DETECTION OF PERFECT AND APPROXIMATE REFLECTIVE SYMMETRY IN ARBITRARY DIMENSION

Darko Dimitrov and Klaus Kriegel
Freie Universität Berlin, Institute of Computer Science
Takustrasse 9, D-14195 Berlin, Germany

Keywords: Reflective Symmetry, Geometric Hashing, Principal Component Analysis.

Abstract: Symmetry detection is an important problem with many applications in pattern recognition, computer vision and computational geometry. In this paper, we propose a novel algorithm for computing a hyperplane of reflexive symmetry of a point set in arbitrary dimension with approximate symmetry. The algorithm is based on the geometric hashing technique. In addition, we consider a relation between the perfect reflective symmetry and the principal components of shapes, a relation that was already a base of few heuristic approaches that tackle the symmetry problem in 2D and 3D. From mechanics, it is known that, if H is a plane of reflective symmetry of the 3D rigid body, then a principal component of the body is orthogonal to H. Here we extend that result to any point set (continuous or discrete) in arbitrary dimension.

1 INTRODUCTION AND RELATED WORK

Symmetry is one of the most important features of shapes and objects, which is proved to be a powerful concept in solving problems in many areas including detection, recognition, classification, reconstruction and matching of different geometrics shapes, as well as compression of their representations. In general, symmetry in Euclidean space can be defined in terms of three transformations: translation, rotation and reflection. A subset P of $\mathbb{R}^d$ is *approximately symmetric* with respect to transformation T if for a big enough subset P' of P, the *distance* between $T(P')$ and P' is less then small constant ε, where the distance is measured using some appropriate metric, for example *Hausdorff*, *RMS (root mean square)* or *bottleneck distance measures* as most commonly used metrics. If $P' = P$ and $\varepsilon = 0$, then $T(P) = P$, and we say that P is *perfectly symmetric* with respect to T. In this paper we are interested in both approximate and perfect symmetry in terms of transformation of reflection through a hyperplane.

In what follows, we briefly survey the most relevant existing algorithms and techniques, we are aware of, for identifying both perfect and approximate symmetry.

Traditional approaches consider perfect symmetry in discrete settings as a global feature. Some of these methods reduced the symmetry detection problem to a detection of symmetries in circular strings (Atallah, 1985; Wolter et al., 1985; Highnam, 1986; Zhang and Huebner, 2002), for which efficient solutions are known (Knuth et al., 1977). Other efficient algorithms based on the octree representation (Minovic et al., 1993), the extended Gaussian image (Sun and Sherrah, 1997) or the singular value decomposition of the points of the model (Shah and Sorensen, 2005) also have been proposed. Further, methods for describing local symmetries were developed. (Blum, 1967) proposed an algorithm based on a medial axis transform. An algorithm presented in (Thrun and Wegbreit, 2005) detects perfect symmetries in range images, exploiting taxonomy of different types of symmetries and relations between them, by explicitly searching an increasing sets of points. A very recent approach, based on generalized moment functions and their spherical harmonics representation, was introduced by (Martinet et al., 2006). However, since the above mentioned methods consider only perfect symmetries, they may be inaccurate in detection the symmetry for shapes with added

Dimitrov D. and Kriegel K. (2007).
DETECTION OF PERFECT AND APPROXIMATE REFLECTIVE SYMMETRY IN ARBITRARY DIMENSION.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IU/MTSV*, pages 128-136

noise or missing data.

As a result to this challenge, several algorithms for measuring imperfect symmetries have been developed. For example, Zabrodsky et al. proposed an algorithm based on a *measure of symmetry*, defined as minimum mean squared distance required to transform a shape into a symmetric shape (Zabrodsky et al., 1993; Zabrodsky et al., 1995). A method of detecting a line of approximate symmetry of 2D images considering only the boundary of the image, using a hierarchy of certain directional codes, was presented in (Parui and Majumder, 1983). Marola introduced a measure of reflective symmetry with respect to a given axis where global reflective symmetry is found by roughly estimating the axis location and then fine tuning the location by minimizing the symmetry measure (Marola, 1989). Kazhdan et al. introduced the *symmetry descriptors*, a collection of spherical functions that describe the measure of a model symmetry with respect to every axis passing through the center of gravity (Kazhdan et al., 2003; Kazhdan et al., 2004). Very recently, Podolak et al. proposed the *planar reflective symmetry transform*, which measures the symmetry of an object with respect to all planes passing through its bounding volume (Podolak et al., 2006). A method of detecting planes of reflective symmetry, by exploiting the topological configuration of the edges of a 2D sketch of a 3D objects, was developed by (Zou and Lee, 2005). Mitra et al. proposed a method of finding partial and approximate symmetry in 3D objects (Mitra et al., 2006). Their approach relies on matching geometry signatures (based on the concept of normal cycles) that are used to accumulate evidence for symmetries in an appropriate transformation space.

Till now, most of the research was dedicated to investigation of symmetry in 2D and 3D. Here, we consider two approaches which lead to algorithms in arbitrary dimension. The contribution of this work is two-fold. First, we propose a novel algorithm, based on geometric hashing, for computing the reflectional symmetry of point sets with approximate symmetry in arbitrary dimension. Second, we give a proof of the relation between the perfect reflective symmetry and the principal components of discrete or continuous geometrical objects in arbitrary dimensions. The relation, in the case when rigid objects in 3D are considered, is known from mechanics and is established by analyzing a moment of inertia (Symon, 1971). Without rigorous proof for other cases than 3D rigid objects, this result was a base as a heuristic in several symmetry detection algorithms (Minovic et al., 1993; O'Mara and Owens, 1996; Sun and Sherrah, 1997). Banerjee et al. also tackle this relation in 3D, in the case when the objects are represented as 3D binary arrays, but a formal proof is missing in their paper (Banerjee et al., 1994).

The rest of the paper is organized as follows: In Section 2 we present the algorithm based on geometric hashing for computing a reflectional symmetry of a point set with approximate symmetry. The behavior of the algorithm in the 2D case is estimated by probabilistic analysis and evaluated on real and synthetic data. In Section 3, we give a proof of the relation between the perfect reflective symmetry and the principal components of geometrical objects in arbitrary dimensions. Conclusions and indications of future work are given in Section 4.

2 DETECTION OF REFLECTIVE SYMMETRY: GEOMETRIC HASHING APPROACH

Geometric hashing is a recognition technique based on matching of transformation-invariant object representations stored in a hash table (Wolfson and Rigoutsos, 1997; Alt and Guibas, 1999). Here, we assume that the given point set $P \subseteq \mathbb{R}^d$ is approximately symmetric, and our goal is to compute the hyperplane of symmetry H_{sym} with a geometric hashing technique. More precisely, hashing is utilized to compute the normal vector of H_{sym}. Additionally, one could use the fact that the center of gravity of P lies on H_{sym} in the case when P has a perfect symmetry, or with high probability near to H_{sym} in the case when P is approximately symmetric. However, to be on the safe side, if some outliers cause that the center of gravity is far from H_{sym}, we can apply a second phase of geometric hashing to compute a point on H_{sym}.

We start from the hypothesis that each point pair (p,q) is a candidate for a pair of points that are symmetric with respect to H_{sym}. Without loss of generality, we assume that the first coordinate of p is less than or equal to the first coordinate of q. If p is symmetric to q, the vector $\overrightarrow{pq}$ is orthogonal to H_{sym}. We note that this vector is characterized uniquely by the tuple of angles $(\alpha_2, \alpha_3, \ldots, \alpha_d)$ where α_i is the angle between $\overrightarrow{pq}$ and the i-th vector of the standard base of $\mathbb{R}^d$.

Since we assume at least a weak form of symmetry, we can expect that the number of point pairs (approximately) symmetric regarding H_{sym}, is bigger than the number of point pairs (approximately) symmetric regarding any other hyperplane H. For example, if we have a perfect symmetric point set with n points, then we have $\frac{n}{2}$ point pairs perfectly symmetric regarding H_{sym}. In contrast to that, the hyperplanes corresponding to remaining $\binom{n}{2} - \frac{n}{2}$ point pairs are randomly distributed. See Fig. 1 for illustration in $\mathbb{R}^2$.

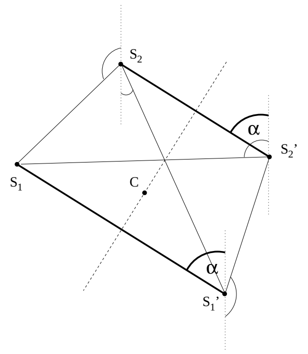

Figure 1: The angle α between y-axis and the line segments $\overline{s_1 s_1'}$ and $\overline{s_2 s_2'}$, formed by symmetric points, occurs two times. All other angles occurs only once.

In the standard approach of geometric hashing a number $K \in \mathbb{N}$ is fixed and the interval $[0, \pi]$ is subdivided into K subintervals of equal length π/K. Then, the hash function maps a tuple of angles $(\alpha_2, \alpha_3, \ldots, \alpha_d)$ to a tuple of integers $(a_2, a_3, \ldots, a_d)$, where each a_i denotes the index of the subinterval containing α_i, i.e.,

$$a_i = \left\lfloor \frac{\alpha_i \cdot K}{\pi} \right\rfloor .$$

Equivalently one can describe this approach with a so-called voting scheme by subdividing the cube $[0, \pi]^{d-1}$ into a grid with K^{d-1} cells. Each cell is equipped with a counter, collecting votes of all point pairs whichs angle tuple is contained in the cell. In the end one has to search for the cell with the maximum number of votes. However, this simple idea has some drawbacks related to the choice of K. Since K^{d-1} is a lower bound for both, time and storage complexity of the algorithm, K should not be too large. Moreover, if K is large, the noise might cause that the peak of votes is distributed over a larger cluster of cells. On the other side, if K is small, the preciseness of the result is not satisfactory.

We overcome these problems generalizing an idea from (Pleißner et al., 1999) that combines a rather coarse grid structure with a quite precise information about the normal vector. To this end, we use counters for the grid's vertices instead of counters for the grid's cells. Any vote $(\alpha_2, \alpha_3, \ldots, \alpha_d)$ for a grid cell $(a_2, \ldots, a_d)$ will be distributed to the incident vertices of the cell such that vertices close to $(\alpha_2, \alpha_3, \ldots, \alpha_d)$ get a larger portion of the vote than more distant vertices.

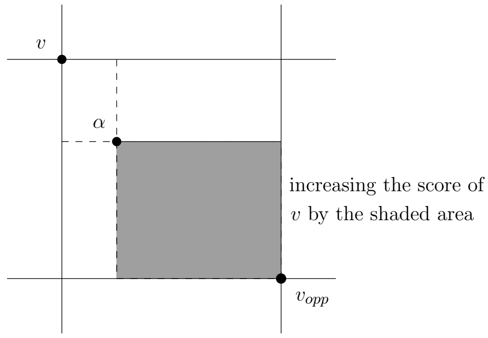

Figure 2: Updating the score for the angle vector $\alpha = (\alpha_1, \alpha_2)$.

To explain this idea more precisely, we introduce some more notations. Let Q be a grid cell, and v a grid vertex incident with Q. Among the vertices incident with Q, there is exactly one, called the opposite vertex v_{opp}, that differs in all $d-1$ coordinates from v. If $\vec{\alpha} = (\alpha_2, \alpha_3, \cdots, \alpha_d)$ is a vote for Q (i.e., a point in Q) we denote by $Q(\vec{\alpha}, v)$ the (axis-parallel) subcube of Q spanned by the points $\vec{\alpha}$ and v. It is clear that the closer $\vec{\alpha}$ is to v the larger is the volume of $Q(\vec{\alpha}, v_{opp})$. Thus, the unit score of $\vec{\alpha}$ will be distributed to all vertices incident with Q such that each vertex v gets the score $vol(Q(\vec{\alpha}), v_{opp})/vol(Q)$. See Figure 2 for illustration in $\mathbb{R}^3$. We remark that K^{d-1} counters suffice for $(K+1)^{d-1}$ grid vertices because the scoring scheme must be treated as cyclic structure in the sense that any vertex of the form $(\beta_2, \ldots, \pi, \ldots, \beta_d)$ is identified with $(\pi - \beta_2, \ldots, 0, \ldots, \pi - \beta_d)$.

Outline of the algorithm.

Input: A set of n points $P \in \mathbb{R}^d, d \geq 2$, with approximate symmetry.
Output: An approximation of H_{sym}.

1. Let X be the set of all point pairs (p, q) from P such that the first coordinate of p is less than or equal to the first coordinate of q. Compute for each pair the angle tuple $\vec{\alpha} = (\alpha_2, \ldots, \alpha_d)$.

2. Install a voting scheme of K^{d-1} counters and set all counters to 0.

3. For each $(p, q) \in X$ with $\vec{\alpha} = (\alpha_2, \ldots, \alpha_d)$ determine the corresponding grid cell Q. For all vertices v incident with Q, add to the counter of v the vote $vol(Q(\vec{\alpha}), v_{opp})/vol(Q)$.

4. Search for the vertex $v = v_{max}$ with the largest score w. Compute the angle tuple of the approximate normal vector of H_{sym} as the weighted center of gravity of v and its neighboring vertices with

the following formula:

$$\vec{\beta} = \frac{wv + \sum_{i=2}^{d} w_i^+ v_i^+ + \sum_{i=2}^{d} w_i^- v_i^-}{w + \sum_{i=2}^{d} w_i^+ + \sum_{i=2}^{d} w_i^-},$$

where $v_i^+, v_i^-, 2 \leq i \leq d$, denote the neighboring vertices of v, and w_i^+, w_i^- their corresponding scores. Let $\vec{n}$ be a normal vector in $\mathbb{R}^d$ corresponding to the angle tuple $\vec{\beta}$.

5. Approximate a point on H_{sym} selecting all pairs $(p,q) \in X$ that vote for v_{max} (i.e., $\vec{\alpha}$ is in a cell incident with v_{max}). For each selected pair project the center $c = (p+q)/2$ onto the line spanned by the normal vector $\vec{n}$ and store the position of the projected point on that line in a 1-dimensional scoring scheme. Use the maximal score to extrapolate the location of a point on H_{sym} analogously as in 4.

Taking into account that we can keep the parameter K small, the crucial step of the algorithm is the third one, because it requires the processing of $\Theta(n^2)$ point pairs. However, it is possible to reduce this effort under the assumption that the center of gravity $c(P)$ is close to H_{sym}. This holds whenever the points without symmetric counterpart are distributed regularly in the sense that their center of gravity is close to the center of gravity of the symmetric point set. In this case it is sufficient to consider votes of pairs (p,q) of points with nearly equal distances to $c(P)$. If δ is a bound for both, the distance of $c(P)$ to H_{sym} and the distortion of the symmetric counterpart of a point with respect to H_{sym}, the first step of the algorithm can be replaced as follows:

- Compute the center of gravity $c(P)$.
- Order the points of P with respect to the distance to $c(P)$.
- For all points $q \in P$ find the first point p_i and the last point p_j in the ordered list such that $dist(p_i, c(P)) \geq dist(q, c(P)) - 2\delta$ and $dist(p_j, c(P)) \leq dist(q, c(P)) + 2\delta$ and form X from the pairs $\{q, p_k\}$, $i \leq k \leq j$.

Although this modification does not improve the run time in the worst case, it effects a remarkable speed up of the algorithm for real world data.

2.1 Probabilistic Analysis and Evaluation of the Algorithm in 2D Case

The 2D version of the algorithm has been implemented and tested on real and synthetic data. The generation of the synthetic data is based on a probabilistic model, which additionally can be used for

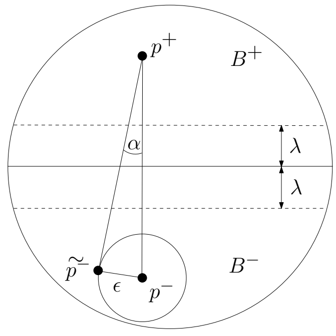

Figure 3: Point set generation.

a probabilistic analysis of the reliability of the algorithm.

The model incorporates the following two aspects of an approximately symmetric point set P. First, for the majority of the points $p \in P$ there is a counterpart $\widetilde{p}$ that is located close to the symmetric position of p, where the symmetry, with out loss of generality, is defined with respect to the x-axis. Second, there is a smaller subset of points in P without symmetric counterpart. To obtain such a point set, we apply the following procedure (see Fig. 3 for illustration). In the upper half of the unit ball B^+, we uniformly generate a random point set P^+ with n points. In the lower half B^- we reflect the point set P^+ over the x-axis and perturb it randomly. So, we obtain the set of points $\widetilde{P}^- = \{(x \pm \delta_x, -y \pm \delta_y) \mid (x,y) \in P^+\}$, where (δ_x, δ_y) is random point from the ball $B((0,0),\varepsilon)$. Additionally, we generate a random point set M in B, with m points, which do not have symmetric counterpart. Point set M represents an additional noise in the form of missing/extra points in the input data set.

Most pairs of symmetric points span a line that is nearly parallel to the y-axis. A vote of such pair will be called a *good* vote. Nevertheless, for points $p^+ \in P^+$ that are close to the x-axis the perturbation of p^- might cause a bigger angle α between the y-axis and the line spanned by $\widetilde{p^-}$ and p^+. A vote from such point pairs, as well as votes from nonsymmetric point pairs, will be called *bad*. Thus, we introduce a parameter $\lambda > 0$ defining a stripe of width 2λ along the x-axis such that all symmetric point pairs without this stripe have good votes.

Our goal is to derive an upper bound for ε that makes almost sure, that the given symmetry line corresponds to a maximal peak in the scoring scheme. We first estimate the width of the interval collecting the votes of the majority of the correct point pairs regarding to the symmetry line. On the other side, we will show that the probability, that another interval of the the same width would collect the same order of

Table 1: Empirical probability of finding correct line of reflective symmetry for different values of the "noise" parameters ε and k.

$k \setminus \varepsilon$	0.01	0.005	0.004	0.003	0.002	0.001	0.0
0.9	0.90	0.92	0.93	0.94	0.94	0.95	0.95
0.8	0.91	0.93	0.94	0.95	0.95	0.96	0.96
0.7	0.91	0.93	0.94	0.94	0.95	0.96	0.97
0.6	0.94	0.93	0.96	0.96	0.97	0.99	0.99
0.5	0.96	0.99	0.96	0.99	1.0	1.0	1.0
0.4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

votes, is very small for bounded ε.

Since the scoring scheme is a cyclic structure, it also makes sense to speak about negative angles: especially, angles $\alpha \in (\frac{\pi}{2}, \pi)$ will be identified with the negative angles $\alpha - \pi \in (-\frac{\pi}{2}, 0)$. According to Fig. 3, for a symmetric point pair outside the λ stripe we have the following bound on the angle α which defines the vote of the pair: $\sin\alpha \leq \frac{\varepsilon}{2\lambda}$, or $|\alpha| \leq \arcsin\frac{\varepsilon}{2\lambda}$. Since $\arcsin\frac{\varepsilon}{h} \leq \frac{\pi}{2}\frac{\varepsilon}{h} \leq \frac{\pi}{2}\frac{\varepsilon}{2\lambda}$, we have

$$|\alpha| \leq \arcsin\frac{\varepsilon}{2\lambda} \leq \frac{\pi\varepsilon}{4\lambda}. \tag{1}$$

We set $\gamma(\varepsilon, \lambda) := \frac{\pi\varepsilon}{4\lambda}$ and introduce for any angle β the random variable V_β counting all votes of the random point set P that fall into the interval $[\beta - \gamma(\varepsilon, \lambda), \beta + \gamma(\varepsilon, \lambda)]$.

Let $A_1 = \pi/2$ denote the area of the upper half of the unit ball and $A_2 = 2\lambda$ denote the area of the rectangle over the horizontal diameter of the unit ball with height λ. Thus, the probability that a point $p \in P^+$ generates a good pair is at least $q = \frac{A_1 - A_2}{A_1} = (1 - \frac{4\lambda}{\pi})$. Since V_0 is at least the sum S of n independent variables

$$X_i = \begin{cases} 1 & \text{with probability } q; \\ 0 & \text{with probability } 1 - q, \end{cases}$$

we have

$$E(V_0) \geq E(S) = nq, \tag{2}$$

and

$$Pr[V_0 < t] \leq Pr[S < t], t > 0. \tag{3}$$

Combining (3) with the Chernoff inequality $Pr[S < E[S] - t] \leq e^{-2t^2/n}$, for $t = E[S]/2 = nq/2$, we obtain the following estimation:

$$Pr(V_0 < nq/2) \leq e^{-q^2 n/2}. \tag{4}$$

Let $N \leq \binom{2n+m}{2}$ be the number of points pairs with bad votes, and consider an angle β, where $|\beta| > 2\gamma(\varepsilon, \lambda)$, i.e., X_β doesn't count any good vote. The expectation of X_β is

$$E(X_\beta) = N\frac{2\gamma(\varepsilon, \lambda)}{\pi} = N\frac{\varepsilon}{2\lambda}. \tag{5}$$

Applying the Markov inequality $Pr[X_\beta > t] \leq \frac{E(X_\beta)}{t}$, for $t = nq/2$, we obtain

$$Pr[X_\beta > nq/2] \leq \frac{N\varepsilon}{\lambda q n}. \tag{6}$$

We would like to note that in the case of X_β, we cannot apply any of the Chernoff's inequalities, which in general give better bounds than the Markov inequality, because X_β is not a sum of independent random variables.

Now, we come to the ultimate goal of this analysis - to estimate $Pr[V_\beta > V_0]$ and to study when it is small, i.e., when the algorithm gives a correct answer with high probability. From

$$Pr[V_\beta > V_0] \leq Pr[V_\beta > t] + Pr[V_0 < t], t > 0, \tag{7}$$

(4) and (6), we obtain

$$Pr[V_\beta > V_0] \leq e^{-q^2 n/2} + \frac{N\varepsilon}{\lambda\pi q}. \tag{8}$$

The first term of the right side of (8) is significantly smaller then the second term. This can be explained by the fact that the first term was obtained by the Chernoff inequality, and the second term by the weaker Markov inequality. However, for $\varepsilon = o(\frac{1}{n})$ the second term will be also small, and then the algorithm will work well with high probability.

As described above, we randomly generated 100 point sets with same parameters ε and k, where k is the ratio between the number of additional points and the number of good point pairs ($k = m/n$). Table 1 shows the empirical probability of finding the correct angle of the symmetry line. We present here only those combination of ε and k for which the empirical probability was at least 0.9. The results indicate that the algorithms is less sensitive to noise, due to missing/extra data, then to noise that comes from imperfect symmetry of the points. This conclusion is consistent with the theoretical analysis we have obtained. Namely, ε and N occur at the same place in the last term of the relation (8). The number of additional points m occurs in the relation (8) through N. The other variable which determines N is n, and its contribution to the value of N is bigger than that of m. Therefore, m has smaller influence to the expression than ε.

We tested the algorithm also on real data sets. The tests were performed on pore patterns of copepods - a group of small crustaceans found in the sea and nearly every freshwater habitat (see Fig. 4). The pores in a pattern were detected as points by the method based on a combination of hierarchical watershed transformation and feature extraction methods presented in (Pleißner et al., 1999). The algorithm successfully detected the symmetry line because the extracted point sets have relatively good reflective symmetry, and majority of the points (around 90%) have a symmetric counterpart.

3 DETECTION OF REFLECTIVE SYMMETRY: PCA APPROACH

Another approach for an efficient detection of the hyperplane of perfect reflective symmetry in arbitrary dimension is that based on principal component analysis (Jolliffe, 2002). To the best of our knowledge, this approach was used as heuristic without rigorous proof (also confirmed in communication with other researchers in this area (O'Mara and Owens, 2005)). A relation between the principal components and symmetry of an object, in the case of rigid objects in 3D, was establish in mechanics by analyzing a moment of inertia (Symon, 1971). This result, in the context of detecting the symmetry, was first exploit by (Minovic et al., 1993). Here we extend that result to any set of points (continuous or discrete) in arbitrary dimension. The central idea and motivation of PCA (also known as the Karhunen-Loeve transform, or the Hotelling transform) is to reduce the dimensionality of a data set by identifying *the most significant directions (principal components)*. Let $P = \{p_1, p_2, \ldots, p_m\}$, where p_i is a d-dimensional vector, and $c = (c_1, c_2, \ldots, c_d) \in \mathbb{R}^d$ be the center of gravity of P. For $1 \leq k \leq d$, we use p_{ik} to denote the k-th coordinate of the vector p_i. Given two vectors u and v, we use $\langle u, v \rangle$ to denote their inner product. For any

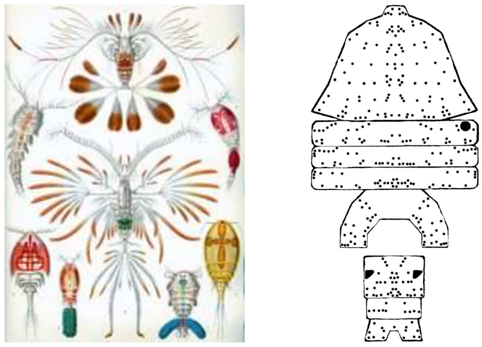

Figure 4: Left side: illustrations of different types of copepods. Right side: a pore pattern of a copepod.

unit vector $v \in \mathbb{R}^d$, the *variance of P in direction v* is

$$var(P, v) = \frac{1}{m} \sum_{i=1}^{m} \langle p_i - c, v \rangle^2. \tag{9}$$

The most significant direction corresponds to the unit vector v_1 such that $var(P, v_1)$ is maximum. In general, after identifying the j most significant directions $B_j = \{v_1, v_2, \ldots, v_j\}$, the $(j+1)$-th most significant direction corresponds to the unit vector v_{j+1} such that $var(P, v_{j+1})$ is maximum among all unit vectors perpendicular to $v_1, v_2, \ldots, v_j$.

It can be verified that for any unit vector $v \in \mathbb{R}^d$,

$$var(P, v) = \langle Cv, v \rangle, \tag{10}$$

where C is the *covariance matrix* of P. C is a symmetric $d \times d$ matrix where the ij-th component, $C_{ij}, 1 \leq i, j \leq d$, is defined as

$$C_{ij} = \frac{1}{m} \sum_{k=1}^{m} (p_{ik} - c_i)(p_{jk} - c_j). \tag{11}$$

The procedure of finding the most significant directions, in the sense mentioned above, can be formulated as an eigenvalue problem. If $\lambda_1 > \lambda_2 > \cdots > \lambda_d$ are the eigenvalues of C, then the unit eigenvector v_j for λ_j is the j-th most significant direction. All λ_js are non-negative and $\lambda_j = var(P, v_j)$. Since the matrix C is symmetric positive definite, its eigenvectors are orthogonal. If the eigenvalues are not distinct, the eigenvectors are not unique. In this case, an orthogonal basis of eigenvectors is chosen arbitrary. However, we can achieve distinct eigenvalues by a slight perturbation of the point set.

In the case when P is a continuous set of d-dimensional vectors, all above expressions have analogons defined in terms of integrals instead of finite sums. Due to the space limitation, we omit them here.

Now, we prove the following connection between hyperplane reflective symmetry and principal components.

Theorem 3.1 *Let P be a d-dimensional point set symmetric with respect to a hyperplane H_{sym} and assume that the covariance matrix C has d different eigenvalues. Then, a principal component of P is orthogonal to H_{sym}.*

Proof. Without loss of generality, we can assume that the hyperplane of symmetry is spanned by the last $d-1$ standard base vectors of the d-dimensional space and the center of gravity of the point set coincides with the origin of the d-dimensional space, i.e., $c = (0,0,\dots,0)$. Then, the components C_{1j} and C_{j1} are 0 for $2 \leq j \leq d$, and the covariance matrix has the form:

$$C = \begin{bmatrix} C_{11} & 0 & \dots & 0 \\ 0 & C_{22} & \dots & C_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & C_{d2} & \dots & C_{dd} \end{bmatrix} \qquad (12)$$

Its characteristic polynomial is

$$det(C - \lambda I) = (C_{11} - \lambda) f(\lambda), \qquad (13)$$

where $f(\lambda)$ is a polynomial of degree $d-1$, with coefficients determined by the elements of the $(d-1) \times (d-1)$ submatrix of C. From this it follows that C_{11} is a solution of the characteristic equation, i.e., it is an eigenvalue of C and the vector (1, 0, ...,0) is its corresponding eigenvector (principal component), which is orthogonal to the assumed hyperplane of symmetry. $\square$

As an immediate consequence of Theorem 3.1 we have:

Corollary 3.2 *Let P be a perfectly symmetric point set in arbitrary dimension. Then, any hyperplane of reflective symmetry is spanned by n-1 principal axes of P.*

The corollary implies a straightforward algorithm for finding the hyperplane of reflective symmetry of a point set in arbitrary dimension.

Outline of the algorithm.

Input: A set of n points $P \in \mathbb{R}^d, d \geq 2$, with approximate symmetry.
Output: An approximation of H_{sym}.

1. Compute the covariance matrix C of P.
2. Compute the eigenvectors of C and the candidate hyperplanes of reflective symmetry.
3. Reflect the points through every candidate hyperplane.
4. Find if each reflected point is enough close to a point in P. The correspondence between reflected points and points in P is bijection.

The first and third step of the algorithm have linear time complexity in the number of points. Computation of the eigenvectors, when d is not very large, can be done in $O(d^3)$ time, for example with Jacobi or QR method (Press et al., 1995). Computing the candidate hyperplanes can be done in $O(d)$. Therefore, for fixed d, the time complexity of the second step is constant. For very large d, the problem of computing eigenvalues is non-trivial. In practice, the above mentioned methods for computing eigenvalues converge rapidly. In theory, it is unclear how to bound the running time combinatorially and how to compute the eigenvalues in decreasing order. In (Cheng et al., 2005) a modification of the *Power method* (Parlett, 1998) is presented, which can give a guaranteed approximation of the eigenvalues with high probability. However, for reasonable big d the most expensive step is the forth one. Here we can apply an algorithm for nearest neighbor search, for example the algorithm based on Voronoi diagram, which together with preprocessing has run time complexity $O(n \log n)$, $d = 2$, or $O(n^{\lceil \frac{d}{2} \rceil})$, $d \geq 3$. If we consider point sets with perfect symmetry, then in the 4-th step, it suffices to check if the reflection of a point of P is identical with other point of P. For this, we will need to sort the points lexicographically, and since this is computationally most expensive part in the whole algorithm, it follows that the above algorithm in the case of detecting perfect symmetry has time complexity $O(n \log n)$ in arbitrary dimension.

In what follows, we discuss two problems that may arise in theory, but are relatively uncommon in practice. The first one considers the case when the eigenvalues are not distinct, and the other the case when one or more variables are zero.

Equality of eigenvalues, and hence equality of variances of PCs, will occur for certain patterned matrices. The effect of this occurrence is that for a group of q equal eigenvalues, the corresponding q eigenvectors span a certain unique q-dimensional space, but, within this space, they are, apart from being orthogonal to one another, arbitrary. In the context of our problem, it means that the d-dimensional point set will have exactly d candidates as hyperplanes of symmetry only when the eigenvalues of the covariance matrix are distinct. For example, if we have 3-dimensional point set, then if exactly 2 eigenvalues of the covariance matrix are equal, than the point set might has rotational and reflective symmetry. If the all 3 are equal, the point set might have any type of symmetry, including spherical symmetry. To justify this geometrically, we can imagine what happens to the covariance ellipsoid in this cases. For example, in the case when all 3 eigenvalues are equal it be-comes a ball. In the case when the eigenvalues are not distinct, we can slightly perturb the point set, and obtain unique approximate hyperplanes of reflective symmetry.

The case when q variances equal zero, implies that the rank of covariance matrix of the point set diminishes for q. Therefore we can reduce the d-dimensional problem to a $(d-q)$-dimensional problem.

Beside its simplicity and efficiency, as it is known, detecting symmetry by PCA has two drawbacks. PCA fails to identify potential hyperplanes of symmetry, when the eigenvalues of the covariance matrix of the object are not distinct. The second drawback is that PCA approach cannot guaranty the correct identification when the symmetry of the shape is too weak.

4 CONCLUSION AND FUTURE WORK

The most of the research effort on symmetry detection was dedicated to shapes and object in 2D and 3D. In this paper, we proposed a novel algorithm which is also able to detect a hyperplane of reflective symmetry in arbitrary dimension. The algorithm is based on the modified version of geometric hashing. We have implemented a 2D variant of the algorithm. The behavior of the algorithm was analyzed with a probabilistic model. The tests on real and synthetic data showed that the algorithm is robust when the symmetry is not too weak, and that it is quite insensitive on outlayers.

The second contribution of this paper is the proof of the relation between the reflective symmetry and principal components of any type of symmetric geometric shapes in arbitrary dimension. The only related result to this is the result known from the mechanics, which establish the above relation for rigid bodies in 3D. We present here a stronger result, which confirms this relation for any symmetric geometric shape in arbitrary dimension. That opens a possibility to generalize some already known ideas from 2D and 3D in higher dimensions.

An implementation of the geometric hashing algorithm in higher dimensions and estimations of its behavior is one of the tasks for future work. Of course, the 3D case is of the biggest practical importance. Comparing the results obtained by both here presented algorithms, as well as comparing them with other algorithms for detecting reflective symmetry is of interest.

REFERENCES

Alt, H. and Guibas, L. (1999). Discrete geometric shapes: Matching, interpolation, and approximation. In Sack, J.-R. and Urrutia, J., editors, *Handbook of Computational Geometry*, pages 121 – 153. Elsevier Science Publishers B.V. North-Holland, Amsterdam.

Atallah, M. J. (1985). On symmetry detection. In *IEEE Trans. on Computers c-34, 7*, pages 663–666.

Banerjee, D. K., Parui, S. K., and Majumder, D. D. (1994). Plane of symmetry of 3d objects. In *Proc. of the 29th Annual Convention of the Computer Society of India. Information Technology for Growth and Prosperity.*, pages 39–44.

Blum, H. (1967). A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, W. Whaten-Dunn, Ed.

Cheng, S.-W., Wang, Y., and Wu, Z. (2005). Provable dimension detection using principal component analysis. In *Proc. of the 21st ACM Symposium on Computational Geometry*, pages 208–217.

Highnam, P. T. (1986). Optimal algorithms for finding the symmetries of a planar point set. *Inf. Process. Lett.*, 22(5):219–222.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, 2nd edition.

Kazhdan, M., Chazelle, B., Dobkin, D., Funkhouser, T., and Rusinkiewicz, S. (2003). A reflective symmetry descriptor for 3d models. *Algorithmica, 38*, pages 201 – 225.

Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. (2004). Symmetry descriptors and 3D shape matching. In *Symposium on Geometry Processing*, pages 115–123.

Knuth, D., Morris, J. H., and Pratt, V. (1977). Fast pattern matching in strings. *SIAM Journal of Computing 6, 2*, pages 323–350.

Marola, G. (1989). On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Transactions o Pattern Analysis and Machine Intelligence*, 11(1):104–8.

Martinet, A., Soler, C., Holzschuch, N., and Sillion, F. (2006). Accurate detection of symmetries in 3d shapes. *ACM Transactions on Graphics*, 25(2):439 – 464.

Minovic, P., Ishikawa, S., and Kato, K. (1993). Symmetry identification of a 3-d object represented by octree. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 15(5), pages 507–514.

Mitra, N. J., Guibas, L., and Pauly, M. (2006). Partial and approximate symmetry detection for 3d geometry. In *ACM Transactions on Graphics*, volume 25, pages 560–568.

O'Mara, D. and Owens, R. (1996). Measuring bilateral symmetry in digital images. In *IEEE TENCON Digital Signal Processing Application*, pages 151–156.

O'Mara, D. and Owens, R. (2005). Private communication.

Parlett, B. N. (1998). *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia,PA.

Parui, S. and Majumder, D. (1983). Symmetry analysis by computer. In *Pattern Recognition*, volume 16, pages 63–67.

Pleißner, K.-P., Hoffmann, F., Kriegel, K., Wenk, C., Wegner, S., Sahlströhm, A., Oswald, H., Alt, H., and Fleck, E. (1999). New algorithmic approaches to protein spot detection and pattern matching in two-dimensional electrophoresis gel databases. *Electrophoresis*, 20:755–765.

Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., and Funkhouser, T. (2006). A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3):549–559.

Press, W. H., Teukolsky, S. A., Veterling, W. T., and Flannery, B. P. (1995). *Numerical Recipes in C: The art of scientific computing*. Cambridge University Press, New York, USA, second edition.

Shah, M. I. and Sorensen, D. C. (2005). A symmetry preserving singular value decomposition. In *SIAM Journal of Matrix Analysis and it's Applications (October)*.

Sun, C. and Sherrah, J. (1997). 3d symmetry detection using the extended gaussian image. In *IEEE PAMI 19, 2*, pages 164–168.

Symon, K. R. (1971). *Mechanics*. Reading, MA: Adison-Wesley, 3rd edition.

Thrun, S. and Wegbreit, B. (2005). Shape from symmetry. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1824–1831. IEEE Computer Scoiety.

Wolfson, H. and Rigoutsos, I. (1997). Geometric hashing: An overview. 24:10–21.

Wolter, J. D., Woo, T. C., and Volz, R. A. (1985). Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer 1*, pages 37–48.

Zabrodsky, H., Peleg, S., and Avnir, D. (1993). Completion of occluded shapes using symmetry. In *Proc. CVPR*, pages 678–679.

Zabrodsky, H., Peleg, S., and Avnir, D. (1995). Symmetry as a continuous feature. *Trans. PAMI 17*, 12:1154–1166.

Zhang, J. and Huebner, K. (2002). Using symmetry as a feature in panoramic images for mobile robot applications. In *Proc. Robotik*, volume 1679 of VDI-Berichte, pages 263–268.

Zou, H. L. and Lee, Y. T. (2005). Skewed mirror symmetry detection from a 2d sketch of a 3d model. In *GRAPHITE '05: Proc. of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 69–76. ACM Press.