud. Moreover, this algorithm may be applied to other alignment problems for example to align wooden or steal beams, cable trays or ducts to each other. One additional idea is to combine the different domains in the same building. There are approaches to align and cluster walls in floor plans [11]. So the piping and duct system might follow the orientation of the built walls or vice versa.

References

[1] W. Blaschke: *Kinematik und Quaternionen*. Deutscher Verlag der Wissenschaften, Berlin 1960.

[2] O. Bottema, B. Roth: *Theoretical kinematics*. Dover Publ, New York 1990.

[3] Q.J. Ge, B. Ravani: *Geometric construction of Bézier motions*. J. Mech. Des. **116**/3, 749–755 (1994).

[4] M. Husty, A. Karger, H. Sachs, W. Steinhilper: *Kinematik und Robotik*. Springer, Berlin 1997.

[5] D. Klawitter: *Clifford Algebras – Geometric Modelling and Chain Geometries with Application in Kinematics*. PhD thesis, Technical University of Dresden, 2015.

[6] J.M. McCarthy, S. Ahlers: *Dimensional Synthesis of Robots using a Double Quaternion*. 9th International Symposium of Robotics Research 1999, ISRR'99, pp. 1–6.

[7] J.M. McCarthy: *An Introduction to Theoretical Kinematics*. MIT Press, 1990.

[8] H. Pottmann, J. Wallner: *Computational Line Geometry*. Springer, Berlin and New York 2001.

[9] B. Ravani, K. Sprott: *Kinematic Generation of Ruled Surfaces*. Adv. Comput. Math. **17**, 115–133 (2002).

[10] E. Study: *Geometrie der Dynamen*. B.G. Teubner, 1903.

[11] Ch. Tonn, O. Bringmann: *VPoint Clouds to BIM: Methods for Building Parts Fitting in Laser Scan Data*. In electronic Proceedings 16th International Conference CAAD Futures 2015, Sao Paulo/Brazil 2015, *The next city – New technologies and the future of the built environment*, pp. 358–369.

[12] P. Zsombor-Murray, S. El Fashny: *A Cylinder of Revolution on Five Points*. J. Geometry Graphics **10**/2, 207–213 (2006).

Received August 5, 2016; final form June 6, 2017