Plane Space Representation in Context of Mode-based Symmetry Plane Detection

Lukáš Hruda[1], Ivana Kolingerová[1], and Miroslav Lávička[2]

[1] University of West Bohemia, Faculty of Applied Sciences, Dept. of Computer Science and Engineering, Pilsen, Czech Republic
[2] University of West Bohemia, Faculty of Applied Sciences, Dept. of Mathematics and NTIS, Pilsen, Czech Republic

Abstract. This paper describes various representations of the space of planes. The main focus is on the plane space representation in the symmetry plane detection in E^3 where many candidate planes for many pairs of points of the given object are created and then the most often candidate is found as a mode in the candidate space, so-called Mode-based approach. The result depends on the representation used in the mode-seeking process. The most important aspect is how well distances in the space correspond to similarities of the actual planes with respect to the input object. So, we describe various usable distance functions and compare them both theoretically and practically. The results suggest that, when using the Mode-based approach, representing planes by reflection transformations is the best way but other simpler representations are applicable as well. On the other hand, representations using 3D dual spaces are not very appropriate. Furthermore, we introduce a novel way of representing the reflection transformations using dual quaternions.

Keywords: Space of Planes · Symmetry · Distance Function · Mode.

1 Introduction and Related Work

Symmetry detection in 3D objects is a very large and progressive field with many possible applications, mainly in computer graphics or computer vision, such as object alignment, compression or reconstruction of incomplete objects. One very popular approach to symmetry detection is to create a number of candidate transformations by matching different points or parts of the input object and then finding those transformations that occur most often in the transformation space (see e.g. [13]). This can also be described as seeking modes in the transformation space, so the approach is called Mode-based symmetry detection. It can be applied to detect symmetries of various types, however, in this paper we only focus on the detection of the planes of symmetry (reflectional symmetries) of 3D objects.

Mitra et al. [13] and Shi et al. [17] used the Mode-based approach to find quite general partial (or local) symmetries – the transformations can contain rotation with or without reflection, translation and uniform scaling or any subgroup of these transformations including pure reflections (symmetry planes).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

To find the modes these methods employ Mean shift clustering algorithm [3]. Li et al. [11] used the same approach to detect symmetry planes of damaged skulls. Other uses for symmetry plane detection are in [12, 9]. Similar approach has recently been used by Hruda et al. in rigid surface registration [8] which can be understood as symmetry detection between two objects. The candidate space contained rigid transformations and a mode was found by a density peak estimation algorithm. This could also be utilized to find the global plane of symmetry. A related approach was used by Podolak et al. [16] and Caillière et al. [2] where Hough transform and voting are employed to find the symmetry planes. The space of planes was divided into non-uniform discrete bins to count plane occurrences. Regardless of the specific algorithm, any Mode-based method for symmetry plane detection requires to define some representation of the space of planes, the definition influences the result. The important aspect is to have distances between points in the space well corresponding to the actual similarity/dissimilarity of the planes in E^3. The mode(s) can be found in an arbitrary non-Euclidean space only using distances between the points, their coordinates are not needed [18, 8]. Proximity queries in non-Euclidean spaces can be accelerated using the Vantage Point Tree data structure [19] as done in [8]. In context of symmetry detection, planes can be understood as transformations reflecting points over the given plane. The problem of computing distances between rigid transformations has been analyzed in [8], however, the same problem does not seem to be sufficiently addressed in the literature for reflection transformations or planes, in spite of the popularity of the Mode-based approach.

This paper describes and analyzes several different representations of the space of planes. For each representation, reasonable distance functions, suitable in Mode-based symmetry plane detection, are discussed. The distance functions are compared to the distance function closest to the ground truth but useless in practice due to its large computation cost. The information about the described representations can be useful in other applications where a plane representation is needed, although the presented distance functions might require some application-based adjustment. We thus believe that researchers from various fields, not restricted to symmetry plane detection, could benefit from this paper.

2 Background

A general plane P can be defined by its implicit equation as $P : ax+by+cz+d = 0$ where $\mathbf{n} = [a, b, c]^T$ is the normal vector of the plane. We always consider the coefficients to be normalized such that $\|\mathbf{n}\| = 1$ in which case d represents the signed distance of the plane from the origin. A function $\mathbf{r}_P(\mathbf{x}) \in E^3$ that reflects an arbitrary point $\mathbf{x} \in E^3$ over the plane P can be defined as shown in Eq. (1).

$$\mathbf{r}_P(\mathbf{x}) = \mathbf{x} - 2(\mathbf{n}^T\mathbf{x} + d)\mathbf{n} \tag{1}$$

2.1 Candidate Creation Algorithm

Let us have a set of input points representing the object for which the set of candidate symmetry planes is to be found: $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_i \in E^3, i =$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

$1, 2, \ldots, N$. We use the point set representation due to its generality. We first create a 3D uniform grid with cell size $\frac{l_{avrg}}{\delta} \times \frac{l_{avrg}}{\delta} \times \frac{l_{avrg}}{\delta}$ where l_{avrg} is the estimated size of the object computed as the average distance of the points of X from their centroid. We mark each cell as either occupied if any point from X falls into it, otherwise unoccupied. Then we start randomly selecting pairs of points $\mathbf{x}_i, \mathbf{x}_j$ from X and construct a plane P such that $\mathbf{r}_P(\mathbf{x}_i) = \mathbf{x}_j$. To avoid clutter in the candidate space we perform a quick check to determine whether P is a plausible candidate by randomly selecting another five points from X, reflecting them over P and checking whether all of them end up in an occupied cell of the previously created grid. If they do P is accepted as a candidate. If at least one of them reflects into an unoccupied cell then P is rejected. We keep iterating this process until we have k accepted candidates and if not stated otherwise we set $\delta = 5$, $k = 2000$. The key idea behind the Mode-based approach is that now there should be significant modes in the candidate space of planes corresponding to the strongest symmetries of the input point set X.

2.2 Dependence on Scale and Position

The a, b, c coefficients are bounded on finite interval $\langle -1; 1 \rangle$. The value of the d coefficient of any candidate plane depends on the position and overall scale of the input object because d represents the distance of the given candidate plane from the origin and if the size of the object changes, the span of the d coefficient will change as well. However, the a, b, c coefficients will stay the same.

The dependence on the position is less obvious. If we translate the input object (all points in X) by some arbitrary vector $\mathbf{t}$, then for an arbitrary candidate plane P, d will change by $\mathbf{t}^T\mathbf{n}$ against the original position. As the change of d depends also on the orientation of the given plane, the change of d is inconsistent throughout the candidate planes and this inconsistency grows with distance of the input object from the origin. Therefore, the position of the input object influences the span of d but does not influence the span of a, b, c.

Generally, the distance functions for planes are negatively influenced by significantly different span of d and a, b, c, therefore, we always translate the input object's centroid into the origin and, if necessary, we also normalize d by l_{avrg} to make the spans of d and of a, b, c similar. For those distance functions where the translation to origin is not necessary, this fact will be pointed out explicitly.

2.3 Ground Truth

As pointed out in [8], distance between transformations cannot be well defined without the context (the object on which the transformations are applied), which is consistent with Section 2.2. Therefore, the most meaningful distance function for planes is the one used for error evaluation of registration results in [8], only with reflection transformations instead of rigid ones. Given two arbitrary planes P_1 and P_2, the distance function measures the exact difference between the effects of two reflections defined by P_1 and P_2 on the input object. It will be

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

considered the ground truth distance function, denoted $D_{GT}(P_1, P_2)$ and defined

$$D_{GT}(P_1, P_2) = \sum_{i=1}^{N} \|\mathbf{r}_{P_1}(\mathbf{x}_i) - \mathbf{r}_{P_2}(\mathbf{x}_i)\|, \quad \text{where} \quad \mathbf{x}_i \in X, i = 1, \ldots, N. \quad (2)$$

The D_{GT} distance function is not affected by the position of the input object, so it does not require the translation to origin, and the object size only effects its overall scale. Unfortunately, the time complexity of computing D_{GT} is $O(N)$ where N is the point count of the input object, which makes it too computationally expensive and, therefore, virtually unusable in any Mode-based symmetry detection algorithm. However, we can use it to compare other distance functions.

3 Plane Space Representations

In this section we describe and visualize various representations of the space of planes in E^3 plus possible distance functions usable in an arbitrary Mode-based symmetry plane detection algorithm. We use the algorithm from Section 2.1 to create a set of candidate symmetry planes of the object shown in Figure 1. The black line in the figure represents the correct symmetry plane, the object is rotated to have this plane perpendicular to the projection. We purposely selected a slightly asymmetrical object. Although the object is visualized as a triangle mesh, only its vertices are used to compute the candidates.

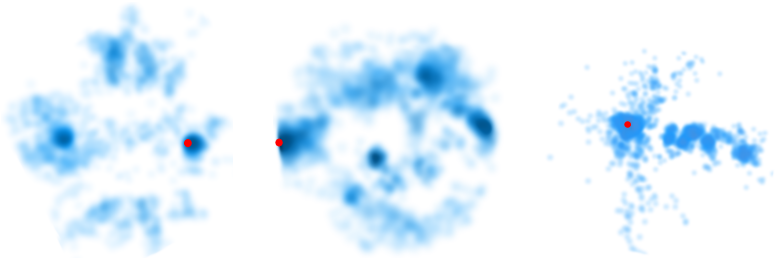
Fig. 1. Model object with its correct symmetry plane

3.1 Dual Representation in E^3

The implicit equation of a plane has four coefficients but there are only three degrees of freedom when defining a plane because the space of planes is a 3-dimensional manifold embedded in 4-dimensional space. We can thus use a dual representation of any plane as a point in E^3. We denote $\rho(P) \in E^3$ a dual representation of a plane P. Euclidean metric can then be used to compute the distance between two planes P_1, P_2 as $D_\rho(P_1, P_2) = \|\rho(P_1) - \rho(P_2)\|$.

One possible representation is to encode the plane orientation into a vector in E^3 with the same direction as the plane normal vector, and d into the length of

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38



(a) ρ_1 (b) ρ_2 (c) ρ_3

Fig. 2. Dual representations of the candidate symmetry planes. The colors represent density (the darker, the larger density), the red spot shows the correct symmetry plane.

this vector. Such dual representation can be defined as $\rho_1(P) = d\mathbf{n}$. Obviously, for $d \to 0$ there is ambiguity because such planes are shrunk into a single point. To solve this problem, the value of d is shifted by a constant μ, then these planes get spread on the surface of a sphere with radius μ instead of being all at the origin. We set $\mu = \frac{1}{2}l_{avrg}$ so that rotating the normal by π and changing d by l_{avrg} make approximately similar change in position of the point in the dual space. The dual representation is therefore finally defined as

$$\rho_1(P) = \begin{cases} (d + \frac{1}{2}l_{avrg})\mathbf{n} & d \geq 0 \\ (d - \frac{1}{2}l_{avrg})\mathbf{n} & d < 0 \end{cases}.$$

Distances in such dual space still do not very well correspond to similarities of the actual planes. Mainly, two planes with d close to 0 and similar normal vectors can be on the other sides of the sphere, and therefore more than 2μ apart, although they are actually very similar. However, such representation can be very good for visualization as each point in the dual space represents the plane quite intuitively. Figure 2a shows the generated candidates on the given model in the dual E^3 space transformed with ρ_1. The darker spots correspond to larger density of the points in the space, the red spot to the correct plane from Figure 1. The viewpoint was selected manually to maximize the information in the image. It can be seen that the correct plane is in a noticeable mode (dense spot) but this mode is split on the sphere surface corresponding to $d = 0$ and its non-negligible part is on the other side. This is undesirable because it makes the mode much less significant than it would be if the two parts were together.

Another duality, also called polar duality (described e.g. in [6]), uses normalization of the plane coefficients such that $d = 1$, then the a, b, c coefficients are used as coordinates in E^3 i.e $\frac{1}{d}\mathbf{n}$. This again poses a problem for $d \to 0$ which makes the dual points approach infinity. We solve this issue in the same way as with ρ_1 by shifting the d coefficient and we define this dual representation as

$$\rho_2(P) = \begin{cases} \frac{1}{(d+\frac{1}{2}l_{avrg})}\mathbf{n} & d \geq 0 \\ \frac{1}{(d-\frac{1}{2}l_{avrg})}\mathbf{n} & d < 0 \end{cases}.$$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

Figure 2b shows the candidates transformed by ρ_2 into the dual space. The right plane is in a noticeable mode, split again into two separate parts very far apart. There are also other significant modes corresponding to very different planes.

Another duality commonly used in computational geometry expresses a plane using its coefficients in explicit representation [1]. There are three possible explicit representations of a plane in E^3:

$$x = -\frac{b}{a}y - \frac{c}{a}z - \frac{d}{a}, \quad y = -\frac{a}{b}x - \frac{c}{b}z - \frac{d}{b}, \quad z = -\frac{a}{c}x - \frac{b}{c}y - \frac{d}{c}.$$

For demonstration, we select the first one, the dual representation is then defined as $\rho_3(P) = [\frac{b}{a}, \frac{c}{a}, \frac{d}{l_{avrg} \cdot a}]$. The division of d by l_{avrg} is necessary to normalize the span of d. Such duality obviously cannot represent planes parallel to the x-axis and planes with $a \to 0$ approach infinity in the dual space. We could solve this by shifting a but this time, we do not include l_{avrg} into the shift because the span of a does not depend on the size of the input object, so we get

$$\rho_3(P) = \begin{cases} [\frac{b}{a+\frac{1}{2}}, \frac{c}{a+\frac{1}{2}}, \frac{d}{l_{avrg}(a+\frac{1}{2})}] & a \geq 0 \\ [\frac{b}{a-\frac{1}{2}}, \frac{c}{a-\frac{1}{2}}, \frac{d}{l_{avrg}(a-\frac{1}{2})}] & a < 0 \end{cases}.$$

Figure 2c shows the candidates in the dual space transformed by ρ_3 and in this case there do not seem to be any significant modes.

In general, the dual representations appear not very appropriate for representing planes in Mode-based symmetry detection due to their singularities. This problem can be solved by shifting the value of some coefficient by a constant but the choice of this constant is rather arbitrary and even then the distances between points in the dual space might not well correspond to similarities of the planes. However, the dual representations can be useful to visualize the candidates as the dual points are 3-dimensional.

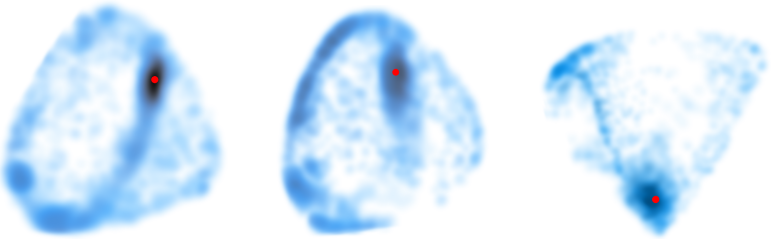
3.2 4D Vector Representation

Probably the most intuitive way of representing a plane is by a 4D vector of the plane coefficients. Given a plane P we represent it by a vector $\mathbf{p} = [a, b, c, \frac{d}{l_{avrg}}]^T$. In such a space we can easily define a distance function as the Euclidean distance of the two 4D vectors. However, $\mathbf{p}$ and $-\mathbf{p}$ represent the same plane so we need to take this into account. The Euclidean distance function is then defined as

$$D_{ED}(P_1, P_2) = \begin{cases} \|\mathbf{p}_1 - \mathbf{p}_2\| & \mathbf{p}_1^T \mathbf{p}_2 \geq 0 \\ \|\mathbf{p}_1 + \mathbf{p}_2\| & \mathbf{p}_1^T \mathbf{p}_2 < 0 \end{cases}.$$

In this case the points cannot be visualized directly, so we use the multidimensional scaling (MDS) technique to transform the points into E^3 while maintaining their distances, w.r.t. the given distance function. However, the projection into E^3 might cause some imprecision in the visualization. Figure 3a shows the candidate planes projected into E^3 with MDS using the D_{ED} distance function and there is a very significant mode visible around the correct symmetry plane.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38



(a) D_{ED} (b) D_{AD} (c) D_{ACD}

Fig. 3. The candidates represented by 4D vectors projected into E^3 with MDS using different distance functions.

The distances in 4D vector space of planes can also be measured as angles between the vectors because the length of the vector $\mathbf{p}$ does not influence the plane P it represents. The angle distance function can be defined as

$$D_{AD}(P_1, P_2) = \arccos\left(\frac{|\mathbf{p}_1^T \mathbf{p}_2|}{\|\mathbf{p}_1\|\|\mathbf{p}_2\|}\right).$$

Figure 3b shows the candidates after using MDS with the D_{AD} distance function and the correct plane is again placed inside a noticeable mode.

We can also use only the cosine of the angle and measure its deviation from 1. The angle cosine distance function can be defined as

$$D_{ACD}(P_1, P_2) = 1 - \frac{|\mathbf{p}_1^T \mathbf{p}_2|}{\|\mathbf{p}_1\|\|\mathbf{p}_2\|}$$

and its visualization using MDS is shown in Figure 3c. There is again a noticeable mode around the correct plane.

The 4D representation of the plane space is more appropriate for any Mode-based symmetry detection algorithm than the dual representations in E^3. However, they are not as convenient for visualization as the points first need to be projected into a lower dimensional space which causes a loss of information.

3.3 Transformation Representation

As already mentioned, the distance between arbitrary two planes P_1 and P_2 can be defined as the distance between the two reflection transformations $\mathbf{r}_{P_1}$ and $\mathbf{r}_{P_2}$ defined according to Eq (1). One way of doing this is using the compound metric evaluated as the most suitable for rigid transformations in [8]. It is based on sum of squared distances between the transformed points and is defined as

$$D_{SSD}(P_1, P_2) = \sqrt{\sum_{i=1}^{N} \|\mathbf{r}_{P_1}(\mathbf{x}_i) - \mathbf{r}_{P_2}(\mathbf{x}_i)\|^2}$$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

(a) D_{SSD} (b) D_{DQ}

Fig. 4. The candidates represented as transformations projected into E^3 with MDS using different distance functions.

where $\mathbf{x}_i \in X, i = 1, \ldots, N$. Certain similarity between D_{SSD} and the ground truth distance function D_{GT} (see Eq. (2)) can be noticed with two major differences. First, D_{SSD} uses squared distances, favouring smaller displacements over larger ones, which leads to different distances. Second, unlike D_{GT}, D_{SSD} can be computed in $O(1)$ with $O(N)$ preprocessing [8]. The transformations must be expressed as $\mathbf{Mx} + \mathbf{t}$ where $\mathbf{M}$ is an orthogonal transformation matrix, $\mathbf{t}$ is an arbitrary translation vector and $\mathbf{x}$ is the transformed point. From Eq. (1) we get

$$\mathbf{r}_P(\mathbf{x}) = \mathbf{x} - 2\mathbf{n}\mathbf{n}^T\mathbf{x} - 2d\mathbf{n} = (\mathbf{I} - 2\mathbf{n}\mathbf{n}^T)\mathbf{x} - 2d\mathbf{n}$$

where $\mathbf{I}$ is the identity matrix. If we denote $\mathbf{M} = (\mathbf{I} - 2\mathbf{n}\mathbf{n}^T)$ and $\mathbf{t} = -2d\mathbf{n}$, then the reflection is $\mathbf{r}_P(\mathbf{x}) = \mathbf{Mx} + \mathbf{t}$. As $\mathbf{M}$ is orthogonal (and symmetric), we can use the same approach as in [8] to compute D_{SSD} in $O(1)$ with $O(N)$ preprocessing. The D_{SSD} distance function, as well as D_{GT}, is not affected by the position of the input object, so the translation to the origin is not required, and the object size only effects the overall scale of the distance function.

Figure 4a shows the candidates projected into E^3 using MDS with the D_{SSD} distance function and the correct plane is again in a significant mode. There is another smaller significant mode visible in the figure, however, this can very likely be caused by the distortion of the MDS projection.

Dual Quaternions Dual quaternions combine the concepts of quaternions and dual numbers. Let us show how to use them to represent a reflection over an arbitrary plane. A general quaternion is defined as $Q = q_0 + q_1 i + q_2 j + q_3 k$ where the i, j, k units multiply according to the following rules

$$i^2 = j^2 = k^2 = ijk = -1, ij = k = -ji, jk = i = -kj, ki = j = -ik.$$

A conjugate Q^* of a quaternion Q is defined as $Q^* = q_0 - q_1 i - q_2 j - q_3 k$. We denote $v(\mathbf{x}) = xi + yj + zk$ a quaternion that represents an arbitrary point $\mathbf{x} = [x, y, z]^T \in E^3$. If $\mathbf{u}$ is an arbitrary unit vector and we set $Q = \cos\alpha + v(\mathbf{u})\sin\alpha$, then $Qv(\mathbf{x})Q^*$ represents the point $\mathbf{x}$ rotated by angle 2α around the axis that passes through the origin and has the direction of $\mathbf{u}$. Similarly, if we set $Q = v(\mathbf{u})$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

then $Qv(\mathbf{x})Q$ represents the point $\mathbf{x}$ reflected over the plane with normal $\mathbf{u}$ that passes through the origin. For details about quaternions we refer to [7].

A dual quaternion, for more detail see e.g. [15], is defined as

$$Q_d = Q + \epsilon Q_\epsilon = q_0 + q_1 i + q_2 j + q_3 k + \epsilon(q_{\epsilon 0} + q_{\epsilon 1} i + q_{\epsilon 2} j + q_{\epsilon 3} k)$$

where Q and Q_ϵ are quaternions and ϵ is the dual unit which commutes with the quaternion units i, j, k and it is that $\epsilon^2 = 0$. A quaternion conjugate of Q_d is defined as $Q_d^* = Q^* + \epsilon Q_\epsilon^*$, a dual conjugate of Q_d is defined as $\overline{Q_d} = Q - \epsilon Q_\epsilon$. These conjugations can be combined into $\overline{Q_d^*} = Q^* - \epsilon Q_\epsilon^*$.

We denote $v_d(\mathbf{x}) = 1 + \epsilon(xi + yj + zk)$ a dual quaternion representing an arbitrary point $\mathbf{x} = [x, y, z]^T \in E^3$. If Q is a quaternion that represents rotation and $Q_\epsilon = \frac{v(\mathbf{t})Q}{2}$ where $\mathbf{t} = [t_x, t_y, t_z]^T$ is an arbitrary translation vector then for $Q_d = Q + \epsilon Q_\epsilon$, using the rules of dual quaternion algebra, it can be shown that

$$Q_d v_d(\mathbf{x})\overline{Q_d^*} = 1 + \epsilon(Qv(\mathbf{x})Q^* + v(\mathbf{t}))$$

which represents the point $\mathbf{x}$ rotated via Q and then translated by $\mathbf{t}$. This shows how to use dual quaternions in connection with rigid transformations. Note that Q_d represents the same transformation as $-Q_d$ with the identity being represented by either 1 or -1. The transformations can be concatenated by multiplying the corresponding dual quaternions and if Q_d represents a rigid transformation, Q_d^* represents its inverse. Next, for Q_{d1}, Q_{d2} representing rigid transformations, these transformations are the same only if $Q_{d1}Q_{d2}^* = 1$ or $Q_{d1}Q_{d2}^* = -1$.

Consider now a plane P and a dual quaternion $Q_d = Q + \epsilon Q_\epsilon$ defined such that $Q = v(\mathbf{n})$ and $Q_\epsilon = \frac{v(\mathbf{t})Q}{2}$ where $\mathbf{t} = -2d\mathbf{n}$. Now Q_d represents a transformation that first rotates by π around the axis that passes through the origin and has the direction of $\mathbf{n}$, and then translates by $-2d\mathbf{n}$. However, if we apply the transformation on $-\mathbf{x}$ instead of $\mathbf{x}$, it can be shown that we will get

$$Q_d v_d(-\mathbf{x})\overline{Q_d^*} = 1+\epsilon(Qv(\mathbf{x})Q+v(\mathbf{t})) = 1+\epsilon(v(\mathbf{n})v(\mathbf{x})v(\mathbf{n})-v(2d\mathbf{n})) = v_d(\mathbf{r}_P(\mathbf{x}))$$

which exactly represents $\mathbf{r}_P(\mathbf{x})$. This shows that a dual quaternion can also represent a reflection transformation by representing a rigid transformation that transforms $-\mathbf{x}$ to $\mathbf{r}_P(\mathbf{x})$. Therefore, to measure distances between reflection transformations we can use a distance function for dual quaternions.

We denote $vec(Q_d) = [q_0, q_1, q_2, q_3, q_{\epsilon 0}, q_{\epsilon 1}, q_{\epsilon 2}, q_{\epsilon 3}]^T \in E^8$ an 8-dimensional vector that is equivalent to Q_d. Given a plane P, we create the corresponding dual quaternion Q_d such that $Q = v(\mathbf{n})$ and $Q_\epsilon = \frac{v(-2d\mathbf{n})Q}{2l_{avrg}}$, i.e. $Q_d = v(\mathbf{n}) + \epsilon\frac{v(-2d\mathbf{n})v(\mathbf{n})}{2l_{avrg}}$. The division by l_{avrg} is to normalize the translation part. Using the algebra of dual quaternions we can actually get that $Q_\epsilon = \frac{d}{l_{avrg}}$, so Q_d can be finally expressed as shown in Eq. (3).

$$Q_d = v(\mathbf{n}) + \epsilon\frac{d}{l_{avrg}} = ai + bj + ck + \epsilon\frac{d}{l_{avrg}} \tag{3}$$

There are two common distance functions for dual quaternions. The first one uses differences between the equivalent 8-dimensional vectors [15]. Suppose

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

two arbitrary planes P_1 and P_2 represented by dual quaternions Q_{d1} and Q_{d2} respectively. Such distance function can be defined as

$$\min\{\|vec(Q_{d1}) - vec(Q_{d2})\|, \|vec(Q_{d1}) + vec(Q_{d2})\|\}$$

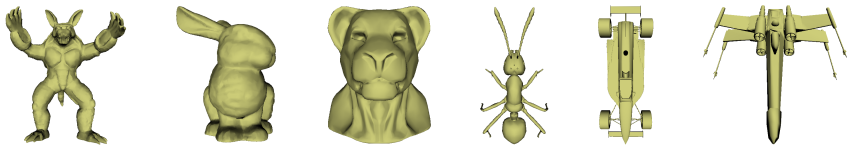
but given Eq. (3) this is exactly the same as D_{ED}. The second distance function [5] uses a difference transformation $Q_{d1}Q^*_{d2}$ and computes its distance from the identity, i.e. from 1 or -1. It is defined as

$$D_{DQ}(P_1, P_2) = \min\{\|vec(1 - Q_{d1}Q^*_{d2})\|, \|vec(1 + Q_{d1}Q^*_{d2})\|\}.$$

Figure 4b shows the candidates projected into E^3 using MDS with D_{DQ} and the correct plane is in an obvious mode.

4 Results

We compared the distance functions by generating the candidate symmetry planes of a given object (using the model algorithm from Section 2.1), comparing values of the given distance function and of the ground truth one. We did this for the six test objects from Figure 5, taken from datasets [10, 4]. The objects are represented by triangle meshes for easier visualization, but we again only used their vertices as the input points for the candidate creation process.



(a) Armadillo 10026 | (b) Bunny 9831 | (c) Lion 2213 | (d) Ant 3495 | (e) Formula 10969 | (f) Space ship 3099

Fig. 5. The test objects used to generate the candidate sets for comparing the distance functions. The number under the name of each object expresses its point count.

Let $C = \{P_1, P_2, ..., P_k\}, k = 2000$ be the set of candidate planes created for a given input object. The error of a given distance function D against the ground truth is defined as

$$Err(D) = \frac{1}{Count(k)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \left| \frac{D_{GT}(P_i, P_j)}{Avrg(D_{GT})} - \frac{D(P_i, P_j)}{Avrg(D)} \right|$$

where

$$Avrg(D) = \frac{1}{Count(k)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} D(P_i, P_j)$$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

is the average distance between candidates in C and $Count(k) = \frac{1}{2}(k^2 - k)$ is the total number of candidate pairs used for the computation. The normalization by $Avrg$ is used because the overall scales of the distance functions do not matter so the differences are computed after both D_{GT} and D are divided by their mean values. Table 1 shows the errors of all the distance functions described above for all the test objects. We include the dual representations in the comparison. The smallest error is obviously achieved using D_{SSD} probably due to the same principle of D_{SSD} and D_{GT}. It is still rather surprising that the D_{SSD} function which uses squared distances is so similar to D_{GT} that uses absolute distances. The D_{ED}, D_{AD} and D_{DQ} all have very similar errors (D_{DQ} usually has the lowest error) which are overall lower than those of D_{ACD} and the distances in the dual spaces, but in case of D_{ACD} this can be explained by its resemblance to the cosine function ($D_{ACD} = 1 - \cos(D_{AD})$). The function D_{ρ_3} exhibits similar or lower error than D_{ACD} on some objects (Arm, Ant) but also considerably larger error on different ones (For, Shi) which suggests that D_{ρ_3} is quite unpredictable.

Table 1. Errors of the distance functions for the candidate sets for different objects.

	Arm	Bun	Ant	For	Lio	Shi	Average
D_{ED}	0.120	0.277	0.163	0.093	0.130	0.234	0.169
D_{AD}	0.133	0.281	0.157	0.098	0.144	0.236	0.174
D_{ACD}	0.299	0.388	0.264	0.250	0.306	0.352	0.309
D_{SSD}	0.012	0.023	0.009	0.014	0.011	0.012	0.013
D_{DQ}	0.118	0.277	0.162	0.093	0.129	0.232	0.168
D_{ρ_1}	0.382	0.399	0.503	0.596	0.326	0.425	0.438
D_{ρ_2}	0.401	0.408	0.488	0.563	0.360	0.489	0.451
D_{ρ_3}	0.280	0.446	0.269	0.730	0.362	0.447	0.422

The graphs in Figure 6 show the relation between D_{GT} and the other distance functions. We generated 50 candidates on the Armadillo and for each pair of the candidates we put its distance computed by D_{GT} on the horizontal axis and the distance computed by a given different distance function on the vertical axis. We normalize each value by $Avrg$. If some distance function D was the same as D_{GT} (apart from overall scale) there would be a perfect linear dependency and the points would lie on a line in the graph. Fig. 6a shows the relations of D_{SSD}, D_{ED}, D_{ACD} to D_{GT}. The similarity of D_{AD}, D_{DQ}, D_{ED} is shown in Fig. 6b, Fig. 6c shows the dual representations. For different objects the graphs are slightly different but very similar. There is an obvious almost linear dependency between D_{GT} and D_{SSD} (see Fig. 6a), however, D_{ED}, D_{AD}, D_{DQ} exhibit relation to D_{GT} that is also nearly linear (see Fig. 6b). For D_{ACD} the resemblance to cosine is visible. The dual representations show rather unstable behavior (see Fig. 6c). This is mostly caused by the shift in some of their coordinates but without it the dual representations would suffer from singularities which would make them unusable. Different shifting constant could lead to better results, at least for some objects, but there is no general way how to choose it.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

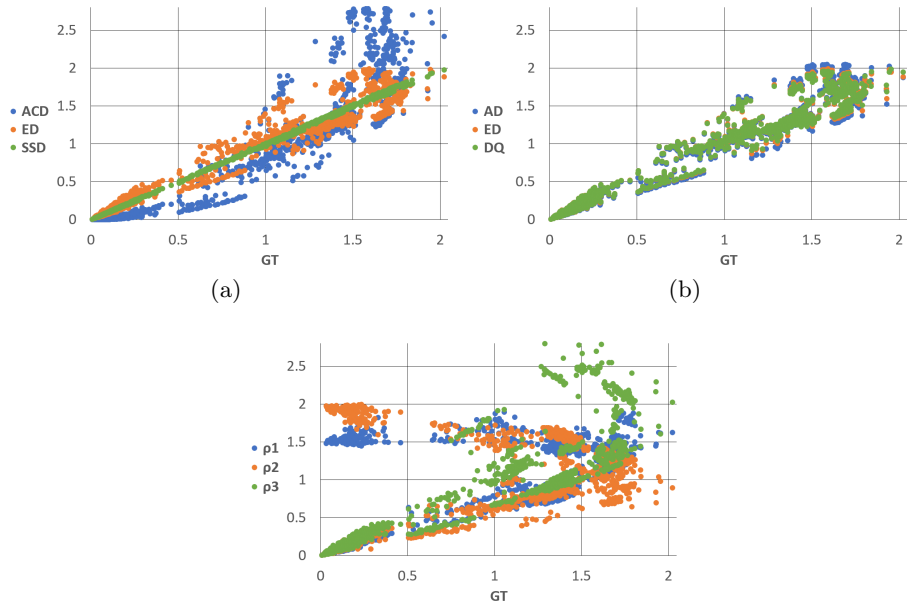


Fig. 6. Relations between a) $D_{SSD}/D_{ED}/D_{ACD}$ and D_{GT} ; b) $D_{DQ}/D_{ED}/D_{AD}$ and D_{GT} ; c) $D_{\rho_1}/D_{\rho_2}/D_{\rho_3}$ and D_{GT} for the Armadillo object.

Table 2 shows the Pearson correlations [14] between all pairs of the distance functions for the data from Figure 6. Value of 1 indicates perfect linear dependency and the closer to 0 the weaker the dependency. Expectedly, D_{SSD} shows the best linear correlation with D_{GT}. The correlations of D_{ED}, D_{AD}, D_{DQ} and even D_{ACD} with D_{GT} are all rather high. On the other hand, the distances in the dual spaces have mostly low correlation with D_{GT}. The high correlations among D_{ED}, D_{AD} and D_{DQ} confirm the similarity of these three distance functions.

Based on the results, the most appropriate representation of the space of planes in any Mode-based symmetry detection method is the transformation representation with the D_{SSD} distance function. But, except for the dual representations, all the distance functions are similar and none of them deviates significantly from D_{GT} making all of them well applicable. However, all the distance functions except D_{SSD} require translating the input object to the origin, otherwise the normalization of the d coefficient would have to be done differently.

4.1 Theoretical Comparison

There are some theoretical differences between the various representations. The dual and the 4D vector representations are basically Euclidean and the candidates can easily be stored in a data structure such as a KD-tree or a grid. In

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

Table 2. Pearson correlations of the distance functions for the Armadillo object.

	GT	ED	AD	ACD	SSD	DQ	ρ_1	ρ_2	ρ_3
GT	1.0000	0.9723	0.9644	0.9120	0.9998	0.9738	0.5361	0.3265	0.7238
ED	0.9723	1.0000	0.9989	0.9679	0.9713	0.9998	0.5537	0.3460	0.7621
AD	0.9644	0.9989	1.0000	0.9767	0.9635	0.9983	0.5499	0.3474	0.7548
ACD	0.9120	0.9679	0.9767	1.0000	0.9105	0.9664	0.5196	0.3111	0.7262
SSD	0.9998	0.9713	0.9635	0.9105	1.0000	0.9728	0.5351	0.3286	0.7214
DQ	0.9738	0.9998	0.9983	0.9664	0.9728	1.0000	0.5516	0.3423	0.7637
ρ_1	0.5361	0.5537	0.5499	0.5196	0.5351	0.5516	1.0000	0.9248	0.4217
ρ_2	0.3265	0.3460	0.3474	0.3111	0.3286	0.3423	0.9248	1.0000	0.1934
ρ_3	0.7238	0.7621	0.7548	0.7262	0.7214	0.7637	0.4217	0.1934	1.0000

case of the D_{AD} and D_{ACD} distance functions some structure can be built using the polar coordinates in 4D. Also, there are quite many possible algorithms for mode-seeking in Euclidean data. The transformation representations and the D_{SSD} and D_{DQ} distance functions are non-Euclidean, with smaller choice of data structures and possible mode seeking algorithms [18, 8, 19]. Also, the implementation of the D_{SSD} and D_{DQ} is more complex since they require implementing the matrix and the dual quaternion algebras.

Although the D_{DQ} distance function does not bring any considerable improvement over simpler distance functions, the idea of representing reflections by dual quaternions seems novel and can possibly find its use in other applications or new symmetry detection methods created in the future.

5 Conclusion

We described several representations of the space of planes suitable for any Mode-based algorithm for symmetry plane detection and computation of the distances in these representations. We showed that the 3D dual space representations are not very appropriate for this purpose but usable for visualizations. In order to represent the space of planes suitably in context of the Mode-based symmetry detection, spaces of higher dimensions must be used and the most appropriate representation is even non-Euclidean. However, the results suggest that apart from the 3D dual spaces all the plane space representations are well applicable in this context, although there are some theoretical differences between them.

Acknowledgement

This work was supported by Ministry of Education, Youth and Sports of the Czech Republic, project PUNTIS (LO1506) under the program NPU I and University specific research project SGS-2019-016 Synthesis and Analysis of Geometric and Computing Models.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38

References

1. Berg, de, M., Cheong, O., Kreveld, van, M., Overmars, M.: Computational geometry : algorithms and applications. Springer, Germany, 3rd edn. (2008)
2. Caillière, D., Denis, F., Pele, D., Baskurt, A.: 3d mirror symmetry detection using hough transform. In: Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on. pp. 1772–1775 (2008)
3. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence **24**(5), 603–619 (2002)
4. Fang, R., Godil, A., Li, X., Wagan, A.: A new shape benchmark for 3d object retrieval. In: International Symposium on Visual Computing. pp. 381–392. Springer (2008)
5. Figueredo, L., Adorno, B.V., Ishihara, J.Y., Borges, G.A.: Robust kinematic control of manipulator robots using dual quaternion representation. In: 2013 IEEE International Conference on Robotics and Automation. pp. 1949–1955 (2013)
6. Gallier, J.: Notes on convex sets, polytopes, polyhedra, combinatorial topology, voronoi diagrams and delaunay triangulations. arXiv:0805.0292 (2008)
7. Goldman, R.: Rethinking Quaternions: Theory and Computation. Morgan and Claypool Publishers (2010)
8. Hruda, L., Dvořák, J., Váša, L.: On evaluating consensus in ransac surface registration. In: Computer Graphics Forum. vol. 38, pp. 175–186 (2019)
9. Kroemer, O., Amor, H.B., Ewerton, M., Peters, J.: Point cloud completion using extrusions. In: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012). pp. 680–685 (2012)
10. Levoy, M., Gerth, J., Curless, B., Pull, K.: The stanford 3d scanning repository. http://www.graphics.stanford.edu/data/3Dscanrep/ (2005)
11. Li, X., Yin, Z., Wei, L., Wan, S., Yu, W., Li, M.: Symmetry and template guided completion of damaged skulls. Computers & Graphics **35**(4), 885–893 (2011)
12. McCrae, J., Singh, K., Mitra, N.J.: Slices: a shape-proxy based on planar sections. ACM Trans. Graph. **30**(6), 168 (2011)
13. Mitra, N.J., Guibas, L.J., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. In: ACM Transactions on Graphics (TOG). vol. 25, pp. 560–568. ACM (2006)
14. Mudelsee, M.: Estimating pearson's correlation coefficient with bootstrap confidence interval from serially dependent time series. Mathematical Geology **35**(6), 651–665 (2003)
15. Pham, H.L., Perdereau, V., Adorno, B.V., Fraisse, P.: Position and orientation control of robot manipulators using dual quaternion feedback. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 658–663 (2010)
16. Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., Funkhouser, T.: A planar-reflective symmetry transform for 3d shapes. ACM Transactions on Graphics (TOG) **25**(3), 549–559 (2006)
17. Shi, Z., Alliez, P., Desbrun, M., Bao, H., Huang, J.: Symmetry and orbit detection via lie-algebra voting. In: Computer Graphics Forum. vol. 35, pp. 217–227 (2016)
18. Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: European conference on computer vision. pp. 705–718 (2008)
19. Yianilos, P.N.: Data structures and algorithms for nearest neighbor search in general metric spaces. In: Soda. vol. 93, pp. 311–21 (1993)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_38