A Fourier-Theoretic Approach for Inferring Symmetries

Xiaoye Jiang* Jian Sun† Leonidas Guibas‡

Abstract

In this paper, we propose a novel Fourier-theoretic approach for estimating the symmetry group $\mathbb{G}$ of a geometric object X. Our approach takes as input a geometric similarity matrix between low-order combinations of features of X and then searches within the tree of all feature permutations to detect the sparse subset that defines the symmetry group $\mathbb{G}$ of X. Using the Fourier-theoretic approach, we construct an efficient marginal-based search strategy, which can recover the symmetry group $\mathbb{G}$ effectively. The framework introduced in this paper can be used to discover symmetries of more abstract geometric spaces and is robust to deformation noise. Experimental results show that our approach can fully determine the symmetries of many geometric objects.

1 Introduction

Symmetries are extremely common in both man-made and natural objects. In the context of computational geometry, we often consider the *symmetry group* of a geometric object with a pre-defined metric. One easy way of describing all symmetries of geometric objects is to look at *group actions*, where we use a set to represent the object, and the symmetries of the object are described by bijective mappings on the set. In this paper, assume we have a discrete set $X = \{x_i\}_{i=1}^n$ which describes a geometric object. As shown in Figure 1-(a), the five tip points on a star can be used as a discrete set X to study the symmetry of such a 3D star model. This is because each symmetry of the star model can be identified with a permutation of the elements in X[1]. For convenience, we say a permutation is "good" if it can be identified as a symmetry of the geometric object. It can be shown that all good permutations of X consist a group $\mathbb{G}$, which is the *symmetry group of* X.

In practice, we are often limited to verifying low-order information about X, such as the similarity of curvatures between pairs of points (first order), or the consistency of distances between pairs of pairs of points (second order). But how can we integrate such low-order pieces of symmetry evidence together to identify all the good permutations of X and derive its symmetry group? This question is quite challenging since the space of all permutations grows factorially with the number of elements in X so that directly searching among all permutations is impossible, unless n is small. In this paper, we propose a Fourier-theoretic approach to address this problem, based on low-order similarities of points in X.

The symmetry group of X is a subgroup of the permutation group $\mathbb{S}_n$, where $n = |X|$. To search for $\mathbb{G}$, we naturally have the following simple strategy: we organize the elements of $\mathbb{S}_n$ in a tree where each node represents all the permutations in its sub-tree, and then search those in $\mathbb{G}$ within the tree, see Figure 1-(b) for an illustration of the tree. Whenever we reach a permutation of X, we check whether it is a good one. However, such a brute force strategy would be computationally intractable for all but very small n.

In this paper, we propose a novel search strategy within the tree of $\mathbb{S}_n$, called the *marginal probability search*, which fully exploits the algebraic structure of the groups $\mathbb{G}$ and $\mathbb{S}_n$. The main contribution of the paper are the following two ways of making use of algebraic structures to facilitate the search of symmetries.

Firstly, we consider the symmetry group $\mathbb{G}$ as an *indicator distribution* (see Theorem 1 in Section 2) over the permutation group $\mathbb{S}_n$. This novel point of view enables us to utilize techniques from the group representation theory to convert low-order information into a set of Fourier coefficients which characterizes the low-frequency components of the distribution over $\mathbb{S}_n$. Those Fourier coefficients can thus be used to efficiently estimate the marginal probability of the permutations represented by an internal node, which serves as the criterion for pruning the sub-tree rooted at that node. Unlike other traditional pruning criteria [4], the marginal probability is much more informative as it not only evaluates the part of the permutation which is already determined but also summarizes the remaining part which is undecided, and thus provides a more efficient pruning.

Secondly, we exploit the group structure of $\mathbb{G}$ and show that the internal nodes on the same level have either the same marginal probability as the node containing the identity permutation or 0 marginal probability for the indicator distribution $\mathbb{G}$. This ensures the correctness of taking the marginal probability of the in-

*Institute of Computational and Mathematical Engineering, Stanford University, xiaoyej@stanford.edu

†Mathematical Sciences Center, Tsinghua University, jsun@math.tsinghua.edu.cn

‡Department of Computer Science, Stanford University, guibas@cs.stanford.edu

[1]Choosing different set X can result in different group actions. In this paper, however, we assume such a set X is given where each symmetry can be identified as permuting X.

ternal nodes as the reference for pruning.

The approach proposed in this paper generalizes the existing work on graph automorphism, in the sense that we can deal with noisy similarity information caused by heavy distortion or deformation of the geometric object and robustly estimate the symmetry group $\mathbb{G}$ from the input. Moreover, any arbitrary order of similarities, e.g., triple-wise or even higher order similarities can be taken as input in our framework. Different orders of similarities can be combined together easily because Fourier analysis can fully disentangle and decompose the information over permutations of different orders into orthogonal components. In addition, our approach does not require a concrete realization of the geometric object. As long as a discrete set X, which characterizes the symmetry of the geometric object, can be effectively extracted, our approach can be used for inferring the symmetry group $\mathbb{G}$.

Related work We note that a great amount of research has already been done on Euclidean symmetry detection in the geometry processing community [7, 10]. However, those approaches often suffer from the curse of dimensionality. The problem of inferring the global symmetry from low-order similarities, is closely related to the *graph automorphism problem*, or more generally, the *colored graph automorphism problem*. However, there are no known polynomial time algorithms for finding the automorphism group of a general graph except for certain special cases such as the triply connected planar graph [2, 16]. The problem we consider also connects to the *orbit partitioning problem* whose goal is to determine whether two vertices or two pairs of vertices lie in the same orbits. However, those problems are generally very difficult, and there are no known polynomial time algorithms [1, 11, 13].

2 Marginal Probability of Cosets

In this section, we give a detailed description on how to organize all the elements in $\mathbb{S}_n$ in a tree. We also show that the marginal probabilities of the nodes on the same level only take two possible values for the *indicator distribution* of $\mathbb{G}$.

We consider a tree decomposition of all permutations in $\mathbb{S}_n$ as depicted in figure 1-(b). All permutations are classified into n sub-trees according to their mappings on the last element, i.e., $\sigma(n)$, where σ denotes a permutation. The n sub-trees are further classified according to their mappings on the last two elements, i.e., $\sigma(n-1)$ and $\sigma(n)$. In general, a node on the k-th level stands for all permutations that maps the tuple $(n-k+1,\cdots,n)$ to a particular k-tuple. Thus, the leaves in the tree represent all the permutations.

Let f be a distribution over $\mathbb{S}_n$. We consider the marginal probability of a node on the kth level:

$$\sum_{\sigma\in\mathbb{S}_n} f(\sigma) I\Big(\sigma(n-k+1,\cdots,n) = (t_{n-k+1},\cdots,t_n)\Big), \qquad (1)$$

which sums up all $f(\sigma)$ such that σ maps the k-tuple $(n-k+1,\cdots,n)$ to the k-tuple $(t_{n-k+1},\cdots,t_n)$ where t_i's are all distinct and each $t_i \in \{1,\cdots,n\}$. Here, I is an indicator function which is 1 if and only if σ maps i to t_i for all $n-k+1 \le i \le n$. Each node in the tree of permutations is associated with such a marginal probability. The following theorem characterizes specific properties of these marginal probabilities.

Theorem 1 *Let* $f(\sigma) = \begin{cases} \frac{1}{|\mathbb{G}|}, & \sigma \in \mathbb{G} \\ 0, & \sigma \notin \mathbb{G} \end{cases}$ (2)

be the indicator distribution for $\mathbb{G}$ in $\mathbb{S}_n$, and let m_k be the marginal probability of all permutations that fix $(n-k+1,\cdots,n)$, i.e., the quantity in (1) with $t_i = i$ $(n-k+1 \le i \le n)$. Then, we have $m_k \neq 0$; and for every node on the k-th level, its marginal probability (1) is either 0 or m_k.

The proof of Theorem 1 is based on coset representation theorems. This theorem immediately translates into a search strategy for estimating the group $\mathbb{G}$. Basically, we perform a top-down search in the tree of $\mathbb{S}_n$. A node which represent all permutations that map $(n-k+1,\cdots,n)$ to a k-tuple $(t_{n-k+1},\cdots,t_n)$ will be kept only if its marginal probability is nonzero. The group $\mathbb{G}$ can be fully decided if all those marginal information is available.

3 Inference with the Similarity Matrix

In real applications, the primary challenge for estimating the group $\mathbb{G}$ is that the marginal probabilities needed for search are not directly observable. Instead, we typically can only verify low-order similarities. In this section, we introduce two related concepts: the *similarity matrix* and the *marginal probability matrix*.

Definition 2 *A low-order similarity matrix S_k of order k (k is usually very small) for X ($|X| = n$) is an N-by-N matrix where $N = n(n-1)\cdots(n-k+1)$ and the (i,j)-entry is a similarity measure s_{ij} for two k-tuples $(t_1^{(i)},\cdots,t_k^{(i)})$ and $(t_1^{(j)},\cdots,t_k^{(j)})$ indexed by i,j.*

We can construct various *similarity measures* s_{ij} for two k-tuples indexed by i and j, for example:

- $k=1$: we can use a *binary rule* by letting $s_{ij}=1$ if and only if points i and j have the same curvature; or use a continuous *Gaussian kernel* $s_{ij} = \exp(-|c_i - c_j|^2)$ where c_i, c_j are the curvatures of the point i,j.
- $k=2$: we can use a *binary rule* by letting $s_{ij}=1$ if and only if distances d_i and d_j are the same, where d_i (d_j) is the distance between two points in the pair i (j); or use a *Gaussian kernel* $s_{ij} = \exp(-|d_i - d_j|^2)$.

Definition 3 *Given a distribution f on permutations ($\sum_\sigma f(\sigma) = 1$), the k-th order marginal probability matrix H_k of f is an N-by-N matrix where $N = n(n-1)\cdots(n-k+1)$ and the (i,j)-entry equals*

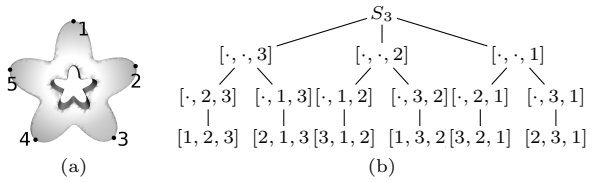

Figure 1: (a) Star. (b) Tree Decomposition of $\mathbb{S}_n$.

$\sum_\sigma f(\sigma)I(\sigma(i) = j)$, *where* i *and* j *index two* k*-tuples:* $(t_1^{(i)}, \cdots, t_k^{(i)})$ *and* $(t_1^{(j)}, \cdots, t_k^{(j)})$.[2]

For a given geometric object X, we can compute its low order similarity matrix. We hope such a matrix can approximate the marginal probability matrix of the indicator distribution if we normalize the similarity matrix so that each row sum equals one. As two real examples, we look at the low order similarity matrices for the star and human (see Figure 1-(a) and Figure 3-(d)).

For the star example, we compute the similarity matrices using the binary rule (see Figure 2-(a,c)). The first order similarity matrix is an all-one matrix which has no information about the symmetry, however, the second order similarity matrix S_2 can completely reveal the symmetry – if we normalize S_2 so that each row-sum is one, then the normalized similarity matrix exactly equals the marginal probability matrix H_2 of the distribution indicating the five-fold dihedral group $\mathbb{G} = \mathbb{D}_5$ as in (2) which characterizes the symmetry.

For the human example (see Figure 2-(b,d)), we compute the similarity matrices using the Gaussian kernel. The first order similarity matrix takes a block diagonal form, which partially reveals the symmetry of the human. However, there are still ambiguities that can not be resolved by first order information – whenever we map the left hand to the right hand, we have to map the left foot to the right foot. However, the second order similarity matrix S_2 constructed by computing $\exp(-|d_i - d_j|^2)$ can help us further clarify the symmetry group of the human – if we normalize S_2, then it well approximates (there are tiny noises within the human model) the marginal probability matrix H_2 of the distribution indicated by one-fold dihedral group $\mathbb{G} = \mathbb{D}_1$.

In the above two examples, we observe that the normalized similarity matrix is a good approximation of the marginal probability matrix for the distribution indicating $\mathbb{G}$ if we use a good similarity measure. Such a matrix can reveal $\mathbb{G}$ better if we incorporate higher order information because in the extremal case the n-th order marginal probability matrix can exactly pinpoint the distribution over permutations. Theoretically, we may prove that normalized low order similarity matrix in the noiseless case (computed using the binary rule) equals the marginal probability matrix in the manifold setting, as long as the signatures we use to construct similarity measures are powerful enough [9].

In this sequel, we assume the normalized low-order similarity matrix estimated from geometric objects approximates a marginal probability matrix of the indicator distribution of $\mathbb{G}$.[3] By using the Fourier transforms over permutation group, we can extract a set of low frequency Fourier coefficients from the normalized low-order similarity matrix, which provides a band-limited approximation (ℓ_2 projection in the Fourier space) for the indicator distribution of $\mathbb{G}$ over $\mathbb{S}_n$ [5]. With such a set of Fourier coefficients, we estimate all the marginal probabilities and search elements in $\mathbb{G}$ in the tree of all permutations.

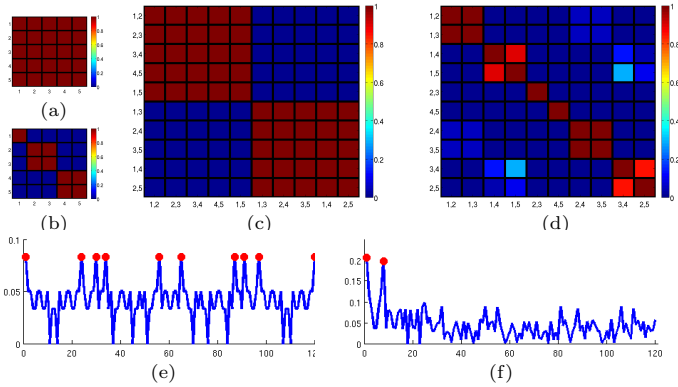

Figure 2: (a,b) First order similarity matrix for the star and human; (c,d) Second order similarity matrix for the star and human. The normalized similarity matrix are of the same block structures except that each row sum equals one. (e,f) Reconstructed distribution over permutations from the normalized second order similarity matrix for the star and human. Red dots denote good permutations.

3.1 Fourier Approach

In this section, we consider the problem of estimating all the marginal probabilities needed for search based on the normalized low-order similarity matrix S. We first translate the matrix S into a set of Fourier coefficients $\hat{f}_\lambda$'s using Specht modules [6]. Those Fourier components (indexed by λ) characterize the low-frequency components of the distribution f over permutations. After that, we compute a pointwise product of f with the indicator $I(\sigma(n) = (t_n))$ in the Fourier domain, so that the distribution over all permutations $\{\sigma\}$ such that σ maps n to t_n can be extracted. The result of such a pointwise distribution can be summarized by a distribution over $\mathbb{S}_{n-1}$ if one relabels $1, 2, \cdots, n$ so that t_n becomes n. We use an algorithm called *Kronecker Conditioning* to compute the pointwise product completely in the Fourier domain [5]. A theorem by [8] gives us a bound on which representations can appear in the result of such a pointwise product. We finally apply an FFT based approach, which will be described later, to compactly summarize such a distribution over $\mathbb{S}_{n-1}$.

The above procedure decomposes the distribution implied by S to n distributions over $\mathbb{S}_{n-1}$. Such a proce-

[2]The m_k defined in Theorem 1 is one element of H_k where $(t_1^{(i)}, \cdots, t_k^{(i)})$ and $(t_1^{(j)}, \cdots, t_k^{(j)})$ both equals $(n-k+1, \cdots, n)$.

[3]We make sure that the normalized similarity matrix is a valid marginal probability matrix by imposing certain inherent constraints such as doubly stochasticity [5].

dure can be used iteratively on each $\mathbb{S}_{n-1}$, until we reach the bottom of the tree decomposition of $\mathbb{S}_n$. At each node, a set of Fourier coefficients are maintained to characterize the distribution over permutations dominated by that node. The key benefit of using Fourier coefficients to summarize the information is due to the simplicity of evaluating marginals in the Fourier domain [5].

In this process, we see that the relabeling is used extensively, so that we can view the permutations as if they are always permuting $(1, \cdots, n-1)$, rather than mapping from $(1, \cdots, n-1)$ to $(1, \cdots, \hat{t}_n, \cdots, n)$. Whenever such a relabeling operation is used, the Fourier coefficients of f on $\mathbb{S}_n$ also change accordingly. It turns out that there is a class of operations, called *shift* operations [3] which can compute the Fourier transform with respect to the reordered sets.

3.2 FFT-Based Method

In this section, we detail how to extract the Fourier coefficients of f restricted on $\mathbb{S}_{n-1}$ from $\hat{f}_\lambda$, which is an essential step in estimating the marginals.

In the tree decomposition of $\mathbb{S}_n$, we see that $\mathbb{S}_n = \cup_{i=1}^n [\![i,n]\!]\mathbb{S}_{n-1}$, where $[\![i,n]\!]$ denotes the cyclic permutation $(i, i+1, \cdots, n)$ (i is mapped to $i+1$, $i+1$ is mapped to $i+2$, etc, n is mapped to i), and $[\![i,n]\!]\mathbb{S}_{n-1}$ is the so-called left $\mathbb{S}_{n-1}$-coset

$$[\![i,n]\!]\mathbb{S}_{n-1} = \{\sigma \in \mathbb{S}_n | \sigma(n) = i\} \tag{3}$$

The fast Fourier transform (FFT) for $\mathbb{S}_n$ works by relating the Fourier transform over $\mathbb{S}_n$ to Fourier transforms over the above n cosets. This idea can be applied recursively, computing the Fourier transform on each $\mathbb{S}_{n-1}$-coset from $n-1$ Fourier transforms on $\mathbb{S}_{n-2}$-cosets, etc., all the way down to $\mathbb{S}_1$-cosets, which are individual permutations. We will present a method to estimate the high order marginals using this approach.

More precisely, we can define the restriction of f to the $[\![i,n]\!]\mathbb{S}_{n-1}$-coset as $f_i(\tau) = f([\![i,n]\!]\tau)$ (which is now a function on $\mathbb{S}_{n-1}$), and observing that the Fourier transform of f can be broken up as

$$\hat{f}_\lambda = \sum_{\sigma\in\mathbb{S}_n} f(\sigma)\rho_\lambda(\sigma) = \sum_{i=1}^n \sum_{\tau\in\mathbb{S}_{n-1}} f([\![i,n]\!]\tau)\rho_\lambda([\![i,n]\!]\tau) \tag{4}$$

$$= \sum_{i=1}^n \rho_\lambda([\![i,n]\!]) \sum_{\tau\in\mathbb{S}_{n-1}} f_i(\tau)\rho_\lambda(\tau) \tag{5}$$

The inner summation on the right of this equation looks almost like the Fourier transform of f_i over the smaller group $\mathbb{S}_{n-1}$, except that ρ_λ is an irreducible representation of $\mathbb{S}_n$ instead of $\mathbb{S}_{n-1}$. In fact, the $\rho_\lambda(\tau)$ matrices do form a representation of $\mathbb{S}_{n-1}$, but in general this representation is not irreducible. Maschke's theorem [14] tells us that we can express it in terms of the ρ_μ irreducible representations of $\mathbb{S}_{n-1}$ in the form

$$\rho_\lambda(\tau) = \bigoplus_{\mu\in\lambda\downarrow_{n-1}} \rho_\mu(\tau) \tag{6}$$

if a particular system of irreducible representations for $\mathbb{S}_n$, called *Young's Orthogonal Representation* (YOR) [14] is used. Here $\lambda\downarrow_{n-1}$ denotes the set of all partitions of $n-1$ dominated by λ, i.e., those partitions that we can get from λ by removing a single box from λ's diagram. Plugging (6) into (5) gives the relationship between $\hat{f}$ and $\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n$:

$$\hat{f}_\lambda = \sum_{i=1}^n \rho_\lambda([\![i,n]\!]) \bigoplus_{\mu\in\lambda\downarrow_{n-1}} (\hat{f}_i)_\mu \tag{7}$$

Such a formula can also be inverted to express $\hat{f}_1$, $\hat{f}_2$, $\cdots$, $\hat{f}_n$ in terms of $\hat{f}$:

$$(\hat{f}_i)_\mu = \frac{n-1}{n d_\mu} \sum_{\lambda\in\mu\uparrow^n} d_\lambda \rho_\lambda([\![i,n]\!])^{-1}(\hat{f}_\lambda)_\mu \tag{8}$$

where $\mu\uparrow^n$ is the set of all partitions of n that dominate μ, i.e., which can be derived from μ by adding a single box, and $(\hat{f}_\lambda)_\mu$ is the block of $\hat{f}_\lambda$ for μ.

The ideas in FFTs can be used to identify a restricted components. For the function given by $f(\sigma)I(\sigma(n) = n)$, we know that f only takes nontrivial values on $\mathbb{S}_{n-1}$. We have the following result to exactly calculate the Fourier coefficients for the function f restricted on $\mathbb{S}_{n-1}$.

Theorem 4 *Given a distribution f on $\mathbb{S}_n$ that only takes nontrivial values on $\mathbb{S}_{n-1}$, then function restricted on $\mathbb{S}_{n-1}$ has Fourier coefficients*

$$\widehat{(f|_{\mathbb{S}_{n-1}})}_\mu \propto \frac{1}{d_\mu} \sum_{\lambda\in\mu\uparrow^n} \sum_{j=1}^{z_{\lambda,\mu}} d_\lambda (\hat{f}_\lambda)_\mu \tag{9}$$

The operation involved in computing the Fourier coefficients for $f|_{\mathbb{S}_{n-1}}$ amounts to finding certain blocks within the $\hat{f}_\lambda$ matrices and adding them together weighted by the appropriate d_λ and d_μ^{-1} constants. We can upper bound the computational complexity by the total size $\sum_\lambda d_\lambda^2$ of the $\hat{f}_\lambda$ matrices. Since we only store the first few low-frequency Fourier components, the computing complexity is thus strongly polynomial.

In summary, the FFT-based method provides a scalable algorithm for computing Fourier coefficients for $f|_{\mathbb{S}_{n-1}}$. The representations that can appear in the computation result is also guaranteed, see proposition 5.

Proposition 5 *Given a set of Fourier coefficients for a distribution f over $\mathbb{S}_n$ whose order dominates $\lambda = (n-p, 1, \cdots, 1)$, the FFT-based method computes a set of Fourier coefficients whose order dominates $\lambda = (n-p-1, 1, 1, \cdots, 1)$ for $f|_{\mathbb{S}_{n-1}}$ with complexity $\mathcal{O}(\sum_\lambda d_\lambda^2)$.*

4 Marginal Probability Search

Based on previous sections, by exploiting the algebraic structure of $\mathbb{G}$, we now formally propose a new algorithm – the marginal-based search. This algorithm searches within the tree of the permutation group $\mathbb{S}_n$, from the root towards deeper levels, until the group $\mathbb{G}$ is fully identified. Starting from the root, we iteratively search and build deeper level nodes based on the marginal information. If the estimated marginal defined in (1) is no less than ϵ times m_k (see Theorem 1 from which we know the left-most node must have nonzero

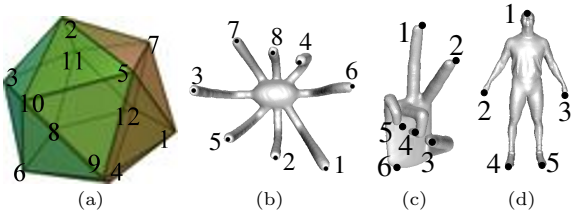

Figure 3: (a) Icosahedron. (b) Octopus. (c) Hand. (d) Human.

Algorithm 1 Marginal-Based Search

Input: A normalized similarity matrix S, ϵ.
Output: A list of automorphisms characterizing $\mathbb{G}$.
Procedure:
Initialize the tree T of $\mathbb{S}_n$
Estimate a set of Fourier coefficients $\{\hat{f}_\lambda\}$.
$k \leftarrow 0$
for k from 1 to n **do**
  Build child nodes for every node on the $(k-1)$-th level.
  for Each node on the k-th level **do**
    $\{\hat{f}_\mu\} \leftarrow$ Fourier coefficients of a distribution over $\mathbb{S}_{n-k}$.
    $m_k \leftarrow$ 0-th order Fourier coefficient of the left-most node.
    if The marginal of the node is less than ϵm_k **then**
      Prune the node.
    end if
  end for
  Prune any node which does not have any k-th level children.
end for

	Name	#Vertices	$\lvert\mathbb{G}\rvert$	Running time
	Tetrahedron	4	24	0.07s
	Hexahedron	8	48	1.07s
	Octahedron	6	48	0.61s
	Dodecahedron	20	120	131.2s
	Icosahedron	12	120	49.5s

Table 1: Detecting Symmetries of Regular Polyhedra

marginal), we keep this node; otherwise, we drop it off.[4] Such an iterative search algorithm is essentially doing a sparse pursuit of $\mathbb{G}$ within $\mathbb{S}_n$, which takes account of the group structures of $\mathbb{G}$, see algorithm 1 for the pseudo code.

Such a marginal-based search algorithm uses a set of Fourier coefficients to approximate a distribution over $\mathbb{S}_{n-k}$ for a node on the k-th level. However, since low-frequency Fourier coefficients characterize a smooth distribution over all the permutations. We typically observe that ϵ will be choosen to be very large, e.g., around 0.8. However, we still have theoretical guarantees about our approach.

Theorem 6 *Suppose S is the first order marginal probability matrix reconstructed from $\hat{f}_\mu$, when all S's are block diagonal dominant matrices*[5] *which indicate orbits partitioning, then Algorithm 1 can find all symmetries in $\mathbb{G}$.*

In the second order matrix case, we restrict our theorem to the special case that all points lie in the same orbit. If all points does not lie in the same orbit, then the inverse Fourier transform formula will put weights

[4]When we are done with building the k-th level nodes in the tree, we also prune the nodes in the current tree which do not have any k-th level children.

[5]Entries off blocks are strictly less than entries within blocks that lie in the same row and column

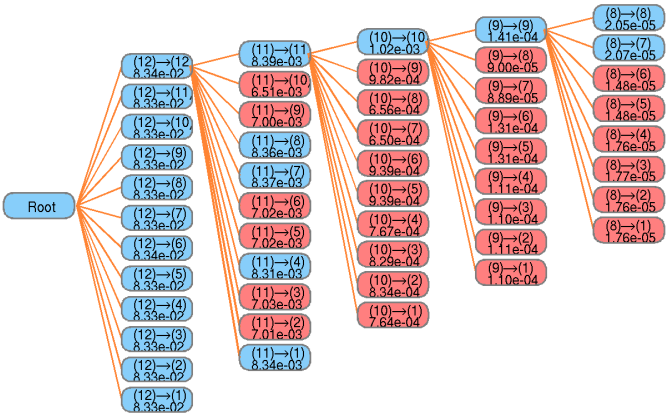

Figure 4: Searching for $\mathbb{G}$ using the marginal-based search algorithm for the perturbed icosahedron. We use a simplified notation for each node, for example, the node of $(11) \to (11)$ denote all permutations that maps $(11, 12) \to (11, 12)$. The blue nodes are those should be kept and the red nodes are those should be dropped off in the groundtruth.

	Dodecahedron	Icosahedron	Octopus	Human
Greedy	76.7%	90.8%	81.3%	100%
Eigen	75.8%	85.8%	75.0%	50%
Morgan	72.5%	87.5%	75.0%	50%
Fourier	79.1%	91.7%	87.5%	100%

Table 2: Accuracy of Different Approaches.

	Dodecahedron	Icosahedron	Octopus	Human
Greedy	248.71	85.09	5.89	0.59
Eigen	220.35	82.10	5.03	0.17
Morgan	218.92	79.50	5.18	0.17
Fourier	140.04	52.31	2.03	0.41

Table 3: Running Time of Different Approaches.

on different entries in S_{ij} which yield a distribution over permutations. However, such a distribution is still an ℓ_2 projection of the noised indicator distribution to the Fourier space.

5 Experiments

We test our algorithm on several examples, including regular polyhedra and 3D geometric objects. All the experiments are performed in Matlab on a regular desktop with 2.4GHz CPU and 3G RAM.

The first example is on detecting the symmetries for all 3D regular polyhedra. We first build a second order similarity matrix S using the continuous Gaussian kernel $s_{ij} = \exp(-|d_i - d_j|^2)$ where where d_i (d_j) is the distance between two points in the pair indexed by i (j). We normalize S and use it as a marginal probability matrix to estimate the symmetry group $\mathbb{G}$ by implementing Algorithm (1). As shown in table 1, our approach can detect the symmetry group $\mathbb{G}$ for all 3D regular polyhedra with reasonable running time. We note that brute force search for the symmetry group $\mathbb{G}$ for dodecahedron and icosahedron would be very difficult, since the sizes of the permutation groups are $20! \approx 2.43 \times 10^{18}$ and $12! \approx 4.79 \times 10^8$.

One benefit of the proposed approach of inferring the symmetry group $\mathbb{G}$ is that it can naturally deal with

noise. As an example, we randomly perturb the vertices of the icosahedron with certain magnitudes. After that, we repeat the previous experiment where we build the similarity matrix, normalize it to get a marginal probability matrix, and then estimate the symmetry group $\mathbb{G}$. It turns out that our approach can still find the symmetry group $\mathbb{G}$ for the perturbation with a relative magnitude up to 0.05 (the edge length is 1).

To demonstrate how the marginal-based search algorithm prunes the nodes in the tree, we show part of the tree implemented during our experiments in Figure 4-(a). As we can see from the figure, on the first, second and fifth level, the nodes that intersect with $\mathbb{G}$ have larger marginals than those that do not. Such a gap tends to be smaller at certain levels, such as level 3 and 4, thus it is very possible that we may include some nodes which do not intersect with $\mathbb{G}$ during the implementation of our algorithm. However, it turns out that when we look several levels down, those nodes will be dropped. The labeling of vertices of the icosahedron illustrated in this tree is shown in Figure 3-(a).

As another example, we detect the symmetries for the octopus, see Figure 3-(b). Unlike the regular polyhedra examples, the symmetry of the octopus can only be defined as isometries which preserve the geodesic distance, rather than the Euclidean distance. We use the fuzzy geodesics proposed in [15] which can be interpreted as a robust distance measure to get an effective similarity matrix between pairs. The later routines for inferring the group $\mathbb{G}$ are the same as in previous experiments. Though in this example the octopus is heavily deformed, we can still recover the dihedral group as its symmetry group.

Using the same technique of fuzzy geodesic metrics, we can get similarity matrix for many other 3D geometric models, such as the hand and human, as shown in Figure 3-(c,d). We can fully determine the 2-fold symmetries of the human model using our approach. However, for the heavily perturbed hand model, many permutations will be identified to be good ones, among which the permutations that have the highest values are still meaningful. For example, the top 2 permutations being identified are the identity and $(1, 2, 3, 4, 5, 6) \rightarrow (6, 3, 2, 4, 5, 1)$.

We finally compare our algorithm with another greedy heuristics [12] whose pruning criteria is based on the current maximum distortions. Several other algorithms such as principle eigenvector analysis (spectral analysis of the similarity matrices), the Morgan algorithm (an iterative procedure to estimate the orbit partitioning), and etc [1, 11] can be used as a pre-processing step which may reduce the size of the searching space. The comparison of accuracy (how many percentages of correct symmetries identified) and running time of different approaches are shown in Table 2 and 3. Throughout these experiments, we distort the geometric models so that it becomes difficult to recognize all the symmetries. Thus, we typically observe that eigenvector analysis and Morgan algorithm often make errors in identifying the orbit of the vertices. Whenever such an error occurs, it decreases the accuracy dramatically. The greedy heuristic algorithm typically has longer running time than our proposed Fourier approach.

6 Acknowledgement

The authors would like to thank the anonymous reviewers for valuable comments and suggestions. Leonidas Guibas and Xiaoye Jiang wish to acknowledge the support of NSF grants FODAVA 0808515, as well as ARO grant W911NF-10-1-0037.

References

[1] P. J. Cameron. *Permutation Groups*. London Mathematical Society, 1999.

[2] S. Datta, N. Limaye, P. Nimbhorkar, T. Thierauf, and F. Wagner. Planar graph isomorphism is in logspace. In *IEEE Conference on Computational Complexity*, 2009.

[3] P. Diaconis. *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics, 1988.

[4] T. Funkhouser and P. Shilane. Partial matching of 3D shapes with priority-driven search. In *Proceedings of SGP*, 2006.

[5] J. Huang, C. Guestrin, and L. J. Guibas. Fourier theoretic probabilistic inference over permutations. *Journal of Machine Learning Reserach*, 2009.

[6] G. D. James. *The Representation Theory of the Symmetric Groups*. Springer-Verlag, 1978.

[7] N. J. Mitra, L. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. In *ACM Transactions on Graphics*, 2006.

[8] F. Murnaghan. The analysis of the kronecker product of irreducible representations of the symmetric group. *American Journal of Mathematics*, 1938.

[9] M. Ovsjanikov, Q. Merigot, F. Memoli, and L. Guibas. One point isometric matching with the heat kernel. In *Proceedings of SGP*, 2010.

[10] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser. A planar-reflective symmetry transform for 3D shapes. In *Proceedings of SIGGRAPH*, 2006.

[11] I. Ponomarenko. Graph isomorphism problem and schurian algebras. *preprint*, 1994.

[12] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Symmetries of non-rigid shapes. In *Proceedings of ICCV*, 2007.

[13] R. Read and D. Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1977.

[14] B. Sagan. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*. Springer-Verlage, 2001.

[15] J. Sun, X. Chen, and T. Funkhouser. Fuzzy geodesics and consistent sparse correspondences for deformable shapes. In *Proceedings of SGP*, 2010.

[16] L. Weinberg. On the maximum order of the automorphism group of a planar triply connected graph. *SIAM Journal on Applied Mathematics*, 1966.