Group-theoretic Algorithms for Matching Problems with Applications to Computer Vision

by

Deepti Pachauri

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2015

To my parents.

ACKNOWLEDGMENTS

It has been a long journey from first concept to the completed thesis. During this journey, I worked with great number of people whose contribution in different ways, deserve special mention. I take this opportunity to acknowledge them and extend my sincere gratitude for helping me going through all those years and much more. I apologize in advance to anyone whom I have neglected to mention.

First and foremost, I would like to thank my advisor Vikas Singh for providing me with the opportunity to complete my PhD thesis at the University of Wisconsin-Madison. I appreciate his valuable feedback, advice, and encouragement to make my Phd experience productive and stimulating. Despite his busy schedule, he was always there to listen my half-baked ideas patiently and help me think broadly which make him a great mentor. More importantly, his principles and values have left an indelible impression in my heart. It has been an honor to be his PhD student. No amount of words can really be adequate in expressing my sincere thanks to him.

I would like to thank my wonderful committee members Jerry Zhu, Rob Nowak, Chuck Dyer and Risi Kondor for their guidance and valuable feedback on my thesis. It was a delight to attend Chuck's offering of CS540 in my first semester at UW Madison. That course was my first brush with machine learning, and was instrumental in both familiarizing me with the area, and for me to realize my interest in the subject. I am very grateful for his encouragement in that class. I was extremely fortunate to have the mentorship and support of Risi. His research provided crucial foundational material for much of my research. The work I did with him, have made a lasting influence on shaping me as a researcher.

I want to thank present and past members of my lab with whom I have had the pleasure to work with or alongside: Jia, Maxwell, Hynwoo,

Won, Vamsi, Sathya, Gregory, Kamiya and Chris. I also recognize the never ending encouragement and support from Nagesh, who has always been an integral part of our lab. Thanks to Angela Thorp who made all administrative issues during my stay at UW very easy. She was one of the first friendly faces when I came to UW Madison and has always been a tremendous help.

Special thanks to Borna Ghosh and Sathya for reading sections of my thesis and giving me valuable comments. Thanks to Debbie Chasman for coffee sessions and constant support. Of course, I cannot get through this section without thanking Akhila Vasan for her constant reminders to stay focused, and Stuti Jha for endless conversations and going through our graduate journey together. I would also like to thank many of my nerdy geeky friends spread across the globe who inspired me Apar, Shivanand, Swami, Sanjib, Anu, Bushra, Ajay, Nutan and many others. I also thank Akshar and Nishtha for their skill in spreading happiness on those scientifically dark days and making them enjoyable.

Of course no acknowledgments would be complete without mentioning my parents, without their constant love, support, and sacrifices, I would have never made this far. I am grateful to them for the "smart genes" they passed on to me and teaching me to strive, to never give up, to question, to take risk, to follow dreams and to fly! I am thankful to my parents-in-law, and extended family for understanding the expectations of a PhD, and for giving me that extra leeway.

A very special thanks to "The Husband" Vikash, I am very fortunate to have your love and support, as well as your patient ear whenever I am feeling down.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Matching one set of objects to another is a fundamental problem in computer science. In computer vision it arises in the context of finding the correspondences between multiple images of the same scene taken from different viewpoints. These problems often reduce to some form of combinatorial search problem, various classes of which are polynomial time solvable whereas many others are NP-hard. The fundamental difficulty in solving matching problems is that the solution space is combinatorial in nature and exponential in size. This thesis takes a **group theoretic** approach for these intractable problems. We are interested in exploiting the algebraic structure of the solution of a matching problem i.e., a permutation. A permutation of order n is a candidate in the $\mathbb{S}_n$, called the **Symmetric Group** of degree n. The high degree of regularity of the symmetric group allow us to unleash a wide range of mathematical tools on matching problems.

In particular, this dissertation studies real-world matching problems that typically occur in computer vision from the following perspectives: *1*) How to leverage non-commutative harmonic analysis on $\mathbb{S}_n$ to learn which features are important to identify good correspondences quickly between image pairs (when all image pairs are extracted from a specific domain); *2*) How to extend the notion of regularity in $\mathbb{S}_n$ to match landmark points across multiple images robustly; *3*) How to leverage the encoded symmetries in $\mathbb{S}_n$ to match unordered images in the presence of large perturbation. Further, techniques developed in this thesis are fully general and equally applicable to matching problems in other domains.

1 INTRODUCTION

Several important computer vision applications, such as image registration (Shen and Davatzikos, 2002), recognition (Duan et al., 2012; Demirci et al., 2006), stereo (Goesele et al., 2007), shape matching (Berg et al., 2005; Petterson et al., 2009), and structure from motion (SFM) (Agarwal et al., 2011; Simon et al., 2007), involve correspondence tasks in different forms. In its most general formulation, the image correspondence problem is expressed as a pairwise matching problem. That is, matching interest points[1] in one image to the interest points in second image. But pairwise matching has numerous applications outside vision, so it is not surprising that matching is among the most well studied topics in combinatorial optimization. This research has led to mature implementations which facilitate numerous applications, including the specific examples listed above.

A major challenge in real-world vision tasks is to find good matchings in the presence of background clutter, occlusion, multiple common objects and large deformations. Therefore local appearance of interest points as well as appearance of group of interest points are often considered in the pairwise formulations (Forsyth and Ponce, 2003; Conte et al., 2004; Hancock and Wilson, 2009). However, many practical computer vision applications involve more general matching problems such as matching multiple images (Snavely et al., 2008; Rao et al., 1993; Demirci et al., 2006). Therefore, effective and efficient matching algorithms beyond conventional pairwise algorithms are required.

In this thesis, we take a broad perspective on matching problems and study them from a new *group theoretic* standpoint. Our primary focus domain is computer vision, however, techniques developed in this thesis

[1] In the computer vision literature terms like interest points, key points, landmarks are used interchangeably.

are fully general and equally applicable to matching problems in domains other than computer vision.

1.1 Background

The theoretical framework for matching problems goes back to the eighteenth century (Schrjiver, 2002), but it was not until 1941 when a precise mathematical description of the problem became available (Hitchcock, 1941). The formal definition of matching problem is

> *Given two sets, $X = \{x_1, \cdots, x_n\}$ and $Y = \{y_1, \cdots, y_n\}$ of n objects[2] each. Find a bijection $X \longrightarrow Y$ such that each $x_i \in X$ is exactly matched with one $y_j \in Y$ in the best possible way.*

The problem is called linear assignment problem when the "best possible" implies maximizing the similarity score for all assignments, i.e.,

$$\arg\max_{\sigma \in \mathbb{S}_n} \sum_{i=1}^{n} Q_{i,\sigma(i)}, \tag{1.1}$$

where $Q_{i,j}$ is the assignment matrix that indicates the similarity score of matching objects x_i and y_j, and $\mathbb{S}_n$ is the set of all permutations of $\{1, 2, \ldots, n\}$. In 1955, Kuhn published the first combinatorial method for the linear assignment problem which solved this problem in polynomial time $\mathcal{O}(n^3)$ (Kuhn, 1955). The area of research received intensive attention in 1950 due to the development of linear and integer programming (Dantzig, 1951).

In computer vision, instances of linear assignment have a long history where each image is treated as a collection of attributed interest points. These points are specific structures in the image such as corners, edges

[2]Unless otherwise specified, we consider the pairwise matching problem that has $|X| = |Y|$ in our discussion. Note problems with $|X| \neq |Y|$ can be transformed to an analogous problem where $|X| = |Y|$.

or blobs, Figure 1.1. Intuitively, each image is represented using local

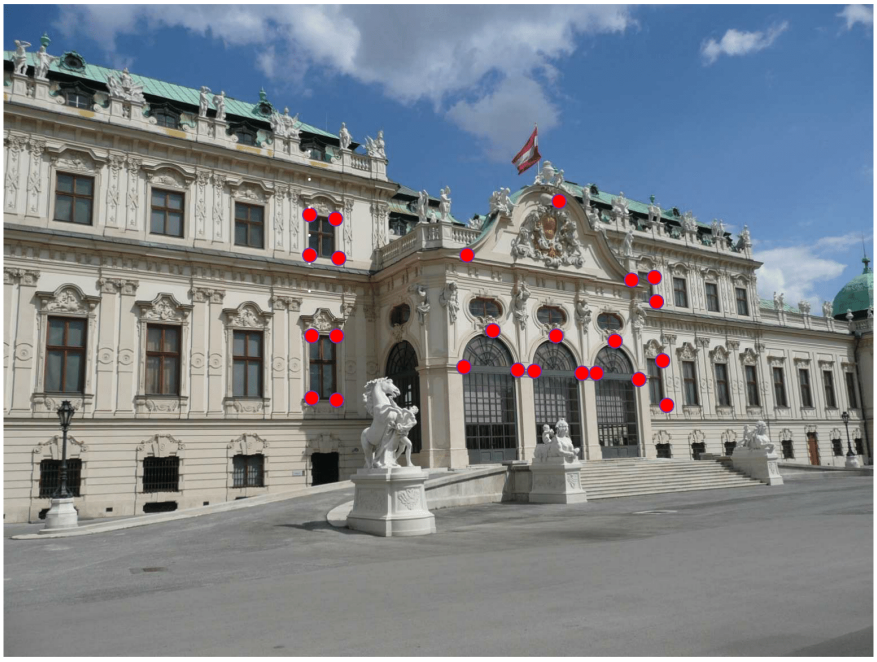


Figure 1.1: Some interest points highlighted in red.

appearance of the image regions around interest points (Lowe, 2004; Mikolajczyk and Schmid, 2004); or using the notion of relative appearance of interest points with respect to others (Belongie et al., 2002); or some edge information (Fischler and Elschlager, 1973). This local appearance information, i.e., local descriptors can be represented in different ways. For example – a vector of filter outputs. For each point in the first image X, the image correspondence problem tries to find the "best matching" point in the second image Y. The similarity score $Q_{i,j}$ for matching a point x_i on the first image to a point y_j on the second image, represents the local

appearance similarity between points x_i and y_j;

$$Q_{i,j} = -d(x_i, y_j) \quad \forall x_i \in X,\ y_j \in Y. \tag{1.2}$$

Here $d(\cdot, \cdot)$ is a valid distance metric on the vector space associated with the local descriptors. The goal is to maximize the overall similarity score between image X and image Y by matching image regions around interest points in the two images, Figure 1.2.

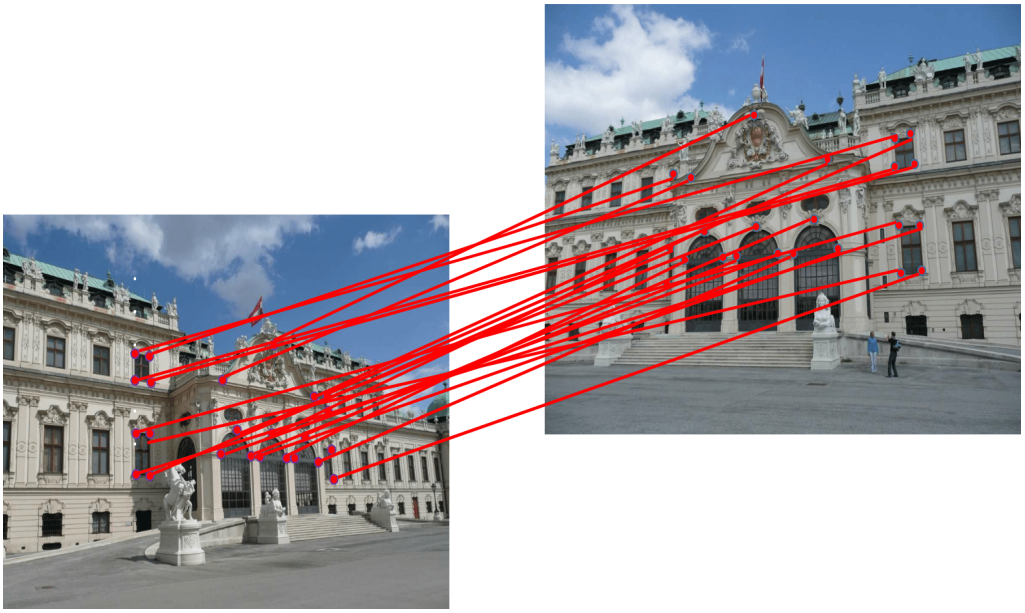


Figure 1.2: Linear assignment of interest points for a pair of images of the same scene.

The linear assignment formulation is appealing largely because it admits an efficient solution (Lee and Orlin, 1994; Kuhn, 1955; Jonker and Volgenant, 1987; Huang and Jebara, 2011), but is limited to first-order relationships between X and Y. The first-order matching model leads to unsatisfactory matching results in the presence of ambiguities or non-discriminative Q. In these situations, higher-order geometric relationships

between the tuples of objects in each set are considered useful in helping the matching model find the correct correspondences. For example, a matching objective that takes both first and second-order relationships between the objects/features into account cannot be handled within a linear assignment formulation – it requires moving to the so-called quadratic assignment problem (QAP). QAP is also famously known as "graph matching problem" which was first introduced by Koopmans and Beckmann in 1957 in the context of a plant location problem (Koopmans and Beckmann, 1957). Intuitively, we may view it as a problem of matching two graphs. One graph is extracted from the set X where elements of X are nodes of the graph and edges correspond to the second-order relationships between the elements of X. The second graph comes from the second-order relationships between the elements of set Y. Given a graph the edge information can be represented as an *adjacency matrix*, denoted as A. The adjacency matrix of a graph is a $n \times n$ matrix where the entry $A_{i,j}$ correspond to the edge between node i and node j of the graph. In a simple case, the adjacency matrix is a $0-1$ matrix where $A_{i,j} = 1$ means node i is connected to node j and $A_{i,j} = 0$ means otherwise. In case of weighted graphs, $A_{i,j} \in R$ and correspond to the structural similarities between node i and node j.

In terms of adjacency matrices, the matching of X and Y can be formulated as standard QAP

$$\arg\max_{\sigma \in \mathbb{S}_n} \sum_{i,j=1}^{n} {A^X}_{\sigma(i),\sigma(j)} {A^Y}_{i,j} \ . \tag{1.3}$$

The objective is to match A^X and A^Y to maximize the overlap between them. The inclusion of second-order relations makes the problem intractable. From the practical point of view, the QAP remains one of the hardest problems in combinatorial optimization and no fast methods are known for it. Existing algorithms are mostly enumeration methods like branch

and bound (Gilmore, 1962; Lawler, 1963; Kondor, 2010) and cutting plane algorithms (Bazaraa and Sherali, 1980; Balas and Mazzola, 1980; Kaufman and Broeckx, 1978). For more details, see (Burkard, 1984).

Despite the difficulties described above, many correspondence problems in vision naturally appear in the form of graph matching problems (Conte et al., 2004). This formulation is well suited for vision applications because graph representations have a strong capacity to encode both the local appearance information – through the notion of interest points, and the global information – through the second-order geometric relationships between interest points. For a given pair of images, the two graphs for matching are extracted from the input images; a graph corresponding to an image is comprised of *1*) interest points on the image which serve as the vertices of the graph; and *2*) edges which denote the second-order relationships between the corresponding interest points. For example – consider the object matching task shown in Figure 1.3. In this example, the object of interest, i.e., touring bike, is identified by 6 interest points which correspond to the frame of the bike and can be represented as a graph (blue edges). By only considering local information, identifying a correct matching is not possible in this example because the background is not controlled and even sophisticated descriptors may go wrong. Similar situations arise in other applications such as activity recognition (Messing et al., 2009) which require the ability to match interest points efficiently in the presence of background clutter. What seems to be appropriate for this situation is a graph matching formulation (QAP) which will enforce pairwise geometric relationships on the assignments of interest points. Correct assignments should preserve the geometry of the bike frame in the two images and solve the correspondence problem effectively.

For over five decades, researchers have extensively investigated matching problem in vision applications by *1*) using richer structural information, i.e., more informative Q in Equation (1.1) and (A, A') in Equation (1.3)

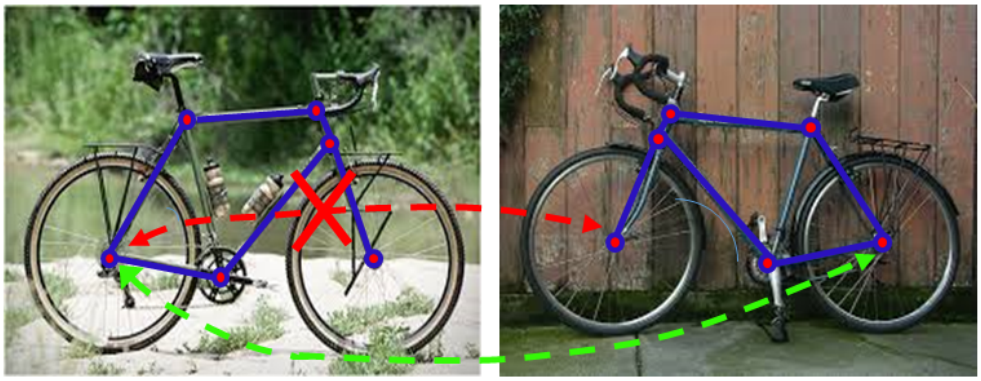


Figure 1.3: Pairwise assignment of interest points using second order geometric relationship.

(Belongie et al., 2002); and 2) by proposing various relaxations to solve the original NP-hard graph matching problem (Leordeanu and Hebert, 2005; Duchenne et al., 2011; Torr, 2003). Nonetheless, how to estimate these Q, A and A', is itself a complex question. Over the years, computer vision has witnessed extensive research aimed at providing suitable interpretations of structure in natural images (Shi and Tomasi, 1994; Harris and Stephens, 1988; Hauagge and Snavely, 2012). Each of these methods provide different proposals for Q and A, and therefore it is difficult to know which choice is the most appropriate to secure a good matching result for a certain vision application. In Chapter 3, we look at this problem and propose a domain driven framework to learn powerful representations for graph matching problems drawn from a specific domain/setting. In particular,

- **How to learn good descriptors to solve domain specific graph matching problems efficiently?**

  The notion of *conditions* reflects properties of the application under study. By restricting the focus to a *specific* domain and tuning weights that best reflect practical considerations in that application,

a less sophisticated approach may still be able to obtain good quality solutions in a fixed amount of time or memory (Xu et al., 2007). For instance consider the task of aligning images from two cameras capturing an indoor scene. The two cameras are placed in such a way that the view point differs significantly between the images from one camera and the images from the second camera. As mentioned earlier, one can extract different types of descriptors from the image data, each of which correspond to different type of information and use them for pattern matching. The relative placement of the two cameras suggest that one should use a descriptor which is robust to the view point changes for matching image pairs in this scenario. Indeed in most practical cases one has no access to the exact definition of "condition". However, when some image pairs with "ground truth" correspondences are available, we can "learn" what is important to a domain specific matching problem.

1.2 Beyond Pairwise

Various applications in computer vision operate with multiple images and not just two images. For instance in surveillance systems, objects are tracked across multiple video frames (Wu et al., 2013). Structure from motion is another application where each landmark is tracked in multiple images to obtain the visibility graph (Wilson and Snavely, 2013). Using pairwise matching algorithms in such scenarios lead to unsatisfactory results. For example, pairwise matching procedures produce incorrect landmark correspondences in SfM in the presence of ambiguous structures. Existing pairwise approaches do not provide any useful insight towards extensions that can handle multiple images. For example:

- **How to solve a multi-way matching problem?**

Multi-way matching problem is a matching of not just two, but m different sets of objects to each other. Many applications such as multi-sensor surveillance (Rao et al., 1993) and multi-target tracking (Huang et al., 2009; Kondor et al., 2007) are instantiations of this problem. Similarly, many computer vision applications require matching interest points across large set of images. Though this problem is a most natural generalization of linear assignment problem which admits efficient solutions, multi-way matching is a NP-hard problem and cannot be solved efficiently. At present, multi-way matching is solved by matching pair of images in series. But the fact that image descriptors are noisy and a single error in the sequence will typically create a large number of erroneous pairwise matches, makes such a strategy problematic in numerous cases.

- **How do the perturbations affect matching scenarios?**

  A reasonable continuation of multi-way matching is the multi-way matching of the unordered data set. Here, unordered data means we have no prior knowledge whether all examples/items should indeed be matched with each other or not. In other words, one should first find consistent clusters of matching instances and then approach the multi-way matching problem for each cluster independently. Large scale structure from motion applications in computer vision which utilize images from the internet is a practical example of multi-way matching problem with unordered data sets. In particular, this application utilizes several hundreds of thousands of images from the internet to reconstruct 3D models of a scene. In the 3D reconstruction pipeline, an important step is to first identify which image shares a common baseline with which image. Existing methods extract this information from the "local appearance similarity" of interest points between pair of images. Unfortunately, when the physical scene contains large repetitive structure such as urban scenes and historic

monuments, automatic SfM methods break down (Roberts et al., 2011; Wilson and Snavely, 2013) due to the ambiguous appearance of interest points. Issues such as appearance ambiguity, make the multi-way matching problem far more difficult. All these considerations point towards the need for a more principled way to deal with unordered multi-way matching problems.

The universality of aforementioned matching scenarios in various applications and the gap in the literature, motivated this thesis. We introduce a new approach for matching problems – the algebra approach. We describe the rationale for this new approach in the next section.

1.3 Rationale for the Group Theoretic approach

The fundamental difficulty in solving matching problems is that the set over which we need to optimize the matching score – the set of all permutations is combinatorial in nature and exponential in size. We make an important observation that this set is not structure-less, but rather constitutes a set with binary operation. More precisely, it is a **group** called the symmetric group $\mathbb{S}_n$. This means that with respect to the natural notion of multiplication, which is $(\sigma_2\sigma_1)(i) = \sigma_2(\sigma_1(i))$, permutations in $\mathbb{S}_n$ satisfy the following axioms:

1. $\sigma_2\sigma_1 \in \mathbb{S}_n$ (closure);
2. $\sigma_3(\sigma_2\sigma_1) = (\sigma_3\sigma_2)\sigma_1$ (associativity);
3. There is an identity $e \in \mathbb{S}_n$ such that $\sigma_1 e = \sigma_1$;
4. Every σ has an inverse $\sigma^{-1} \in \mathbb{S}_n$ such that $\sigma^{-1}\sigma = \sigma\sigma^{-1} = e$.

Remarkably, these axioms are sufficient to give $\mathbb{S}_n$ a beautiful structure, which allow to encode symmetries (associated with various invariant properties of a space and functions) and reversible transformations (that preserve the invariant properties).

In this thesis, we apply this group theoretic interpretation of the original combinatorial space to solve the matching problems. This view of $\mathbb{S}_n$ allows us to use an entire spectrum of tools from abstract algebra including representation theory and non-commutative harmonic analysis, to develop efficient algorithms. The real promise of this new approach lies in its generality; by setting matching problems in a broader algebraic framework, it has the potential to serve as a basis for developing new matching algorithms that exploit the high degree of regularity of the solution space and are better informed by the characteristics of the underlying task. A fundamental influence of this thesis is that it provides an interesting insight for a wider class of problems. We note that the new algorithms proposed in this thesis, can be easily extended to other applications which may involve other compact groups and transformations.

1.4 Contributions

Specific contributions of this thesis are as follows:

- We developed a supervised learning algorithm to solve graph matching problems specific to a domain of interest. In particular, we used harmonic analysis on $\mathbb{S}_n$ to learn "effective" features which can provide the good correspondences between graph pairs even when a less sophisticated method is used for matching.

- We developed an unsupervised algorithm to solve multi-way matching problem for an ordered data set and evaluated its performance on various stereo matching data sets. Further, we designed a semi-supervised version of this algorithm which can utilize nominal user supervision to solve multi-way matching problems in the presence of less informative features.

- We introduced a new method to solve matching problems for unordered data set in the presence of large perturbations. We developed an invariance based affinity measure to obtain the certificate of association between instances. We considered the structure of motion application in our experiments when large repetitive structures are present in the scene, and demonstrated very good results. Further, we considered the scene summarization problem and used the new metric to find the representative images.

1.5 Outline

The rest of the thesis is organized as follows. In the first part of the thesis, we review the important concepts of non-commutative group that are leveraged in this thesis (Chapter 2). In the second part, from Chapter 3-5, we develop effective algorithms for matching problems and demonstrate impressive experimental results on various computer vision applications. We demonstrate a supervised learning solution for graph matching problems in Chapter 3. We generalize the matching problems to matching not just two, but m different sets in Chapter 4. In the last part, we explore the implications of the invariance property of functions defined on the symmetric group (Chapter 5). We conclude and discuss future research directions in Chapter 6.

In this chapter, we describe the basic concepts of group theory which we will utilize for the later chapters. We first present the standard definition of a group and its properties. Then, we review these concepts and define notations specifically for the symmetric group, denoted by $\mathbb{S}_n$. We will then describe briefly the algebraic structure associated with $\mathbb{S}_n$ which will be fundamental in characterizing the solution space of matching problems. A more comprehensive treatment of group theory is available in (Burnside, 1911; Weyl, 1997; James, 1987; Diaconis, 1988; Terras, 1999; Kondor, 2008).

2.1 The Definition and Properties of a Group

A **group** is a set with a binary operation, denoted as $(G, *)$, satisfying following axioms:

1. **Closure:** $\forall g_i, g_j \in G : g_i * g_j \in G$
2. **Associativity:** $\forall g_i, g_j, g_k \in G : g_i * (g_j * g_k) = (g_i * g_j) * g_k$
3. **Identity:** $\exists e \in G : \forall g_i \in G : e * g_i = g_i * e = g_i$
4. **Inverse:** $\forall g_i \in G : \exists\, g_j \in G : g_i * g_j = g_j * g_i = e$

Here the symbol $(*)$ denotes the binary operation on the set G: a map $G \times G \to G$. Sometimes, this map is also called *multiplication*. In general, the property $g_i * g_j = g_j * g_i$ may not be true, when a binary operation satisfies this property, then the group $(G, *)$ is called *commutative*, otherwise it is *non-commutative*. Given a group structure, one can study its substructures called **subgroups**. A subgroup of a group is a subset of the underlying set, that naturally inherits the group structure from the original group.

We can gain a better understanding of these concepts by recalling the definition of a group. That is, *a group is a collection of reversible transformations acting on various structures such as spaces and sets, with certain specific properties*. For example, the collection of transformations should have an identity transformation which preserves the specific structure, and each transformation also have its reverse transformation in the collection which is important to *cancel* the action of the original transformation. Group axioms severely restrict the nature of transformations that can be defined on a structure, and reveal the invariant properties of the structure under the action of these transformations, i.e., under the action of the group. One such action is the *transitive* group action; the action of G on a set S is transitive if for any $x, y \in S \; \exists \; g \in G$ such that $g(x) = y$. When a subgroup acts on G then it decomposes G into a disjoint union of subsets. This action is transitive and the disjoint union is called the **homogeneous space** of G with respect to the subgroup. Each disjoint set is called the **coset** of the subgroup. More generally, when G acts transitively on a space then the space is called a homogeneous space with respect to G.

A *representation* ρ of a finite group G is a linear map on a vector space V such that every group element $g_i \in G$ is assigned an invertible matrix $\rho(g_i)$ and $\rho(g_i)\ \rho(g_j) = \rho(g_i * g_j)$ holds for every $g_j \in G$. This also implies $\rho(g_i^{-1}) = \rho^{-1}(g_i)$ and $\rho(e) = I$ where I is the identity matrix in V. Thus, ρ is a homomorphism from $G \rightarrow GL(V)$, where $GL(V)$ is the set of linear maps $V \rightarrow V$, (Chen et al., 1989; Serre, 1977). The dimension of V is denoted by d_ρ and called the dimension of ρ. The ρ characterization of G facilitates the understanding of the two alternative definitions of a group – "a set with binary operation" and "collection of transformations acting on a structure". Firstly, the ρ representation makes the binary operation $(G, *)$ explicit, i.e., the binary operation corresponds to a matrix multiplication between the elements of the set G. Secondly, considering group actions as transformations in the vector space simplifies the study of the structure in

question.

Two different d_ρ dimensional representations ρ_1 and ρ_2 of G are said to be *equivalent* if there exists a non-singular matrix T, such that

$$T^{-1}\rho_1(g)T = \rho_2 \quad \forall g \in G. \tag{2.1}$$

The matrix T represents a simple basis transformation of the vector space V. If there is a subspace W in V which is stable under G, i.e., all vectors $v \in W$ are transformed by G into vectors also contained in W,

$$\rho(g)\, W \subseteq W \quad \forall\, g \in G\,.$$

Then V is said to be *decomposable* and ρ restricted to W is called the *subrepresentation*. A representation ρ is said to be *reducible* if there exists a basis transformation T which simultaneously block diagonalizes ρ

$$T^{-1}\rho(g)T = \rho_1(g) \oplus \rho_2(g) \oplus \cdots \quad \forall g \in G, \tag{2.2}$$

where $\rho_1, \rho_2 \cdots$ are representations of dimensions smaller than d_ρ, such that

$$d_{\rho_1} + d_{\rho_2} + \cdots = d_\rho. \tag{2.3}$$

If ρ cannot be reduced by the procedure described above, i.e., no stable W can be found, then ρ is called the *irreducible* representation of G. Irreducible representations or irreps are the building blocks of representations, and any matrix valued representation of G can be uniquely written as a tensor sum of irreps as in (2.2). We further explain these concepts using symmetric group as an example.

2.2 The Symmetric Group

The symmetric group $\mathbb{S}_n$ is a set of permutations

$$\mathbb{S}_n = \{\sigma_1, \sigma_2, \cdots, \sigma_p\}, \tag{2.4}$$

where each element $\sigma_i \in \mathbb{S}_n$ is a one-to-one mapping of n letters into itself. For instance, suppose we take the set of integers $\{1, 2, \ldots, n\}$. The permutations of this set forms the group $\mathbb{S}_n$ of order n and cardinality $n!$. The binary operation[1] which characterizes this set as a *group* is the function composition. In particular, the composition of two permutations σ_i and σ_j means the following:

$$(\sigma_i\sigma_j)(k) = \sigma_i(\sigma_j(k)) \; . \tag{2.5}$$

For example, consider $\sigma_i, \sigma_j \in \mathbb{S}_4$ which maps $\{1, 2, 3, 4\}$ as

$$\begin{aligned} \sigma_i(1) &= 3 & \qquad \sigma_j(1) &= 2 \\ \sigma_i(2) &= 1 & \qquad \sigma_j(2) &= 3 \\ \sigma_i(3) &= 4 & \qquad \sigma_j(3) &= 4 \\ \sigma_i(4) &= 2 & \qquad \sigma_j(4) &= 1 \end{aligned}$$

then the composition $(\sigma_i\sigma_j)$ implies

$$\begin{aligned} (\sigma_i\sigma_j)\,(1) &= \sigma_i(\sigma_j(1)) = \sigma_i(2) = 1 \\ (\sigma_i\sigma_j)\,(2) &= \sigma_i(\sigma_j(2)) = \sigma_i(3) = 4 \\ (\sigma_i\sigma_j)\,(3) &= \sigma_i(\sigma_j(3)) = \sigma_i(4) = 2 \\ (\sigma_i\sigma_j)\,(4) &= \sigma_i(\sigma_j(4)) = \sigma_i(1) = 3 \; . \end{aligned}$$

With respect to this definition of function composition, elements in $\mathbb{S}_n$ satisfy following properties:

[1] Dropping $(*)$ symbol for notational simplicity.

1. The product of two permutations is also a permutation

$$\sigma_i \sigma_j \in \mathbb{S}_n \qquad \forall\ \sigma_i, \sigma_j \in \mathbb{S}_n \tag{2.6}$$

2. The product of permutations is associative

$$(\sigma_i \sigma_j)\sigma_k = \sigma_i(\sigma_j \sigma_k) \qquad \forall\ \sigma_i, \sigma_j, \sigma_k \in \mathbb{S}_n \tag{2.7}$$

3. There is an identity permutation e in $\mathbb{S}_n$ such that

$$\sigma_i\ e = e\ \sigma_i = \sigma_i \qquad \forall\ \sigma_i \in \mathbb{S}_n \tag{2.8}$$

4. Every permutation has an inverse permutation in $\mathbb{S}_n$ such that

$$\sigma_i \sigma_i^{-1} = \sigma_i^{-1} \sigma_i = e \qquad \forall\ \sigma_i \in \mathbb{S}_n \tag{2.9}$$

The product of permutations is not *commutative*, i.e., $\sigma_i \sigma_j \neq \sigma_j \sigma_i\ \forall\ \sigma_i, \sigma_j \in \mathbb{S}_n$. Therefore $\mathbb{S}_n$ is a **non-commutative** group.

2.2.1 Notations of $\mathbb{S}_n$

There are various popular notations for representing $\sigma \in \mathbb{S}_n$, one of which is the Cauchy's *two-line* notation. This notation list all objects in top row and their permuted image in the second row.

$$\sigma := \begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix}\ .$$

For example, a σ that maps $\sigma(1) = 2, \sigma(2) = 5, \sigma(3) = 3, \sigma(4) = 4, \sigma(5) = 1$ can be written as

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 3 & 4 & 1 \end{pmatrix}\ . \tag{2.10}$$

Another common notation is the *cycle* notation. The cyclic notation $(i\ j\ k \cdots l)$ refers to a mapping in which each object is followed by its image, and the image of the last object is given by the first object, i.e.,

$$(i\ j\ k \cdots l) := i \to j,\ j \to k,\ \cdots,\ l \to i\ .$$

The number of objects between $(\cdots)$ is the length of the cycle. All permutations can be expressed as a product of disjoint cycles. Two cycles are called disjoint if their intersection is an empty set. The cycle notation for the σ in (2.10) is $(1\ 2\ 5)\ (3)\ (4)$. Here $(1\ 2\ 5)$ is a 3-cycle and $\{(3),\ (4)\}$ are 1-cycles [2]. In this example, these cycles are disjoint.

The cycle notation is not unique since every cycle can be written differently, for instance $(1\ 2\ 5)\ (3)\ (4)$ is same as $(3)\ (4)\ (5\ 1\ 2)$. The expression is unique up to the length of the cycles, i.e., *cycle type*. The cycle type of a permutation is the lengths of the cycles in the decomposition of the permutation. So $(1\ 2\ 5)\ (3)\ (4) \in \mathbb{S}_5$ has cycle type $(3, 1, 1)$. Note that the sum of all the cycles in the decomposition is n. Thus, the cycle type of a permutation is an integer **partition** of n. A partition λ of n (denoted $\lambda \vdash n$) is a k-tuple $\lambda = (\lambda_1, ..., \lambda_k)$ of weakly decreasing positive integers $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i = n$. For example, $n = 5$ is characterized by seven integer partitions and every $\sigma \in \mathbb{S}_5$ has one of the seven cycle types [3].

$$(5);\ (4,1);\ (3,2);\ (3,1,1);\ (2,2,1);\ (2,1,1,1);\ (1,1,1,1,1)\ .$$

The cycle type is exclusively determined by the permutation. Two permutations $\sigma, \sigma' \in \mathbb{S}_n$ are *equivalent*, $\sigma \sim \sigma'$, if they have the same cycle type because the cycle structure is invariant to the operation $\tau^{-1}\sigma'\tau\ \ \forall\ \tau \in \mathbb{S}_n$. This means, the *conjugacy classes* of $\mathbb{S}_n$ are realized over integer partitions of n. Consider an element $(2\ 3)$ in $\mathbb{S}_3$. The cycle type of this element is $(2, 1)$.

[2] In fact 1-cycle is the identity permutation and often dropped from the cycle notation.

[3] A cycle type (partition) is denoted in parentheses separated by a comma i.e., $(\cdot\ ,\ \cdot)$. A cycle is denoted in parentheses separated by a space i.e., $(\cdot\ \ \cdot)$.

The definition of conjugacy predicts that any conjugation will preserve its cycle type. For example,

$$(1\ 2\ 3)(2\ 3)(1\ 3\ 2) = (1\ 3)\ .$$

In this example, the conjugating permutation $(1\ 3\ 2)$ has a cycle type (3) but the cycle type of the resultant permutation $(1\ 3)$ is same as $(2\ 3)$ i.e., $(2, 1)$.

A cycle of length two is particularly interesting which is called a *transposition*. Each cycle can be decomposed into products of transpositions, for instance $(1\ 2\ 5) = (1\ 2)(2\ 5)$. Moreover, each transposition can be uniquely factored into a special set of transpositions, called *adjacent transposition*, meaning the transposition of type $\tau_k = (k\ \ k+1)$ which interchanges k with $k+1$ and leaves everything else fixed. Any arbitrary transposition $(i\ \ j)$ can be written as:

$$(i\ \ j) = \tau_i\tau_{i+1}\cdots\tau_{j-1}\tau_{j-2}\cdots\tau_{i+1}\tau_i\ .$$

For example, the transposition $(2\ 5)$ factors as

$$\begin{aligned}(2\ \ 5) &= \tau_2\ \tau_3\ \tau_4\ \tau_3\ \tau_2 \\ &= (2\ 3)(3\ 4)(4\ 5)(3\ 4)(2\ 3)\ .\end{aligned}$$

Such factorization is interesting because adjacent transpositions themselves generate the whole $\mathbb{S}_n$. For example, $\mathbb{S}_3$ can be written as

$$\begin{aligned}(1\ 3) &= (1\ 2)(2\ 3)(1\ 2) \\ (1\ 2\ 3) &= (1\ 2)(2\ 3) \\ (1\ 3\ 2) &= (1\ 3)(3\ 2) = (1\ 2)(2\ 3)(1\ 2)(2\ 3)\end{aligned}$$

2.2.2 Subgroup Structure of $\mathbb{S}_n$

All symmetric groups of order less than n, denoted by $\mathbb{S}_{n-k}$ where $\{1 \leq k \leq n-1\}$, are the subgroups of $\mathbb{S}_n$. For example

$$\mathbb{S}_3 \supset \mathbb{S}_2 \supset \mathbb{S}_1 = \mathtt{id}\,. \tag{2.11}$$

Here `id` is the symmetric group which contain only 1 element, the permutation identity. In terms of cycle notation, we can rewrite (2.11) as

$$\{e, (1\ 2), (2\ 3), (1\ 3), (1\ 2\ 3), (1\ 3\ 2)\} \supset \{e, (1\ 2)\} \supset e = \mathtt{id}\,.$$

The subgroup $\mathbb{S}_{n-k}$ of $\mathbb{S}_n$ includes permutations $\pi \in \mathbb{S}_n$ that permutes $\{1, 2, \ldots, (n-k)\}$ but leaves $\{(n-k+1), \ldots, n\}$ fixed. Consider the following example

$$\begin{aligned}
(1\ 2) &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 3 & \mathbf{4} & \mathbf{5} \end{pmatrix} \\
(2\ 3) &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 2 & \mathbf{4} & \mathbf{5} \end{pmatrix} \\
(1\ 3) &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 1 & \mathbf{4} & \mathbf{5} \end{pmatrix} \\
(1\ 2\ 3) &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & \mathbf{4} & \mathbf{5} \end{pmatrix} \\
(1\ 3\ 2) &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & \mathbf{4} & \mathbf{5} \end{pmatrix}
\end{aligned}$$

In this example, all permutations leave $\mathbf{4}, \mathbf{5}$ fixed. One can see that all of these permutations constitute $\mathbb{S}_3$ which is a subgroup of $\mathbb{S}_5$.

The set $\sigma\mathbb{S}_{n-k}$ is the left coset of $\mathbb{S}_{n-k}$ in $\mathbb{S}_n$. Similarly, $\mathbb{S}_{n-k}\sigma$ is the right

coset of $\mathbb{S}_{n-k}$ in $\mathbb{S}_n$.

$$\sigma\mathbb{S}_{n-k} = \{\sigma\pi : \pi \in \mathbb{S}_{n-k}\} \text{ is a left coset of } \mathbb{S}_{n-k} \text{ in } \mathbb{S}_n$$
$$\mathbb{S}_{n-k}\sigma = \{\pi\sigma : \pi \in \mathbb{S}_{n-k}\} \text{ is a right coset of } \mathbb{S}_{n-k} \text{ in } \mathbb{S}_n$$

The left and the right cosets for the symmetric group $\mathbb{S}_3$ and the subgroup $\mathbb{S}_2$ are:

Left cosets			Right cosets		
$e\,\mathbb{S}_2$	$=$	$\{e, (1\,2)\}$	$\mathbb{S}_2\,e$	$=$	$\{e, (1\,2)\}$
$(1\,3)\,\mathbb{S}_2$	$=$	$\{(1\,3), (1\,2\,3)\}$	$(1\,3)\,\mathbb{S}_2$	$=$	$\{(1\,3), (1\,3\,2)\}$
$(2\,3)\,\mathbb{S}_2$	$=$	$\{(2\,3), (1\,3\,2)\}$	$(2\,3)\,\mathbb{S}_2$	$=$	$\{(2\,3), (1\,2\,3)\}$.

Important properties of cosets are:

1. The order of coset is determined by the order of the subgroup $\mathbb{S}_{n-k}$ (Lagrange's theorem). For example, the coset $(1\,3\,2)\,\mathbb{S}_2 = \{(1\,3\,2), (2\,3)\}$, has order 2

2. $\mathbb{S}_n$ is the union of the left (and right) cosets of $\mathbb{S}_{n-k}$ in $\mathbb{S}_n$

3. Any two left (right) cosets are either disjoint or the same, i.e., coset of $\mathbb{S}_{n-k}$ partition $\mathbb{S}_n$. The left and the right cosets give two different partitions of $\mathbb{S}_n$

4. For $\sigma_1, \sigma_2 \in \mathbb{S}_n$, we have $\sigma_1\,\mathbb{S}_{n-k} = \sigma_2\,\mathbb{S}_{n-k}$ if and only if $\sigma_1^{-1}\sigma_2 \in \mathbb{S}_{n-k}$. Similarly, $\mathbb{S}_{n-k}\,\sigma_1 = \mathbb{S}_{n-k}\,\sigma_2$ if and only if $\sigma_1\sigma_2^{-1} \in \mathbb{S}_{n-k}$. For example, $(1\,3\,2)\,\mathbb{S}_2 = (2\,3)\,\mathbb{S}_2$ because $(1\,3\,2)^{-1}(2\,3) = (1\,2) \in \mathbb{S}_2$

5. The *index* of $\mathbb{S}_{n-k}$ in $\mathbb{S}_n$ is the total number of distinct left (right) cosets of $\mathbb{S}_{n-k}$ in $\mathbb{S}_n$, given by $|\mathbb{S}_n|/|\mathbb{S}_{n-k}|$. For example, the index of $\mathbb{S}_2$ in $\mathbb{S}_3 = |3!|/|2!| = 3$

2.2.3 Representations of $\mathbb{S}_n$

The representation theory of $\mathbb{S}_n$ is a particular case of the representation theory of finite groups, which is a well-studied topic. A concrete and detailed theory can be found here (James, 1987). Here we present two representations of $\mathbb{S}_n$ which is relevant for the content in this dissertation. We first discuss the $n-$dimensional permutation matrix representation of $\mathbb{S}_n$. Later, we describe the irreducible representation of $\mathbb{S}_n$.

The Permutation Representation

Also known as the *defining representation* of $\mathbb{S}_n$, the permutation matrix representation is an n-dimensional representation which maps a permutation $\sigma \in \mathbb{S}_n$ to a matrix $P_\sigma \in \mathbb{R}^{n\times n}$ such that [4]

$$[P(\sigma)]_{q,p} := \begin{cases} 1 & \text{if } \sigma(p) = q \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

There are $n!$ permutation matrices of size $n \times n$ on $\mathbb{S}_n$ which means this is a *faithful* representation. A representation ρ of a group G is faithful if different elements $g \in G$ are represented by distinct linear mappings $\rho(g)$. For instance, the permutation matrix representation corresponding to $\mathbb{S}_3$ is a collection of following $3! = 6$ matrices

[4] In general, ρ denotes representation of a group G. In case of $\mathbb{S}_n$, we use $P(\sigma)$ for permutation matrix representation and ρ for irreducible representation.

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (12) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (23) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$(13) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (123) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (132) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} .$$

Besides being intuitive by design, the permutation matrix representation is a unitary representation, i.e., $P^{-1} = P^{\top}$, and preserves the group structure, i.e., $P(\sigma_1\sigma_2) = P(\sigma_1)P(\sigma_2)$. For example

$$P_{(12)} \circ P_{(23)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.13}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.14}$$

$$= P_{(123)} \, . \tag{2.15}$$

A unitary basis transformation can simultaneously block diagonalize $P_{(\sigma)}$ into a direct sum of two lower dimensional representations: *1*) a trivial representation of $1-$ dimension; 2) standard $(n-1)$ dimensional irreducible representation. This means, the permutation matrix representation is a *reducible* representation. One such unitary transformation for $\mathbb{R}^{n\times n}$ can be obtained by constructing an orthonormal basis such that $\mathbb{1}$ is the first

basis vector. For example,

$$T = \begin{bmatrix} 0.5774 & 0.5774 & 0.5774 \\ -0.5774 & 0.7887 & -0.2113 \\ -0.5774 & -0.2113 & 0.7887 \end{bmatrix}$$

is an orthonormal basis in $\mathbb{R}^{3\times 3}$ such that

$$T^{-1}P_e\,T = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \qquad T^{-1}P_{(12)}\,T = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & -0.866 & -0.5 \\ 0 & -0.5 & 0.866 \end{array}\right]$$

$$T^{-1}P_{(23)}\,T = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right] \qquad T^{-1}P_{(13)}\,T = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 0.866 & -0.5 \\ 0 & -0.5 & -0.866 \end{array}\right]$$

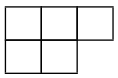
$$T^{-1}P_{(123)}\,T = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & -0.5 & -0.866 \\ 0 & 0.866 & -0.5 \end{array}\right] \quad T^{-1}P_{(132)}\,T = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & -0.5 & 0.866 \\ 0 & -0.866 & -0.5 \end{array}\right].$$

The Irreducible Representation

There are several ways to construct irreducible representations of $\mathbb{S}_n$ (Sagan, 2001). One such representation is called the Young's orthogonal representation (YOR). The YOR matrices are real and unitary, i.e., YOR matrices are orthogonal, making them computationally attractive.

YOR is realized over conjugacy classes, namely partitions of n. A partition λ of n (denoted $\lambda \vdash n$) is a k-tuple $\lambda = (\lambda_1, ..., \lambda_k)$ of weakly decreasing positive integers $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i = n$. A graphical sketch of a partition λ is a left-justified arrangement of empty boxes with λ_i boxes in i-th row. This is called a *Young diagram*, denoted as F^λ. For example, the Young diagram corresponding to partition $\lambda = (3, 2)$

is



A Young diagram bijectively filled with the integers $\{1, 2, \cdots, n\}$ produces a combinatorial object called a *Young tableau*. A tableau t is called *standard* if the assignment of numbers increases from left to right in each row and top to bottom in each column. The following are all the standard Young tableaux of shape $\lambda = (3, 2)$:

$$\begin{array}{|c|c|c|}\hline 1&3&5\\\hline 2&4\\\cline{1-2}\end{array} \quad \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 3&4\\\cline{1-2}\end{array} \quad \begin{array}{|c|c|c|}\hline 1&3&4\\\hline 2&5\\\cline{1-2}\end{array} \quad \begin{array}{|c|c|c|}\hline 1&2&4\\\hline 3&5\\\cline{1-2}\end{array} \quad \begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array}$$

Elements of $\mathbb{S}_n$ act on a standard tableau in the obvious manner by permuting the numerals. The set of standard tableau for λ forms the basis for corresponding irreducible representation ρ_λ as vector space. In particular, the YOR corresponding to shape λ is a matrix of size $d_{\rho_\lambda} \times d_{\rho_\lambda}$, where d_{ρ_λ} denotes the number of unique standard $\lambda-$tableaux. The dimensionality of $\rho_\lambda = d_{\rho_\lambda}$ is determined by the *hook-length formula*, which is

$$d_{\rho_\lambda} = \frac{n!}{\Pi_{r\in\lambda} h_\lambda(r)} \tag{2.16}$$

Here, $h_\lambda(r)$ is called the *hook-length* for square r in the Young diagram of shape λ. $h_\lambda(r)$ is the total number of all squares directly to right of r and directly below r (including r itself). For example, consider the partition $\lambda = (2, 2, 1)$. The hook-lengths for each square r are shown in the respective squares,

From this example, we get

$$d_{\rho_{(2,2,1)}} = \frac{5!}{4 \cdot 2 \cdot 3 \cdot 1 \cdot 1} = 5. \tag{2.17}$$

In practice, the representation matrix corresponding to a given permutation $\sigma \in \mathbb{S}_n$ is constructed by first writing σ as a product of adjacent transpositions $\tau_1 \tau_2 \cdots \tau_{n-1}$, and then by composing the adjacent transposition YOR matrices together. Individual Young's orthogonal matrix corresponding to τ_i is calculated as follows:

$$\rho_\lambda(\tau_i)_{t,t'} = \begin{cases} (d_t(i, i+1))^{-1} & \text{if } t = t' \\ \sqrt{1 - (d_t(i, i+1))^{-2}} & \text{if } t' = \tau_i(t) \\ 0 & \text{otherwise,} \end{cases}$$

where d_t is the number of steps it takes to move i to $i + 1$[5]. Thus, $\rho_\lambda(\tau_i)$ is very sparse. Since adjacent transpositions generate the whole $\mathbb{S}_n$, the set of ρ_λ calculated for adjacent transpositions are sufficient to generate the complete set of irreducible representations for $\mathbb{S}_n$.

Branching rule and adapted representation. *Branching rule* describes the connection between irreps of $\mathbb{S}_n$ and its subgroups which can be explained by understanding the relation between the Young diagrams of n and $n - k$ where $k = 1, \cdots, n - 1$. A closer look at the Young diagram

[5]North and east movements are taken as positive, and south and west movements are taken negative.

for a partition $\mu \vdash n-1$ suggests that one can add a square to the current Young diagram and generate one or more Young diagrams for partitions of n, denoted by $\lambda \uparrow^n$. For example, Figure 2.1 illustrates the complete branching structure for $n = 4$. This is a directed graph, known as the *Bratteli diagram*, where partitions are nodes and edges describe *which μ is contained in which λ?*

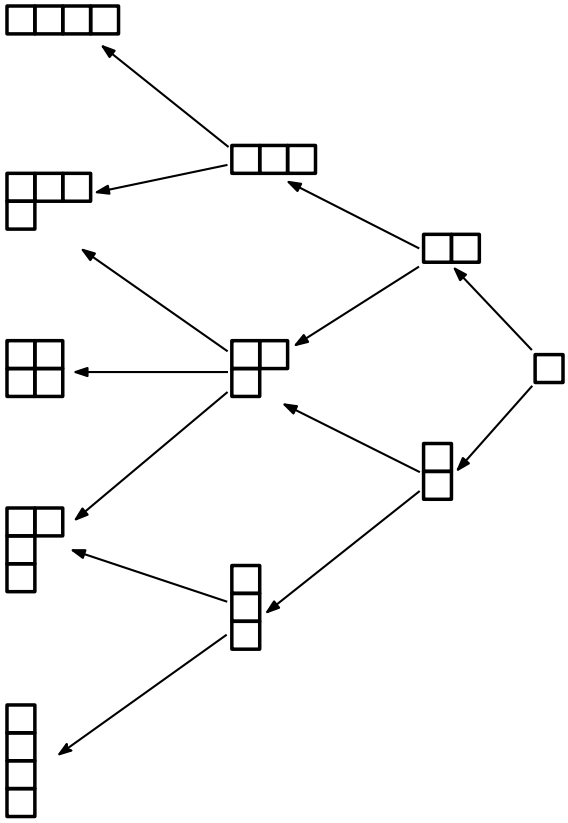


Figure 2.1: The Bratelli diagram for $\mathbb{S}_4$.

The extended Brattelli diagram further explains the connection between the standard tableaux and irreducible representation of $\mathbb{S}_n$ and $\mathbb{S}_{n-k}$. Given all standard tableaux of shape $\lambda \vdash n$, a collection of standard tableaux for $\mu \vdash n-1$ can be generated by removing the number "n" , Figure 2.2. This connection reveals how the representation of $\mathbb{S}_n$ for the shape λ is related to the irreducible representation of $\mathbb{S}_{n-k}$.

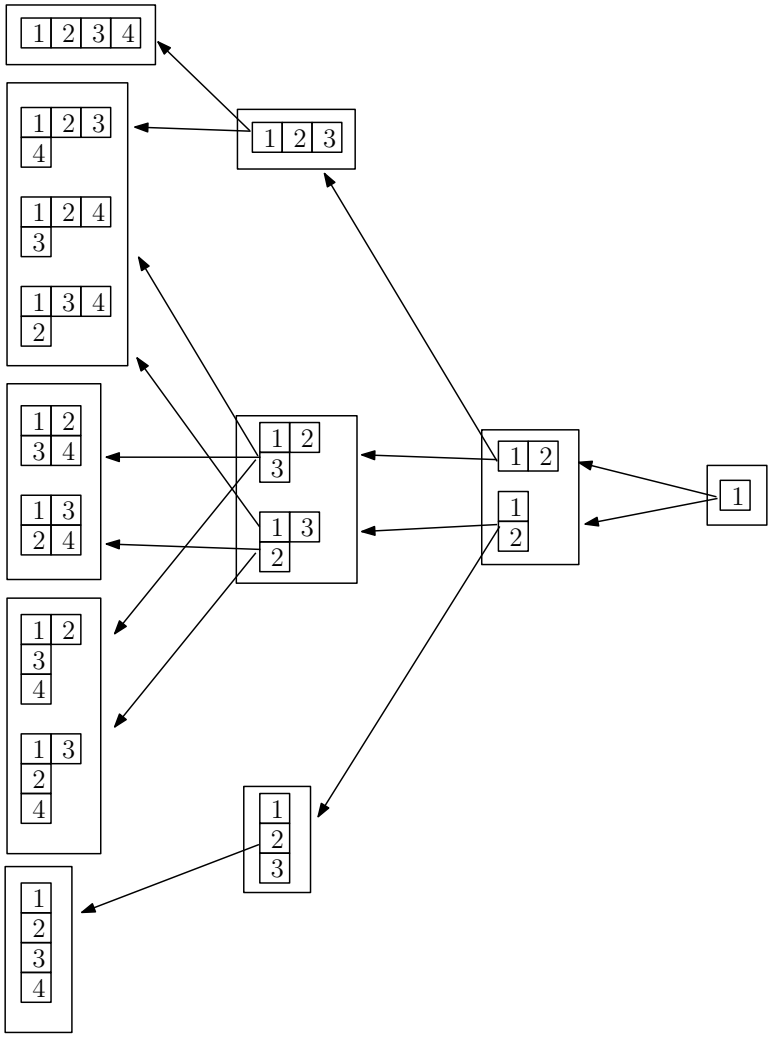


Figure 2.2: The extended Bratelli diagram for $\mathbb{S}_4$.

Consider $\lambda = (3,1) \vdash (n = 4)$, the irreducible representation for this shape restricted to $\mathbb{S}_3$, denoted by $\rho_{(3,1)} \downarrow_{\mathbb{S}_3}$, can be written as

$$\rho_{(3,1)\downarrow_{\mathbb{S}_3}} = \rho_{(3)} \oplus \rho_{(2,1)} \ . \tag{2.18}$$

Recall that the basis of YOR is found through standard tableau. In the light of the branching rule, each irreducible representation ρ_λ of $\mathbb{S}_n$ splits when restricted to $\mathbb{S}_{n-1}$. More specifically

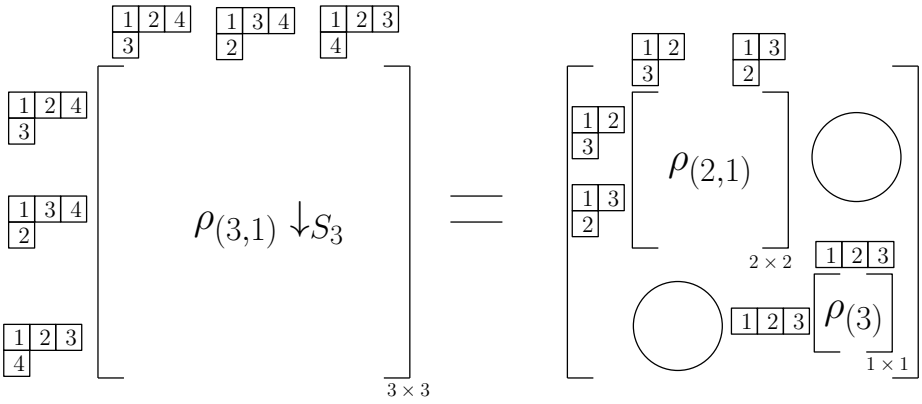


Figure 2.3: Irreducible representation $\rho_{(3,1)\downarrow_{\mathbb{S}_3}}$ restricted to $\mathbb{S}_3$.

This implies that the YOR is *adapted*, and the resulting representation is called an *adapted* representation. A similar construction involving other smaller subgroups of $\mathbb{S}_n$ generalizes the notion of an adapted representation.

Formally, a complete set of YOR representations of $\mathbb{S}_n$ is adapted if there is a set of inequivalent YOR representations of subgroup $\mathbb{S}_{n-k}$. This property is important to derive efficient algorithms such as the fast Fourier transform of functions defined on $\mathbb{S}_n$.

2.3 Harmonic Analysis on $\mathbb{S}_n$

Harmonic analysis of a function $f : \mathbb{S}_n \to \mathbb{C}$ is defined via the notion of representations

$$\hat{f}(\rho) = \sum_{\sigma \in \mathbb{S}_n} f(\sigma)\rho(\sigma) \qquad \rho \in \mathcal{R}. \tag{2.19}$$

Here $\mathcal{R}$ denotes the complete set of inequivalent irreducible representations. Harmonic analysis of f means interpreting the coefficients of its "Fourier expansion". In particular, the scaled matrix entry $[\widehat{f}(\rho)]_{ij}$ is the *coefficient* of the function $(\sigma) \longrightarrow [\rho(\sigma)]_{ij}$ in the expansion of f, (Maslen, 1998). Much of the practical interest in Fourier transform, (Diaconis, 1989; Rudin, 1962), is due to the underlying combinatorial connection between the irreducible representation of $\mathbb{S}_n$ and the irreducible representation of smaller groups $\mathbb{S}_{n-k}$. Sparsity and unitarity of the irreducible representation further makes the harmonic analysis of functions defined on $\mathbb{S}_n$ extremely useful.

2.3.1 Fast Fourier Transform

Clausen was the first to propose an algorithm for Fast Fourier transform on non-commutative group $\mathbb{S}_n$. There are two key ideas which make the algorithm (Clausen, 1989) practically useful. Firstly, the proposed algorithm operates on the nested chain of subgroups

$$\mathbb{S}_n \supset \mathbb{S}_{n-1} \supset \cdots \supset \mathbb{S}_1 = \mathtt{id}$$

where $\mathbb{S}_k$ is a subgroup that permutes the first k objects among themselves and leaves the rest of the $n-k$ objects fixed. Next, it uses the YOR adapted representation which systematically relates the irreducible representation of $\mathbb{S}_n$ to the irreducible representation of the smaller group $\mathbb{S}_k$ as described in the previous section.

The algorithm systematically factors the elements of $\mathbb{S}_n$ into a product

of *contiguous cycles*. A contiguous cycle $[\![i,j]\!] \in \mathbb{S}_n$ is a permutation of the form

$$[\![i,j]\!](k) = \begin{cases} k+1 & \text{for } k = i, i+1, .., j-1, \\ i & \text{for } k = j, \\ k & \text{otherwise} \end{cases} \qquad 1 \leq i \leq j \leq n. \tag{2.20}$$

For example, the contiguous cycle $[\![1,3]\!] \in \mathbb{S}_4$ is a permutation which permutes $\{1,2,3,4\}$ as

$$\begin{aligned} [\![1,3]\!]\,(1) &= 2 \\ [\![1,3]\!]\,(2) &= 3 \\ [\![1,3]\!]\,(3) &= 1 \\ [\![1,4]\!]\,(4) &= 4 \end{aligned}$$

Two important properties of $[\![i,j]\!]$ are:

1. Each $[\![i,j]\!]$ factors into $j-i-1$ adjacent transpositions;
2. There is a unique factorization of any $\sigma \in \mathbb{S}_n$ into a product of contiguous cycles which is adapted to the chain of subgroups

$$\mathbb{S}_n \supset \mathbb{S}_{n-1} \supset \cdots \supset \mathbb{S}_1 = \mathtt{id}\,.$$

Each subgroup creates a coset partition of $\mathbb{S}_n$. Clausen's FFT algorithm uses this nested chain of subgroups and proceeds by recursively breaking down Fourier transformation into smaller independent transforms. In particular, for a function $f : \mathbb{S}_n \to \mathbb{C}$, we define $f_i(\sigma') = f([\![i,n]\!]\sigma')$ for $\sigma' \in \mathbb{S}_{n-1}$ and $i = 1, 2, \cdots, n$ as follows

$$\hat{f}_i(\rho_\mu) = \sum_{\sigma' \in \mathbb{S}_{n-1}} \rho_\mu(\sigma') f_i(\sigma') \qquad \mu \vdash n-1. \tag{2.21}$$

Finally, the full Fourier transform $\widehat{f}$ is assembled from $\widehat{f}_1, \widehat{f}_2, \ldots, \widehat{f}_n$

$$\hat{f}(\rho_\lambda) = \sum_{i=1}^{n} \rho_\lambda([\![i, n]\!]) \bigoplus_{\mu \in \lambda \downarrow_{n-1}} \hat{f}_i(\rho_\mu) \qquad \lambda \vdash n. \tag{2.22}$$

Here $\lambda \downarrow_{n-1}$ denotes the set of irreps of $\mathbb{S}_{n-1}$ featured in the restriction of ρ_λ to $\mathbb{S}_{n-1}$.

2.4 Summary

In the previous chapter, Chapter 1, we showed that many matching problems are formulated as combinatorial optimization problems. Though the specific form of the objective function may depend on the specific application of interest. We further argued that a clear structural understanding of the solution space may provide useful insights in developing efficient solution strategies for these problems. In this chapter, we provided a richer description of $\mathbb{S}_n$; definitions and properties, notations and representations and various algebraic properties of the functions defined on $\mathbb{S}_n$. In the rest of this thesis, we demonstrate how these concepts can be applied to matching problems arising in computer vision. Specifically, in Chapter 3, we study the problem of learning how to match graphs efficiently: given a nominal amount of training images, whether we can learn good composites of base features for matching pair of images which yield the user desired solutions efficiently. We reformulate this problem and use following insights

1. QAP objective is a function on $\mathbb{S}_n$.
2. The Fourier transform of QAP function on $\mathbb{S}_n$ is sparse and highly bandlimited.
3. The underlying coset structure of $\mathbb{S}_n$ motivates a risk minimization

strategy for learning weights over features that makes a search process faster.

4. The representation theory of $\mathbb{S}_n$ makes the proposed procedure computationally tractable.

In Chapter 4, we study the multi-way matching problem: matching not just two, but m different sets of objects to each other. This problem arises in many contexts, including finding the correspondences between feature points across multiple images in computer vision. We use

1. The group structure of $\mathbb{S}_n$ to *enforce* an important consistency condition on the putative matchings, i.e., the *transitivity* condition.

2. An appropriate distance metric on $\mathbb{S}_n$ to write the loss function for the combinatorial optimization problem at hand.

3. The representation theory of $\mathbb{S}_n$ which makes the proposed formulation computationally viable.

In Chapter 5, we further study the multi-way matching problem in the presence of structured noise which naturally originates in realistic scenarios such as structural ambiguity and occlusion. We use

1. The group structure of $\mathbb{S}_n$ to *identify* the consistent matching instances.

2. The representation theory of $\mathbb{S}_n$ to precisely encode the observed matchings as the distributions on $\mathbb{S}_n$.

3. The correlation theorem to encode the invariance structure of the induced distributions on $\mathbb{S}_n$.

3 INCORPORATING DOMAIN KNOWLEDGE IN MATCHING PROBLEMS VIA HARMONIC ANALYSIS

In this chapter, we study the problem of learning how to match domain specific graphs efficiently when a nominal amount of training images are available. The main idea is that by restricting the focus to a specific domain and tuning weights that best reflect practical considerations in that application, we may be able to learn good composites of base features for matching pair of images which yield the user desired solutions efficiently.

Consider the set of images in Figure 3.1 taken from different viewpoints. The matching seeks to assign feature points in one image to its most similar feature point in the other image. While matching pairs of images, not only

Figure 3.1: A set of images of the same object (Hartley and Zisserman, 2000).

do we want landmarks in one image to be matched to similar landmarks

in the other image, but we also want the distances between landmarks in the first image to be similar to the distances between the corresponding landmarks in the second one. This leads to a more general optimization problem

$$\arg\max_{\sigma\in\mathbb{S}_n} \sum_{i,j=1}^{n} A_{\sigma(i),\sigma(j)}\, A'_{i,j}, \tag{3.1}$$

known as the **quadratic assignment problem** (QAP).

Unfortunately, the QAP is NP–hard, and it is also notoriously hard to approximate or solve heuristically. Combinatorial search methods such as branch and bound almost never manage to solve real world QAP instances effciently, while convex optimization methods suffer from the fact that the feasible region (called the second order permutation polytope) has an exponential number of faces.

It is natural to ask whether one can use side information to obtain solutions to typical QAPs (seen in a specific application) easier. Here, we propose a new approach for doing this by *learning* a modified objective function f^{ω} from a set of prior "training" QAP instances. The two criteria that f^{ω} must satisfy are

1. $\arg\max_{\sigma\in\mathbb{S}_n} f^{\omega}(\sigma)$ should be close to the maximum of the original objective function f.
2. f^{ω} should be much easier to optimize than f.

The vector $\boldsymbol{\omega}$ that parametrizes f^{ω} is determined by methods similar to Structural Risk Minimization. This form of risk minimization strategy (Finley and Joachims, 2008), as in Structural SVMs, when applied to the training set, allows our parameter $\boldsymbol{\omega}$ to generalize to unseen examples well. Note that Structural SVMs are very well understood if the original inference problem ($\arg\max f^{\omega}$) is polynomial time solvable (see (Tsochantaridis et al., 2006; Taskar et al., 2003)). Unfortunately, the original graph matching problem requires finding a $\boldsymbol{\omega}$ such that the minimizer of $f^{\omega}(\cdot)$ matches σ^* — this is itself a QAP. Existing theoretical guarantees for Structural SVM are

known to be far less satisfactory for such intractable problems (Tsochantaridis et al., 2006; Joachims et al., 2009). Fortunately, we may observe that the σ's of our interest are not arbitrary objects, rather constitute the symmetric group, and this opens the door to look at the properties of specific sub-classes of f^ω and/or σ that may provide useful insights into efficient solution strategies, and are better informed by the the characteristics of the underlying vision or learning task.

In this chapter, we restrict our attention to the problem of learning parameters for graph matching. The learning graph matching problem (Caetano et al., 2009) seeks to solve for parameters of compatibility functions so that the solution from an approximate method matches the permutation provided by the user as best as possible. Also, ideally we want that the cost of obtaining the approximate solution is much less than solving the original matching problem optimally. Since the main objects of interest are permutations, it makes this an ideal sandbox to develop and present our main ideas. Note, the model developed here is fully general and equally applicable to learning version of matching problems in domains other than computer vision.

Graph Matching and other Related Work. Graph Matching is the problem of finding correspondences between the nodes of two graphs to maximize alignment. A common and general way to model it is to write it as a Quadratic Assignment Problem (QAP), which is among the most well-studied combinatorial optimization problems in the literature (Pardalos et al., 1994; Cela, 1998; Umeyama, 1988; Vishwanathan et al., 2007). Many alternative approaches for graph matching are also known (Leordeanu and Hebert, 2005; Hancock and Wilson, 2009; Caelli and Caetano, 2005). In the context of *learning* graph matching (Caetano et al., 2009; Leordeanu and Hebert, 2009), one is interested in the following question: if the optimal correspondence between the nodes of a pair of graphs is known (and many such pairs are available), how should one use this knowledge to

learn correspondences (rather, cheaply find the correspondences) between another pair of graphs which were extracted under similar *conditions*. The notion of *conditions* reflects properties of the application under study – for instance, (Caetano et al., 2009) uses the example of image pairs acquired under similar illumination from an airport surveillance camera, where the matching task refers to aligning the "feature points" from such images. The to be determined parameter $\boldsymbol{\omega}$ then corresponds to weights which appropriately adjust the joint feature map of node and edge compatibilities, so that the match found by the solver agrees with the user provided solution. Learning graph matching serves another very important need – by restricting our focus to a *specific* domain and tuning weights that best reflect practical considerations in that application, a less sophisticated approach may still be able to obtain good quality solutions in a fixed amount of time or memory (Xu et al., 2007). This has implications in most situations where training data is available. The algorithm in (Caetano et al., 2009) uses a nice structure learning formalization for this problem. But finding the most violated constraint precisely in the construction (Caetano et al., 2009) itself involves solving a QAP, and can only be approximately estimated (e.g., via its linear assignment relaxation).

3.1 The form of QAP in Fourier space

The key intuition behind our approach is the notion of harmonic analysis associated with functions defined on $\mathbb{S}_n$. The Fourier transform of a general function $f : \mathbb{S}_n \longrightarrow \mathbb{R}$ consists of the matrices

$$\widehat{f}(\lambda) = \sum_{\sigma \in \mathbb{S}_n} f(\sigma)\, \rho_\lambda(\sigma), \tag{3.2}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ is the so-called integer partition of n and plays the role of "frequency", while $\rho_\lambda : \mathbb{S}_n \longrightarrow \mathbb{C}^{d_\lambda \times d_\lambda}$ is the corresponding

irreducible representation (irrep) of $\mathbb{S}_n$ as described in Chapter 2. The algorithm for fast Fourier transform generally works by first computing Fourier transforms of f restricted to small cosets, and then recursively assembling such small transforms into ever large ones until we reach the Fourier transform in Equation (3.2) on the entire group. In (Kondor, 2010), the author proposed a QAP solver which used this structure, Section 3.1, to search $\mathbb{S}_n$ by employing the Inverse Fast Fourier Transform (iFFT) to restrict f to various cosets and bound it.

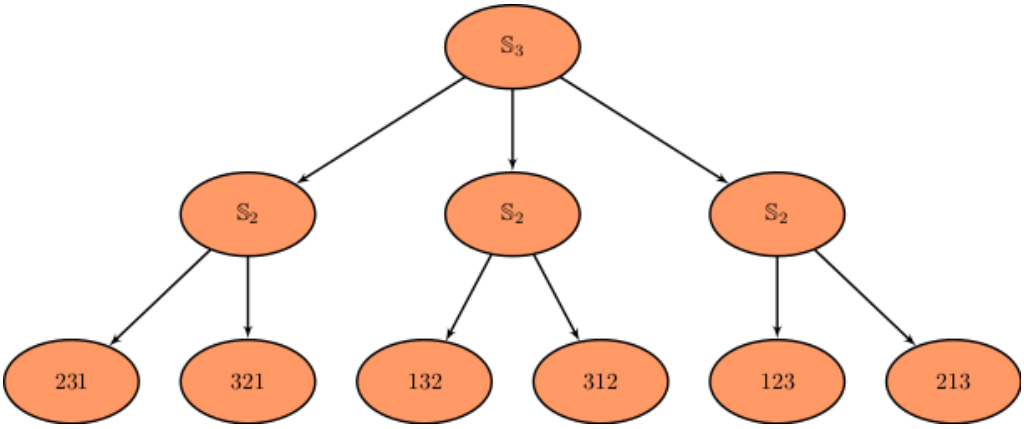


Figure 3.2: Left coset tree for $\mathbb{S}_3$ showing all members of $\mathbb{S}_3$ as leaves. Second level of the tree represents $\mathbb{S}_2$–coset of $\mathbb{S}_3$ which corresponds to candidate permutations $[\![1,3]\!], [\![2,3]\!], [\![3,3]\!]$.

Given a graph G of n vertices with adjacency matrices A, (Kondor, 2010) defined a function called graph function on G $f_A \colon \mathbb{S}_n \longrightarrow \mathbb{R}$,

$$f_A(\sigma) = A_{\sigma(n),\sigma(n-1)}. \tag{3.3}$$

Further, it expressed the standard *quadratic assignment* objective

$$\max_{\sigma \in \mathbb{S}_n} \; f(\sigma) = \sum_{i,j=1}^{n} A_{i,j} A'_{\sigma(i),\sigma(j)}, \tag{3.4}$$

as the **graph correlation** between the two graphs G and G'

$$f(\sigma) = \frac{1}{(n-2)!} \sum_{\pi \in \mathbb{S}_n} f_A(\sigma\pi) f_{A'}(\pi) \tag{3.5}$$

where f_A and $f_{A'}$ are defined in Equation (3.3). In Fourier space it takes the following form

$$\widehat{f}(\lambda) = \frac{1}{(n-2)!} \widehat{f}_A(\lambda) \widehat{f}_{A'}(\lambda)^\dagger,$$

where † stands for the conjugate transpose. Given that functions of the form Equation (3.3) are band-limited to $\{(n), (n-1,1), (n-2,2), (n-2,1,1)\}$ (i.e., all components of $\widehat{f}_A$ other than those indexed by these integer partitions are identically zero), the Fourier transform of the objective function will also be restricted to $\widehat{f}((n))$, $\widehat{f}((n-1,1))$, $\widehat{f}((n-2,2))$ and $\widehat{f}((n-2,1,1))$. The final component of Fourier space QAP solver are the bounds for f (restricted to various cosets) based on its Fourier components. The simplest such bound is

$$f(\sigma) \leq \frac{1}{n!} \sum_{\lambda} d_\lambda \left\| \widehat{f}(\lambda) \right\|_* \tag{3.6}$$

where $\| \ \|_*$ denotes the nuclear norm. The bandlimited property of QAP in Fourier space made the computation of these Fourier bounds economical. The final algorithm in (Kondor, 2010) is a branch and bound type optimization algorithm that runs in $O(n^3)$ time per branch visited and is competitive with more conventional exact QAP solvers. Unfortunately, in many practical problems the resulting algorithm still takes an exponential amount of time to run, simply because it needs to visit so many branches.

Our proposed approach uses the Fourier space QAP objective and the machinery of non-commutative Fourier transforms to develop a supervised learning framework for graph matching problem.

3.2 The Objective function for Learning

Observe that QAPs are hard because the objective function is relatively flat, and according to most reasonable metrics, the diameter of $\mathbb{S}_n$ is small compared to its size. In practical problems, however, one can sometimes overcome such barriers by making inventive use of side information. In particular, we take the approach of using similar QAP instances (derived from the application of interest) to learn a modified objective function that will more effectively drive our algorithm to the correct solution. Since solving a QAP from scratch for each modification of parameters is clearly an intractable option, our main goal will be to adapt tools from non-commutative harmonic analysis to recast the problem in a way that sidesteps this burden. The framework described next formalizes this idea.

3.2.1 Parameterizing the QAP objective

QAP instances derived from Vision problems typically rely on some analytic form approximating the perceptual similarity between two feature points. For instance, how similar are a pair of shape features (Belongie et al., 2002), u and v, extracted from different images? If a node to node match which is perceptually correct turns out to be only marginally better than many other incorrect matches, this necessarily suggests that the features are not very discriminative. One practical consequence is that a Branch and Bound type search will need to work much harder (i.e., explore many sub-trees) to find the global solution.

Our core strategy is to incorporate domain information in the QAP objective. To do this, we use the simple idea of composing various QAPs to write a base QAP objective which has certain desirable properties. Since the matching in computer vision applications is based on the low-level feature descriptors, it seems logical to use (rather differently weigh) these features in the design of a function that can be used to inform the base QAP

objective. It is here that we will leverage standard learning algorithms to learn the shared structure by observing multiple instances drawn from the specific domain.

Using training data (examples from the application of interest) we will be able to *learn* the parameters such that learnt parameters will induce domain friendly QAP instances – biasing the search towards the more interesting matches first, while simultaneously suppressing the influence of misleading features.

3.2.2 Learning Graph Matching

There are various ways of extracting features in vision. The list of options may include *edge features such as a Delaunay triangulation*, Euclidean distance between interest points, Shape context features etc. Consider D such representations are available to us, where each encodes the corresponding graph neighborhood.

We use each of these representations to generate D adjacency matrices for each graph. Each entry $A^d_{i,j}$ is a squared distance between the feature values of vertex i and j of G according to representation $d \in D$. Similarly, we define A'^d for G'. We can write the QAP objective in Equation (3.5) as before using (A^d, A'^d)

$$f^d(\sigma) = \frac{1}{(n-2)!} \sum_{\pi \in \mathbb{S}_n} f_{A^d}(\sigma\pi) f_{A'^d}(\pi)$$

We want to find a match such that edge (i, j) in G should be assigned to an edge (i', j') in G' that is of a similar length (or weight) simultaneously in all encoded adjacency matrices. However, features may be noisy and not very discriminative, and can lead to wrong assignments or take longer for the optimization scheme to find the correct solution. We instead parameterize

each $f^d(\sigma)$ and write a parameterized QAP objective for learning

$$f^{\omega}(\sigma) = \sum_{d=1}^{D} f^d_{\omega_d}(\sigma) \tag{3.7}$$

where the subscript ω_d represents parameterization, that is, we use the parameter vector $\omega \in \mathbb{R}^D$ to modulate the QAP function $f^{\omega}(\sigma)$.

The learning algorithm then essentially amounts to adjusting the ω appearing in the *parameterized* QAP objective using the *true* assignments σ^* given for each training pair (G_m, G'_m). Our general goal is to find

$$\omega^* = \arg\min_{\omega} \sum_{m=1}^{M} L(\hat{\sigma}_m(\omega), \sigma^*_m) + \Omega(\omega) \tag{3.8}$$

for some loss function $L(\cdot, \cdot)$ and regularizer Ω, where $\hat{\sigma}_m(\omega)$ is the optimal permutation for example m. Note that $\hat{\sigma}_m(\omega)$ itself corresponds to solving a QAP objective given a QAP objective modulated by parameter ω. The goal is to "learn" ω by performing *stochastic descent* at *each level of a tree of cosets* using an appropriate loss function (in experiments, we used hinge loss). We observe that there may be many ω's that will yield the same loss; the regularization Ω used here seeks the minimal difference from the ω in the previous step, so that the path/node leading to σ^*_m is preferable to any other node at this level, by a small margin. The exact algorithm is discussed next.

3.3 Algorithm: Learning in Fourier space

In the following section, we explain our learning scheme. The Fourier space machinery discussed in Section 3.1 bounds the objective function for learning (introduced in Section 3.2).

3.3.1 Fourier Space Upper Bounds for Learning

When working in a so-called adapted system of representations, the Fourier matrices at level n of a general function f can be expressed in terms of the Fourier matrices at level $n-1$ as

$$\widehat{f}(\lambda) = \sum_{i=1}^{n} \rho_\lambda([\![i,n]\!]) \bigoplus_{\mu \in \lambda\downarrow n-1} \widehat{f}_i(\mu). \tag{3.9}$$

where $\bigoplus$ is the direct sum of Fourier matrices, and $\lambda\downarrow_{n-1}$ is used to denote the integer partitions of the "ancestor partitions".

The Fourier space QAP solver proceeds by searching the tree of cosets, which corresponds to assigning vertex $n, n-1, \ldots$ of G to some sequence of vertices of G'. At level $n-k$ in the coset tree, it decides which vertex of G' it should assign the vertex $n-k$ to by comparing the Fourier space bounds. (Kondor, 2010) used the inverse map of Equation (3.9) to define the Fourier space bounds for standard QAP, where the inverse map at level $n-1$ is given by

$$\hat{f}_i(\mu) = \sum_{\lambda \in \mu\uparrow^n} \frac{d_\lambda}{n d_\mu} [\rho_\lambda([\![i,n]\!])^\top \hat{f}(\lambda)]_\mu, \tag{3.10}$$

Fourier space bounds are defined as

$$B_n \longrightarrow i = \sum_{\mu \vdash n-1} \| \sum_{\lambda \in \mu\uparrow^{n-1}} \frac{d_\lambda}{n d_\mu} [\rho_\lambda([\![i,n]\!])^\dagger \hat{f}(\lambda)]_\mu \|_* \,. \tag{3.11}$$

We replace $\hat{f}$ by $\hat{f}^\omega$ in (3.11) for estimating the parameters. It turns out that for a fixed ω, $f^\omega(\sigma)$ is easily computable from each $f^d_{\omega_d}$ *entirely in Fourier space*, without having to perform a very costly full Fourier transform. The exact form in which ω interacts with $f^d_{\omega_d}$ depends on the problem formulation. To keep the presentation simple, we formulate our QAP

objective as follows

$$f^{\omega}(\sigma) = \sum_{d=1}^{D} \omega_d f^d(\sigma) \tag{3.12}$$

The Fourier transform of f^{ω} can be expressed as

$$\widehat{f}^{\omega}(\lambda) = \sum_{d=1}^{D} \omega_d \widehat{f}^d(\lambda), \qquad \lambda \vdash n \tag{3.13}$$

The fact that makes f^{ω} *learnable* is that the inverse map $\widehat{f}_i^{\omega}(\mu)$ can be expressed as: $\widehat{f}_i^{\omega}(\mu) = \sum_{d=1}^{D} \omega_d \widehat{f}_i^d(\mu)$, where $\widehat{f}_i^d(\mu)$ is the inverse map of $\widehat{f}^d(\lambda)$ according to (3.10). The identity follows directly from linearity of Fourier transform.

Ultimately, we may use the convex nature of the nuclear norm that makes *Jensen's inequality* handy to derive an easy-to-optimize set of bounds,

$$\|\widehat{f}_i^{\omega}(\mu)\|_* = \|\sum_{d=1}^{D} \omega_d \widehat{f}_i^d(\mu)\|_* \leq \sum_{d=1}^{D} \omega_d \|\widehat{f}_i^d(\mu)\|_*. \tag{3.14}$$

With these concepts in hand, the only missing ingredient is the actual procedure to calculate the parameters ω that is presented next.

3.3.2 Stochastic gradient descent solver

In general, our loss function L as in (3.8) is a hinge loss on the relative bounds between the correct nodes and their incorrect siblings. This takes the form

$$\sum_{k=1}^{n} \sum_{i \in \text{children}((n-k+1)^*)} \left[\widehat{f}_i^{\omega}(\mu) - \widehat{f}_{i_{n-k}^*}^{\omega}(\mu) + 1\right]^+ \tag{3.15}$$

which is summed over all examples, and together with the regularization term represents the function we seek to minimize. Also, i_{n-k}^* denotes the correct node at level $n-k$ for some σ^*. Note that bounding the siblings of the correct nodes will bound f for *all* incorrect permutations.

We employ a stochastic gradient descent approach similar to (Shalev-Shwartz et al., 2007) on upper bounds defined in (3.11). During the training phase, we know which node to node assignment we should make, that is, we know which of these bounds, say the one corresponding to vertex i^*_{n-k}, we want to be the largest. A random training example and node i is selected and compared to its correct sibling i^*_{n-k}. We then reduce the objective for one term by taking a gradient step on ω. When $\Omega(w) = \frac{\nu}{2}\|\omega\|_2^2$ and $\widehat{f}_i^{\omega}$ is replaced with the bounds in Equation (3.14), each update takes the form

$$\omega_d \leftarrow \omega_d - \eta \begin{cases} \|\widehat{f}_i^d(\mu)\|_* - \|\widehat{f}^d_{i^*_{n-k}}(\mu)\|_* + \frac{\nu}{M\mathcal{O}(n^2)}\omega_d \\ \frac{\nu}{M\mathcal{O}(n^2)}\omega_d \end{cases} \tag{3.16}$$

η is an exponentially decaying step length parameter, M is the number of training examples, and the $\frac{\nu}{M\mathcal{O}(n^2)}$ term arises from splitting the regularizer over all nodes considered by the optimization.

This process is like structural SVM, where we try to find parameters so that the model predicts i^*_{n-k} instead of i, with the bound for each correct node in the coset trees of the training set greater than its incorrect siblings by some margin. As per (3.14), the learnt parameters are goodness measures of individual graph correlation function f^d contributing in (3.12).

3.4 Experiments

We considered the task of aligning pairwise images using local features extracted from the image data including interest points, and shape context features.

Edge–features: We performed Delaunay triangulation on interest points to generate unweighted edges. This provides unweighted adjacency ma-

trix.

Distance–features: We used 2D coordinates of interest points to calculate Euclidean distance between points. We extracted weighted adjacency matrices at various scales using 2D distances.

Shape Context–features: As in (Caetano et al., 2009), we also included *Shape context* features (Belongie et al., 2002). Briefly, the feature vector is a descriptor in Log-Polar space that describes the localized shape at each node. We generated weighted adjacency matrices by using normalized histogram differences of subsets of shape context features.

Dataset and Setup: We performed an experimental evaluation on 3 datasets and our experimental set up is similar to (Caetano et al., 2009). Graph pair instances were generated such that the two graphs are separated by a varying baseline (referred to as "offset" below). Our training data include multiple QAP's for each pair of images. The algorithm learnt a suitable ω. We performed standard 10–fold cross-validation on train/test data for each offset. The number of pairs in the train/test splits varied based on the offset. We summarize our results on various datasets below.

Hotel/House Dataset: We considered the CMU house dataset, which contains 111 frames of a video sequence of a toy house. Landmark points were identified and hand–labeled in each frame. Quantitative results corresponding to the matches found on the test set are shown in Figure 3.3. (Red plots) present the accuracy of matches (on test sets) as the offset (separation between frames) varies. We compared our results against a greedy ("No Learning") matching on two feature settings (blue plots). As expected, no-learn-greedy approach perform poorly if not all features are informative (blue dashed line). For small offsets, no-learn-greedy takes advantage of the fact that the problem instance is easy and the shape

context features are useful, but its performance gradually deteriorates as the offset increases (blue bold line). A greedy assignment using learnt weights still performs well. Qualitative results corresponding to the learnt matches on the test set shown in Figure 3.4. Note that the test phase does *not* perform any backtracking in Branch and Bound.

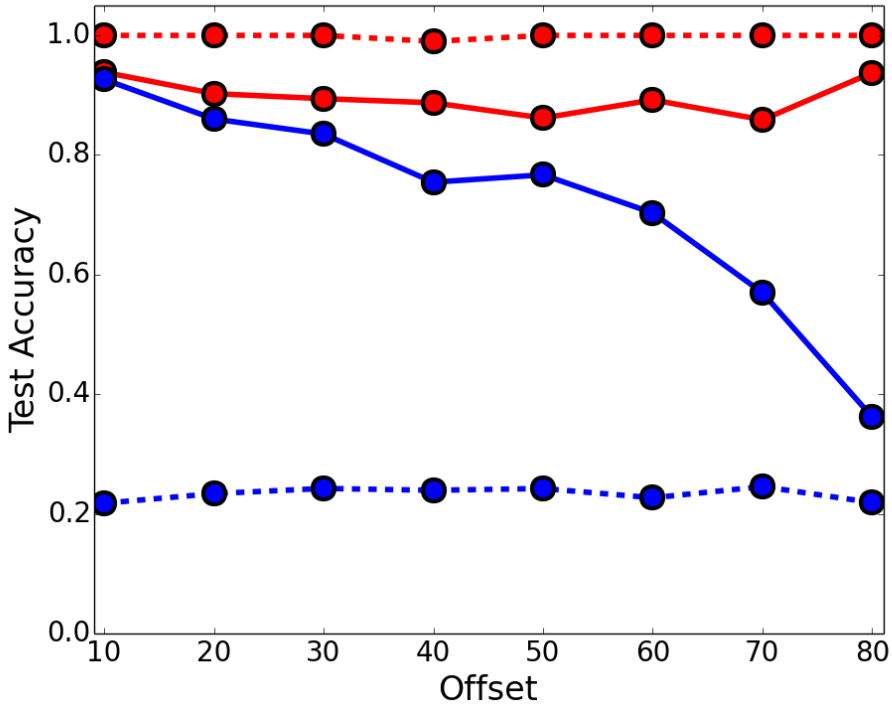


Figure 3.3: House dataset. Proposed learning method (red) compared with no-learn baseline (blue). (Dashed) Delaunay, distance and uninformative features. (Bold) Delaunay, distance and shape context features.

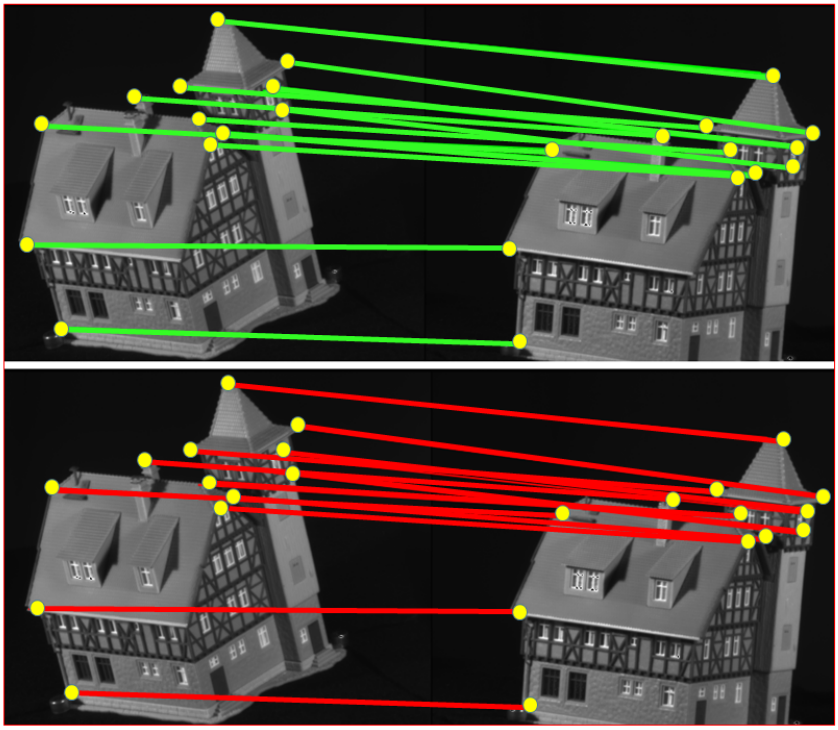


Figure 3.4: House dataset. Matches found for a representative image pair. (Yellow circles) landmarks, (green lines) ground truth, (red lines) proposed method.

We performed a similar experiment on CMU hotel dataset, which contains 101 frames of a video sequence of a toy hotel. Again, we see good overall agreement with the ground truth using the learnt parameters, results are shown in Figure 3.5 and Figure 3.6.

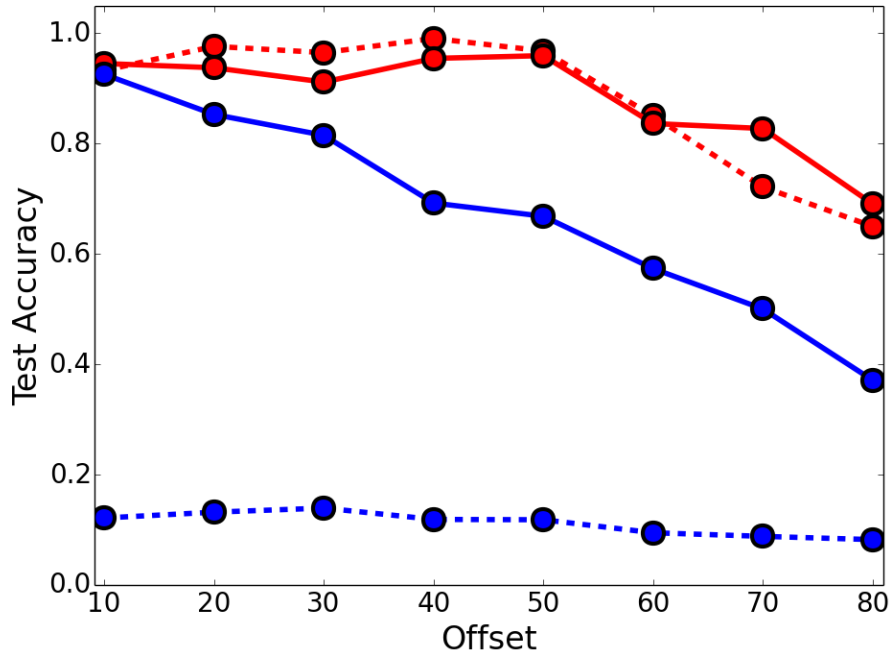


Figure 3.5: Hotel dataset. Proposed learning method (red) compared with no-learn baseline (blue). (Dashed) Delaunay, distance and uninformative features. (Bold) Delaunay, distance and shape context features.

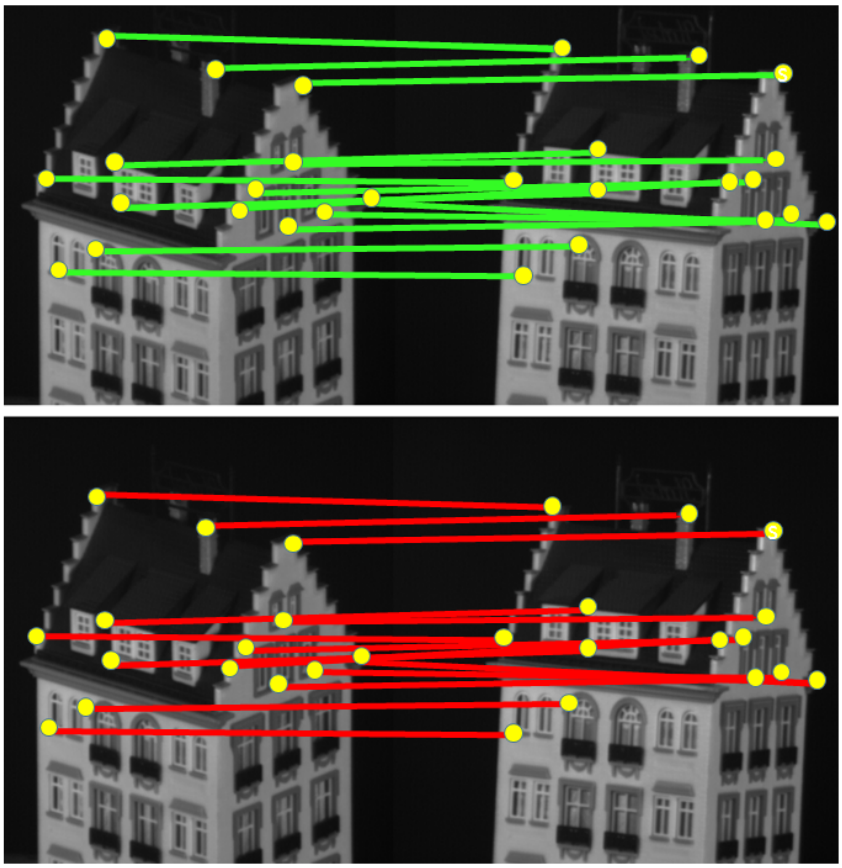


Figure 3.6: Hotel dataset. Matches found for a representative image pair. (Yellow circles) landmarks, (green lines) ground truth, (red lines) proposed method.

Silhouette Dataset: For our second experiment, we used the Silhouette dataset. We applied horizontal shear to twice its width and transformed the images synthetically. Results for this experiment are shown in Figure 3.7 and Figure 3.8.

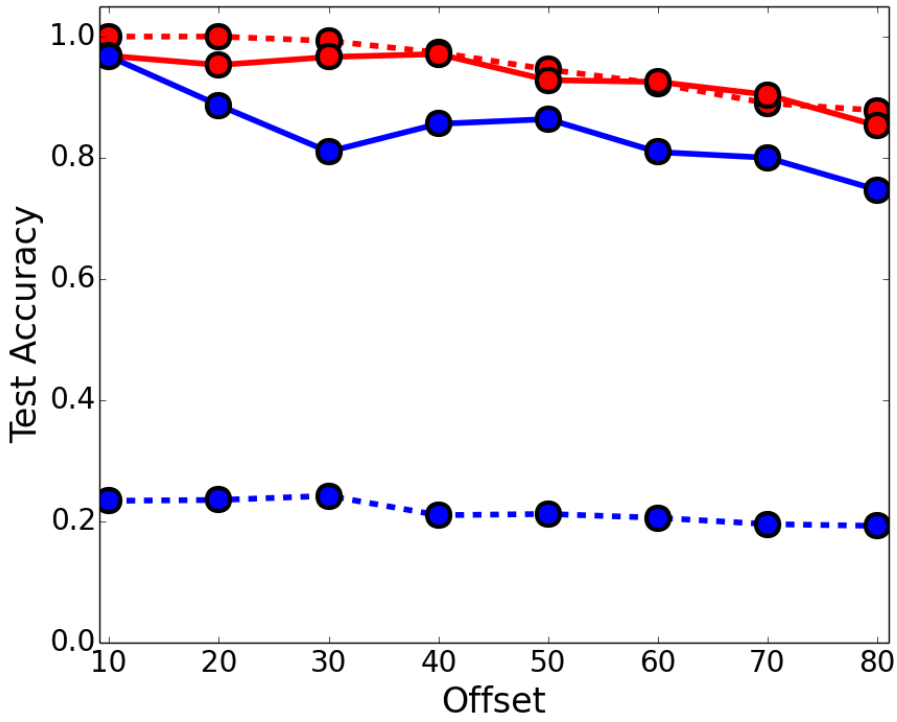


Figure 3.7: Silhouette dataset. Proposed learning method (red) compared with no-learn baseline (blue). (Dashed) Delaunay, distance and uninformative features. (Bold) Delaunay, distance and shape context features.

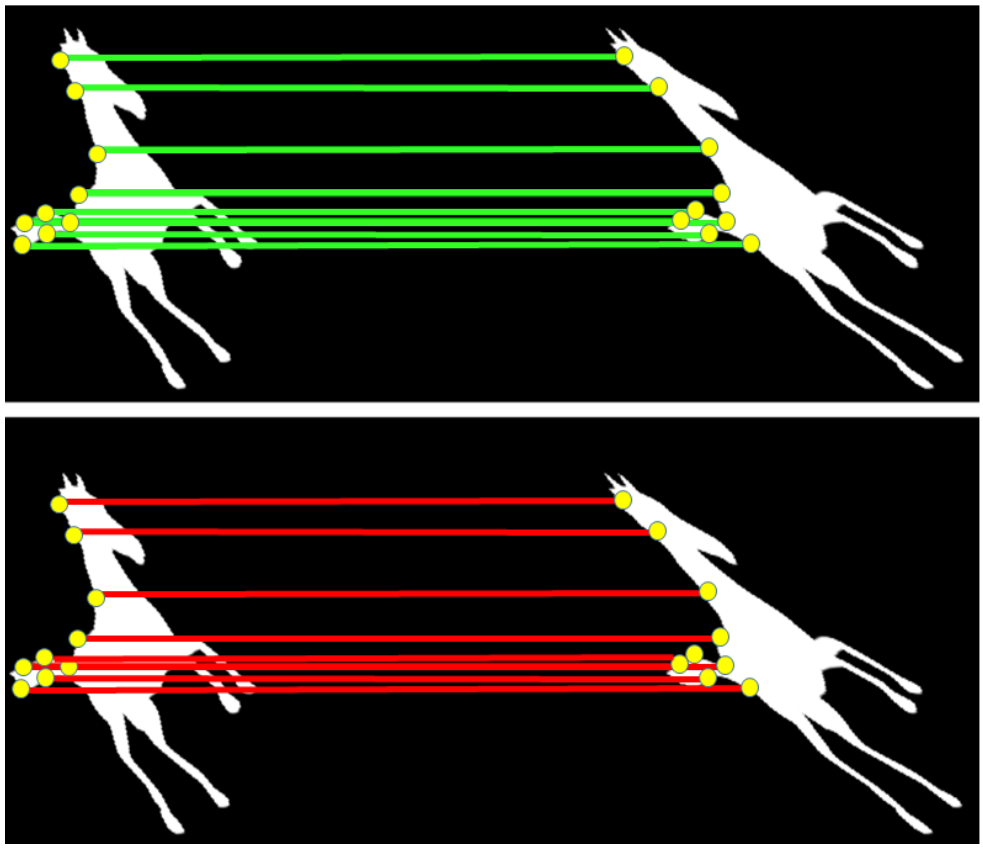


Figure 3.8: Silhouette dataset. Matches found for a representative image pair. (Yellow circles) landmarks, (green lines) ground truth, (red lines) proposed method.

Finally, we analyzed learnt parameters for the setup that includes Delaunay and distance–based adjacency matrices at different scales along with various uninformative features. This setup makes the original matching problem much more challenging. In our experiments, it is evident that the proposed method induced more weight on parameters corresponding to Delaunay and distance, which are known to be the informative features and reduced weights on uninformative features (including non-informative distance scales). Table 3.1 shows an example of average weights produced in 10–fold cross-validation setup for every dataset.

	Delaunay	Dist1	Dist2	Dist3	Uinf	Uinf
House	0.44	0	0	0.47	0	0.01
Shear	0.72	0	0	0.66	0.01	0
Hotel	0.35	0	0	0.58	0	0

Table 3.1: Learnt weights for base QAP objective with Delaunay, distance, and uninformative (Uinf) features based QAPs.

3.5 Summary

In this chapter, we presented a supervised learning algorithm for the graph matching problem in the context of applications in computer vision. Our formulation used ideas from the theory of non-commutative Fourier analysis. In particular, we showed how the representation theory of $\mathbb{S}_n$ makes the procedure computationally tractable, and how Branch and Bound schemes can be modified to learn information relevant for problem instances coming from an application of interest. Our experiments demonstrated that the supervised approach improved the pairwise matching results significantly. The procedure generalizes to other problems that can be cast as appropriate functions on $\mathbb{S}_n$, and provides a complemen-

tary approach to a number of problems typically tackled using structure learning.

Machine learning researchers and practitioners are aware that collecting data with ground truth label is time consuming, but we realized that this task is rather challenging in the matching context because the desired target label is a complex object, i.e., a permutation. Further, this construction of obtaining ground truth data in numerous matching problems involving multiple images is even more challenging. Instead we can look at a different type of supervision such as: what if some minimal training data is available but only in the form of pairwise matching, how can we use such useful data and solve a matching problem beyond pairwise? These challenges prompted us to investigate a more flexible but practically useful framework for solving multi-way matching problem.

4 SOLVING MULTI-WAY MATCHING PROBLEM BY PERMUTATION SYNCHRONIZATION

In this chapter, we consider the problem of matching not just two, but m different sets $X_1, X_2, \ldots, X_m$. Our primary motivation and running example is the classic problem of matching landmarks (feature points) across many images of the same object in computer vision. Presently, multi-matching is usually solved sequentially, by first finding a putative permutation τ_{12} matching X_1 to X_2, then a permutation τ_{23} matching X_2 to X_3, and so on, up to $\tau_{m-1,m}$. While one can conceive of various strategies for optimizing this process, such as the one described in the previous Chapter 3, the fact remains that when the data are noisy, a single error in the sequence will typically create a large number of erroneous pairwise matches (Roberts et al., 2011; Jiang et al., 2012; Volkovs and Zemel, 2012). Further, existing sequential pairwise approaches do not provide any useful insight towards extensions that can handle multiple images.

In this chapter, we investigate a more flexible but practically useful framework for solving multi-way matching problem. In particular, we describe a new method, Permutation Synchronization, that estimates the entire matrix $(\tau_{ji})_{i,j=1}^m$ of assignments *jointly*, in a single shot, and is therefore much more robust to noise. Note that the recovered matchings must satisfy the consistency condition, i.e., $\tau_{kj}\tau_{ji} = \tau_{ki}$. While finding an optimal matrix of permutations satisfying these relations is, in general, combinatorially hard, we show that for the most natural choice of loss function, the problem has a natural relaxation to just finding the n leading eigenvectors of the cost matrix. In addition to vastly reducing the computational cost, using recent results from random matrix theory, we show that the eigenvectors are very effective at aggregating information from all $\binom{m}{2}$ pairwise matches, and therefore make the algorithm surprisingly robust to noise. Our experiments show that in landmark matching problems

Permutation Synchronization can recover the correct correspondence between landmarks across a large number of images with small error, even when a significant fraction of the pairwise matches are incorrect.

The term "synchronization" is inspired by the recent celebrated work of Singer et al. on a similar problem involving finding the right rotations (rather than matchings) between electron microscopic images (Singer and Shkolnisky, 2011; Hadani and Singer, 2011a,b). Historically, multi-matching has received relatively little attention. However, independently of, and concurrently with the present work, Huang and Guibas (Huang and Guibas, 2013) have recently proposed a semidefinite programming based solution, which parallels our approach.

4.1 Synchronizing permutations

Consider a collection of m sets $X_1, X_2, \ldots, X_m$ of n objects each, $X_i = \{x_1^i, x_2^i, \ldots, x_n^i\}$, such that for each pair (X_i, X_j), each x_p^i in X_i has a natural counterpart x_q^j in X_j, Figure 4.1. For example, in computer vision, given

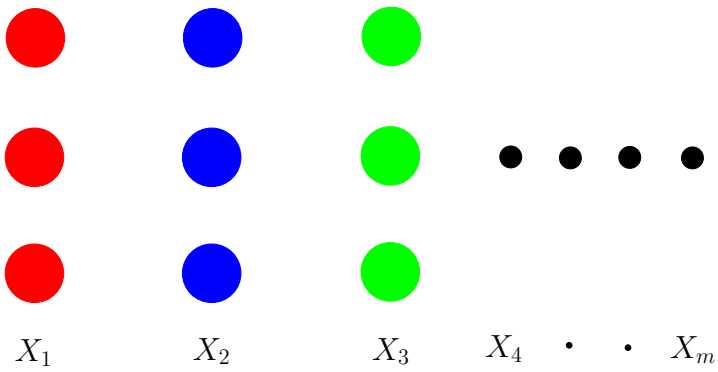


Figure 4.1: Red balls correspond to $n = 3$ objects $\{x_1^1, x_2^1, x_3^1\}$ in set X_1. Similarly, other sets are represented in blue, green and so on. Multi-way matching problem find a consistent bijection $\tau_{ji} : \tau_{ji} \Rightarrow x_p^i \sim x^j_{\tau_{ji}(p)}$ $\forall$ each pair (X_i, X_j)

m images of the same scene taken from different viewpoints, $x_1^i, x_2^i, \ldots, x_n^i$ might be n visual landmarks detected in image i, while $x_1^j, x_2^j, \ldots, x_n^j$ are n landmarks detected in image j, in which case $x_p^i \sim x_q^j$ signifies that x_p^i and x_q^j correspond to the same physical feature.

Since the correspondence between X_i and X_j is a bijection, one can write it as $x_p^i \sim x^j_{\tau_{ji}(p)}$ for some permutation $\tau_{ji} \colon \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$. Key to our approach to solving multi-matching is that with respect to the natural definition of multiplication, $(\tau'\tau)(i) := (\tau'(\tau(i))$, the $n!$ possible permutations of $\{1, 2, \ldots, n\}$ form a $group$, called the symmetric group of degree n, denoted $\mathbb{S}_n$.

We say that the system of correspondences between $X_1, X_2, \ldots, X_m$ is *consistent* if $x_p^i \sim x_q^j$ and $x_q^j \sim x_r^k$ together imply that $x_p^i \sim x_r^k$. In terms of permutations this is equivalent to requiring that the array $(\tau_{ij})_{i,j=1}^m$ satisfy

$$\tau_{kj}\tau_{ji} = \tau_{ki} \qquad \forall i, j, k. \tag{4.1}$$

Alternatively, given some reference ordering of $x_1, x_2, \ldots, x_n$, we can think of each X_i as realizing its own permutation σ_i (in the sense of $x_\ell \sim x^i_{\sigma_i(\ell)}$), and then τ_{ji} becomes

$$\tau_{ji} = \sigma_j \sigma_i^{-1}. \tag{4.2}$$

The existence of permutations $\sigma_1, \sigma_2, \ldots, \sigma_m$ satisfying Equation (4.2) is equivalent to requiring that $(\tau_{ji})_{i,j=1}^m$ satisfy Equation (4.1). Thus, assuming consistency, solving the multi-matching problem reduces to finding just m different permutations, rather than $O(m^2)$, Figure 4.2. However, the σ_i's are of course not directly observable. Rather, in a typical application we have some tentative (noisy) $\tilde{\tau}_{ji}$ matchings which we must *synchronize* into the form Equation (4.2) by finding the underlying $\sigma_1, \ldots, \sigma_m$, Figure 4.3.

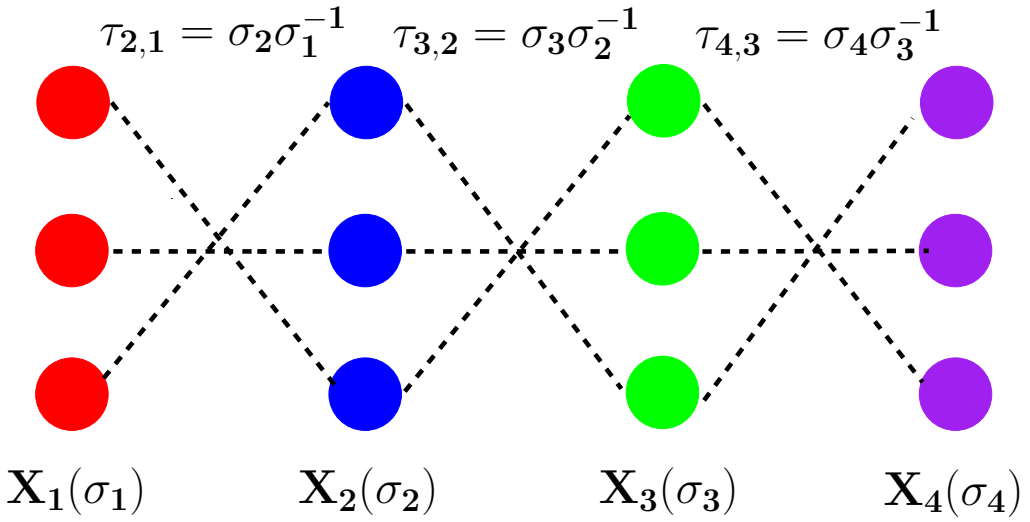


Figure 4.2: Reference set model for a 4-way matching problem such that $\tau_{ji} = \sigma_j\sigma_i^{-1}$.

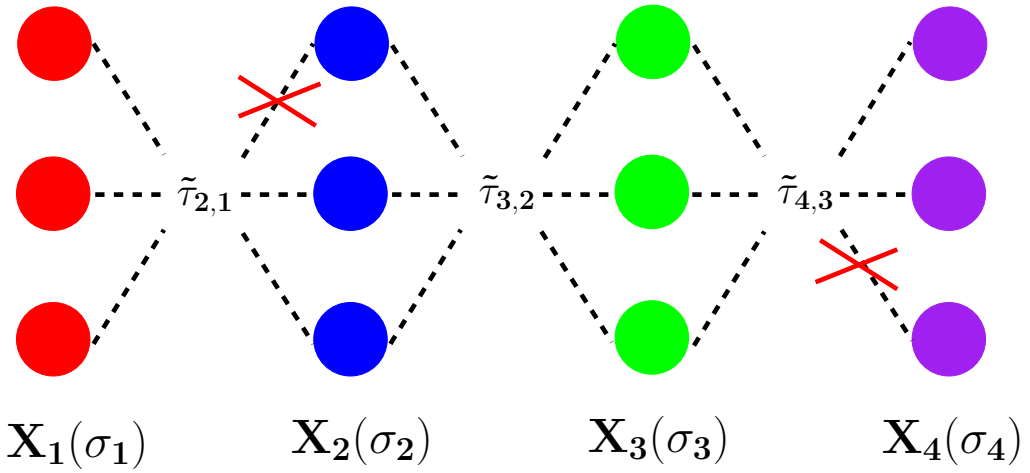


Figure 4.3: Noisy pairwise observations in a 4-way matching problem.

Given $(\tilde{\tau}_{ji})_{i,j=1}^{m}$ and some appropriate distance metric d between permutations, we formalize Permutation Synchronization as the combinatorial optimization problem

$$\underset{\sigma_1,\sigma_2,\ldots,\sigma_m \in \mathbb{S}_n}{\text{minimize}} \sum_{i,j=1}^{N} d(\sigma_j \sigma_i^{-1}, \tilde{\tau}_{ji}). \tag{4.3}$$

The computational cost of solving (4.3) depends critically on the form of the distance metric d, and the choice representation. A matrix valued function $\rho\colon \mathbb{S}_n \longrightarrow \mathbb{C}^{d\times d}$ is said to be a representation of the symmetric group if $\rho(\sigma_2)\,\rho(\sigma_1) = \rho(\sigma_2\sigma_1)$ for any pair of permutations $\sigma_1, \sigma_2 \in \mathbb{S}_n$ as described in Chapter 2 for more details on the representation theory of $\mathbb{S}_n$. Moreover, if ρ is a unitary representation then we can limit ourselves to the simplest choice

$$d(\sigma, \tau) = 1 - \langle \rho(\sigma), \rho(\tau) \rangle\,, \tag{4.4}$$

where $\langle A, B\rangle$ is the matrix inner product

$$\langle A, B\rangle := \operatorname{tr}(A^\top B) = \sum_{p,q=1}^{n} A_{p,q}\, B_{p,q}.$$

Furthermore, it allow us to rewrite Equation (4.3) as a maximization problem

$$\underset{\sigma_1,\sigma_2,\ldots,\sigma_m \in \mathbb{S}_n}{\text{maximize}} \sum_{i,j=1}^{m} \langle \rho(\sigma_j \sigma_i^{-1}),\ \rho(\tilde{\tau}_{ji}) \rangle. \tag{4.5}$$

4.1.1 Representations and eigenvectors

One can choose a representation to balance the trade-off between computational complexity and representational richness. We use the permutation matrix representation in our analysis. The permutation matrix representa-

tion $P(\sigma)$ is a matrix representation in $\mathbb{R}^{n\times n}$ such that

$$[P(\sigma)]_{q,p} := \begin{cases} 1 & \text{if } \sigma(p) = q \\ 0 & \text{otherwise.} \end{cases}$$

The objective in (4.4) simply counts the total number of objects assigned *differently* by σ and τ (with proper normalization). Further, we can rewrite (4.5) as

$$\underset{\sigma_1,\sigma_2,\ldots,\sigma_m\in\mathbb{S}_n}{\text{maximize}} \sum_{i,j=1}^{m} \langle P(\sigma_j\sigma_i^{-1}),\ P(\tilde{\tau}_{ji})\rangle,$$

suggesting the generalization

$$\underset{\sigma_1,\sigma_2,\ldots,\sigma_m\in\mathbb{S}_n}{\text{maximize}} \sum_{i,j=1}^{m} \left\langle P(\sigma_j\sigma_i^{-1}),\ T_{ji}\right\rangle, \tag{4.6}$$

where the T_{ji}'s can now be any matrices, subject to $T_{ji}^\top = T_{ij}$. Intuitively, each T_{ji} is an objective matrix, the (q,p) element of which captures the utility of matching x_p^i in X_i to x_q^j in X_j. This generalization is very useful when the assignments of the different x_p^i's have different confidences. For example, in the landmark matching case, if, due to occlusion or for some other reason, the counterpart of x_p^i is not present in X_j, then we can simply set $[T_{ji}]_{q,p} = 0$ for all q.

The generalized Permutation Synchronization problem (4.6) can also be written as

$$\underset{\sigma_1,\sigma_2,\ldots,\sigma_m\in\mathbb{S}_n}{\text{maximize}} \langle \mathcal{P},\ \mathcal{T}\rangle, \tag{4.7}$$

where

$$\mathcal{P} = \begin{pmatrix} P(\sigma_1\sigma_1^{-1}) & \ldots & P(\sigma_1\sigma_m^{-1}) \\ \vdots & \ddots & \vdots \\ P(\sigma_m\sigma_1^{-1}) & \ldots & P(\sigma_m\sigma_m^{-1}) \end{pmatrix} \quad \text{and} \quad \mathcal{T} = \begin{pmatrix} T_{11} & \ldots & T_{1m} \\ \vdots & \ddots & \vdots \\ T_{m1} & \ldots & T_{mm} \end{pmatrix}. \tag{4.8}$$

Clearly, P is a representation of $\mathbb{S}_n$ (actually, the so-called defining representation), since $P(\sigma_2\sigma_1) = P(\sigma_2)\,P(\sigma_1)$. Moreover, P is orthogonal, because each $P(\sigma)$ is real and $P(\sigma^{-1}) = P(\sigma)^\top$. For example, consider a 3-way matching problem where each set X_1, X_2, X_3 has 4 objects each, $X_i = \{x_1^i, x_2^i, \ldots, x_4^i\}$. For the following observed pairwise matchings

$$\begin{aligned}
\tilde{\tau}_{21} &= (1\,4\,3\,2) & \qquad \tilde{\tau}_{12} &= (1\,2\,3\,4)\\
\tilde{\tau}_{23} &= (1\,3\,2\,4) & \qquad \tilde{\tau}_{32} &= (1\,4\,2\,3)\\
\tilde{\tau}_{31} &= (1\,4\,3\,2) & \qquad \tilde{\tau}_{13} &= (1\,2\,3\,4)\,,
\end{aligned}$$

the objective matrix T can be written as

$$T = \left[\begin{array}{c|c|c} T_{11} & T_{12} & T_{13} \\ \hline T_{21} & T_{22} & T_{23} \\ \hline T_{31} & T_{32} & T_{33} \end{array}\right] .$$

Here, each block T_{ij} is a permutation matrix of size 4×4, i.e., $P(\tilde{\tau}_{ij})$. More specifically,

$$T = \left[\begin{array}{cccc|cccc|cccc}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

Our fundamental observation is that $\mathcal{P}$ has a very special form.

Proposition 4.1. *The synchronization matrix $\mathcal{P}$ is of rank n and is of the form $\mathcal{P} = U \cdot U^\top$, where*

$$U = \begin{pmatrix} P(\sigma_1) \\ \vdots \\ P(\sigma_m) \end{pmatrix}.$$

Proof. From P being a representation of $\mathbb{S}_n$,

$$\mathcal{P} = \begin{pmatrix} P(\sigma_1)\,P(\sigma_1)^\top & \dots & P(\sigma_1)\,P(\sigma_m)^\top \\ \vdots & \ddots & \vdots \\ P(\sigma_m)\,P(\sigma_1)^\top & \dots & P(\sigma_m)\,P(\sigma_m)^\top \end{pmatrix}, \tag{4.9}$$

implying $\mathcal{P} = U \cdot U^\top$. Since U has n columns, rank$(\mathcal{P})$ is at most n. This rank is achieved because $P(\sigma_1)$ is an orthogonal matrix, therefore it has linearly independent columns, and consequently the columns of U cannot be linearly dependent. ■

Corollary 1. *Letting $[P(\sigma_i)]_p$ denote the p'th column of $P(\sigma_i)$, the normalized columns of U,*

$$u_\ell = \frac{1}{\sqrt{m}} \begin{pmatrix} [P(\sigma_1)]_\ell \\ \vdots \\ [P(\sigma_m)]_\ell \end{pmatrix} \qquad \ell = 1, \dots, n, \tag{4.10}$$

are mutually orthogonal unit eigenvectors of $\mathcal{P}$ with the same eigenvalue m, and together span the row/column space of $\mathcal{P}$.

Proof. The columns of U are orthogonal because the columns of each constituent $P(\sigma_i)$ are orthogonal. The normalization follows from each column of $P(\sigma_i)$ having norm 1. The rest follows by Proposition 4.1. ■

4.1.2 An easy relaxation

Solving (4.7) is computationally difficult, because it involves searching the combinatorial space of a combination of m permutations. However, Proposition 4.1 and its corollary suggest relaxing it to

$$\underset{\mathcal{P}\in\mathfrak{M}_m^n}{\text{maximize}} \langle \mathcal{P},\, \mathcal{T} \rangle \,, \tag{4.11}$$

where $\mathfrak{M}_n^m$ is the set of mn–dimensional rank n symmetric matrices whose non-zero eigenvalues are m. This is now just a generalized Rayleigh problem, the solution of which is simply

$$\mathcal{P} = m \sum_{\ell=1}^{n} v_\ell \, v_\ell^\top , \tag{4.12}$$

where $v_1, v_2, \ldots, v_\ell$ are the n leading normalized eigenvectors of $\mathcal{T}$. Equivalently, $\mathcal{P} = U \cdot U^\top$, where

$$U = \sqrt{m} \begin{pmatrix} | & | & \ldots & | \\ v_1 & v_2 & \ldots & v_n \\ | & | & \ldots & | \end{pmatrix} . \tag{4.13}$$

Thus, in contrast to the original combinatorial problem, (4.11) can be solved by just finding the m leading eigenvectors of $\mathcal{T}$.

Of course, from $\mathcal{P}$ we must still recover the individual permutations $\sigma_1, \sigma_2, \ldots, \sigma_m$. However, as long as $\mathcal{P}$ is relatively close in form (4.8), this is quite a simple and stable process. One way to do it is to let each σ_i be the permutation that best matches the $(i, 1)$ block of $\mathcal{P}$ in the linear assignment sense,

$$\sigma_i = \arg\min_{\sigma\in\mathbb{S}_n} \langle P(\sigma), [\mathcal{P}]_{i,1} \rangle \,,$$

which is solved in $O(n^3)$ time by the Kuhn–Munkres algorithm (Kuhn,

1955)[1], and then set $\tau_{ji} = \sigma_j \sigma_i^{-1}$, which will then satisfy the consistency relations. The pseudo-code of the full algorithm is given in Algorithm 1.

Algorithm 1 Permutation Synchronization

Require: the objective matrix $\mathcal{T}$
  Compute the n leading eigenvectors $(v_1, v_2, \ldots, v_n)$ of $\mathcal{T}$ and set $U = \sqrt{m}\,[v_1, v_2, \ldots, v_n]$
  for $i = 1$ to m **do**
    $P_{i1} = U_{(i-1)n+1:in,\,1:n}\, U_{1:n,\,1:n}^{\top}$
    $\sigma_i = \arg\max_{\sigma \in \mathbb{S}_n} \langle P_{i1}, \sigma \rangle$ [Kuhn-Munkres]
  end for
  for each (i, j) **do**
    $\tau_{ji} = \sigma_j \sigma_i^{-1}$
  end for
Ensure: the matrix $(\tau_{ji})_{i,j=1}^{m}$ of globally consistent matchings

4.2 Analysis of the relaxed algorithm

Let us now investigate under what conditions we can expect the relaxation (4.11) to work well, in particular, in what cases we can expect the recovered matchings to be *exact*.

In the absence of noise, i.e., when $T_{ji} = P(\tilde{\tau}_{ji})$ for some array $(\tilde{\tau}_{ji})_{j,i}$ of permutations that already satisfy the consistency relations (4.1), $\mathcal{T}$ will have precisely the same structure as described by Proposition 4.1 for $\mathcal{P}$. In

[1] Note that we could equally well have matched the σ_i's to any other column of blocks, since they are only defined relative to an arbitrary reference permutation: if, for any fixed σ_0, each σ_i is redefined as $\sigma_i \sigma_0$, the predicted relative permutations $\tau_{ji} = \sigma_j \sigma_0 (\sigma_i \sigma_0)^{-1} = \sigma_j \sigma_i^{-1}$ stay the same.

particular, it will have n mutually orthogonal eigenvectors

$$v_\ell = \frac{1}{\sqrt{m}} \begin{pmatrix} [P(\tilde{\sigma}_1)]_\ell \\ \vdots \\ [P(\tilde{\sigma}_m)]_\ell \end{pmatrix} \qquad \ell = 1, \ldots, n \tag{4.14}$$

with the same eigenvalue m. Due to the n–fold degeneracy, however, the matrix of eigenvectors (4.13) is only defined up to multiplication by an arbitrary rotation matrix O on the right, which means that instead of the "correct" U (whose columns are (4.14)), the eigenvector decomposition of $\mathcal{T}$ may return any $U' = UO$. Fortunately, when forming the product

$$\mathcal{P} = U' \cdot U'^{\top} = U\, O\, O^{\top} U^{\top} = U \cdot U^{\top}$$

this rotation cancels, confirming that our algorithm recovers $\mathcal{P} = \mathcal{T}$, and hence the matchings $\tau_{ji} = \tilde{\tau}_{ji}$, with no error.

Of course, rather than the case when the solution is handed to us from the start, we are more interested in how the algorithm performs in situations when either the T_{ji} blocks are not permutation matrices, or they are not synchronized. To this end, we set

$$\mathcal{T} = \mathcal{T}_0 + \mathcal{N}, \tag{4.15}$$

where $\mathcal{T}_0$ is the correct "ground truth" synchronization matrix, while $\mathcal{N}$ is a symmetric perturbation matrix with entries drawn independently from a zero-mean normal distribution with variance η^2.

In general, to find the permutation best aligned with a given $n \times n$ matrix T, the Kuhn–Munkres algorithm solves for

$$\begin{aligned} \widehat{\tau} &= \arg\max_{\tau \in \mathbb{S}_n} \langle P(\tau), T \rangle \\ &= \arg\max_{\tau \in \mathbb{S}_n} (\mathrm{vec}(P(\tau)) \cdot \mathrm{vec}(T)) \,. \end{aligned} \tag{4.16}$$

Therefore, writing $T = P(\tau_0) + \epsilon$, where $P(\tau_0)$ is the "ground truth", while ϵ is an error term, it is guaranteed to return the correct permutation as long as

$$\| \operatorname{vec}(\epsilon) \| < \min_{\tau' \in \mathbb{S}_n \setminus \{\tau_0\}} \| \operatorname{vec}(\tau_0) - \operatorname{vec}(\tau') \| /2.$$

By the symmetry of $\mathbb{S}_n$, the right hand side is the same for any τ_0, so w.l.o.g. we can set $\tau_0 = e$ (the identity), and find that the minimum is achieved when τ' is just a transposition, e.g., the permutation that swaps 1 with 2 and leaves $3, 4, \ldots, n$ in place. The corresponding permutation matrix differs from the identity in exactly 4 entries, therefore a sufficient condition for correct reconstruction is that $\|\epsilon\|_{\text{Frob}} = \langle \epsilon, \epsilon \rangle^{1/2} = \|\text{vec}(\epsilon)\| < \frac{1}{2}\sqrt{4} = 1$. As n grows, $\|\epsilon\|_{\text{Frob}}$ becomes tightly concentrated around ηn, so the condition for recovering the correct permutation is $\eta < 1/n$.

Permutation Synchronization can achieve a lower error, especially in the large m regime, because the eigenvectors aggregate information from all the T_{ji} matrices, and tend to be very stable to perturbations. In general, perturbations of the form (4.15) exhibit a characteristic phase transition. As long as the largest eigenvalue of the random matrix $\mathcal{N}$ falls below a given multiple of the smallest non-zero eigenvalue of $\mathcal{T}_0$, adding $\mathcal{N}$ will have very little effect on the eigenvectors of $\mathcal{T}$. On the other hand, when the noise exceeds this limit, the spectra get fully mixed, and it becomes impossible to recover $\mathcal{T}_0$ from $\mathcal{T}$ to any precision at all.

If $\mathcal{N}$ is a symmetric matrix with independent $\mathcal{N}(0, \eta^2)$ entries, as $nm \to \infty$, its spectrum will tend to Wigner's famous semicircle distribution supported on the interval $(-2\eta(nm)^{1/2}, 2\eta(nm)^{1/2})$, and with probability one the largest eigenvalue will approach $2\eta(nm)^{1/2}$ (Wigner, 1958; Füredi and Komlós, 1981). In contrast, the non-zero eigenvalues of $\mathcal{T}_0$ scale with m, which guarantees that for large enough m the two spectra will be nicely separated and Permutation Synchronization will have very low error. While much harder to analyze analytically, empirical evidence suggests that this type of phase transition behavior is characteristic of any reasonable

noise model, for example the one in which we take each block of $\mathcal{T}$ and with some probability p replace it with a random permutation matrix (Figures 4.4 to 4.6).

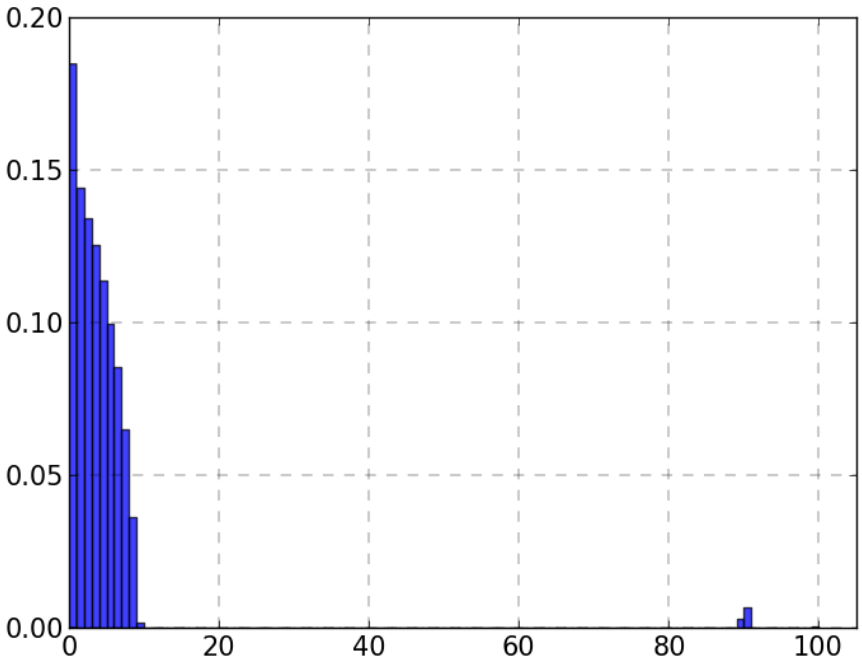


Figure 4.4: Singular value histogram of $\mathcal{T}$ under the noise model where each $\tilde{\tau}_{ji}$ with probability $p = \{0.10\}$ is replaced by a random permutation ($m = 100$, $n = 30$). Note that apart from the extra peak at zero, the distribution of the stochastic eigenvalues is very similar to the semicircular distribution for Gaussian noise. As long as the small cluster of deterministic eigenvalues is clearly separated from the noise, Permutation Synchronization is feasible.

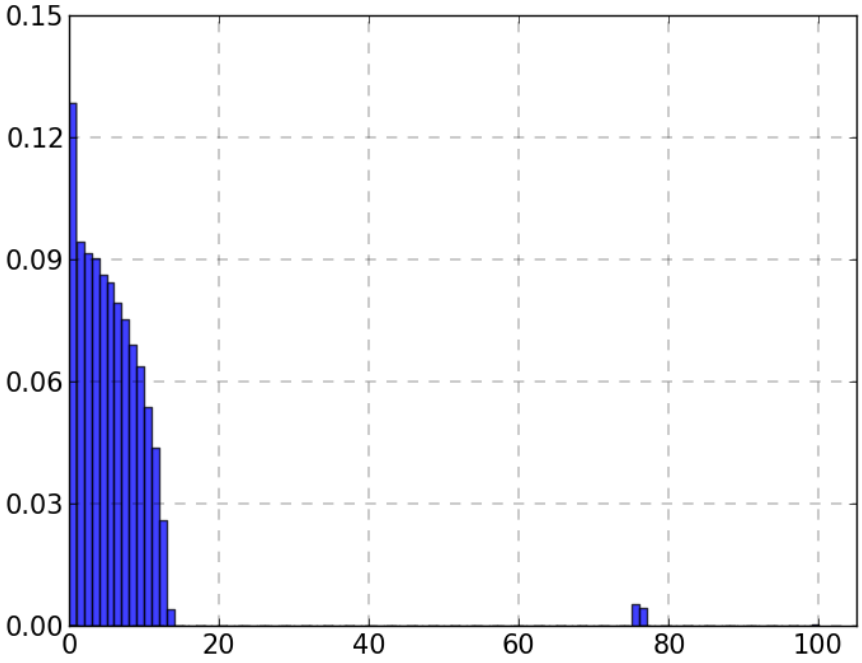


Figure 4.5: Singular value histogram of $\mathcal{T}$ under the noise model where each $\tilde{\tau}_{ji}$ with probability $p = \{0.25\}$ is replaced by a random permutation ($m = 100$, $n = 30$). Note that apart from the extra peak at zero, the distribution of the stochastic eigenvalues is very similar to the semicircular distribution for Gaussian noise. As long as the small cluster of deterministic eigenvalues is clearly separated from the noise, Permutation Synchronization is feasible.

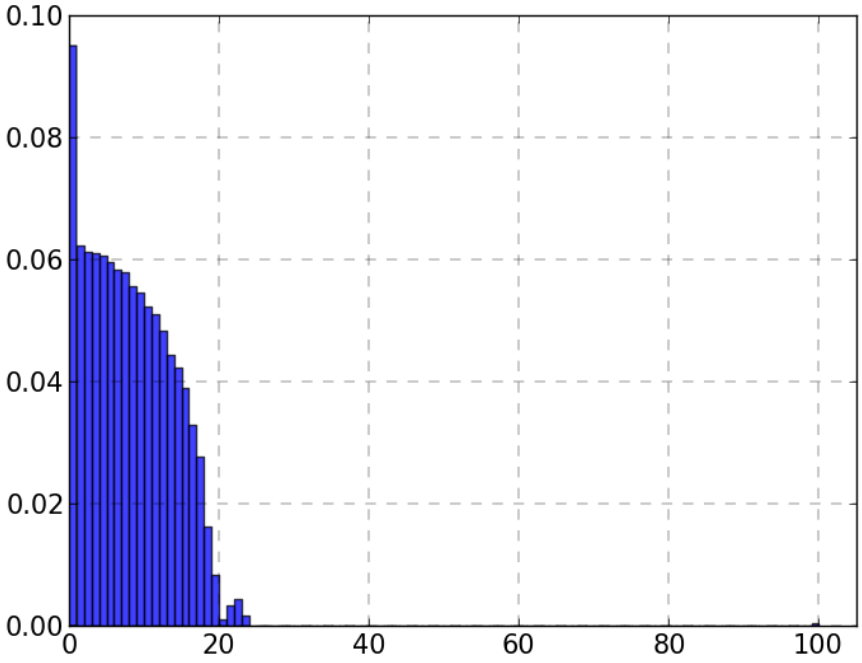


Figure 4.6: Singular value histogram of $\mathcal{T}$ under the noise model where each $\tilde{\tau}_{ji}$ with probability $p = \{0.85\}$ is replaced by a random permutation ($m = 100$, $n = 30$). Note that apart from the extra peak at zero, the distribution of the stochastic eigenvalues is very similar to the semicircular distribution for Gaussian noise. As long as the small cluster of deterministic eigenvalues is clearly separated from the noise, Permutation Synchronization is feasible.

To derive more quantitative results, we consider the case where $\mathcal{N}$ is a so-called (symmetric) Gaussian Wigner matrix, which has independent $\mathcal{N}(0, \eta^2)$ entries on its diagonal, and $\mathcal{N}(0, \eta^2/2)$ entries everywhere else. It has recently been proved that for this type of matrix the phase transition occurs at $\lambda_{\text{min}}^{\text{det}}/\lambda_{\text{max}}^{\text{stochastic}} = 1/2$, so to recover $\mathcal{T}_0$ to any accuracy at all we must have $\eta < (m/n)^{1/2}$ (Benaych-Georges and Nadakuditi, 2011). Below this limit, to quantify the actual expected error, we write each leading normalized eigenvector $v_1, v_2, \ldots, v_n$ of $\mathcal{T}$ as $v_i = v_i^* + v_i^\perp$, where v_i^* is the projection of v_i to the space $\mathcal{U}_0$ spanned by the non-zero eigenvectors $v_1^0, v_2^0, \ldots, v_n^0$ of $\mathcal{T}_0$. By Theorem 2.2 of (Benaych-Georges and Nadakuditi, 2011) as $nm \longrightarrow \infty$,

$$\|v_i^*\|^2 \xrightarrow{a.s.} 1 - \eta^2 \frac{n}{m} \qquad \text{and} \qquad \|v_i^\perp\|^2 \xrightarrow{a.s.} \eta^2 \frac{n}{m}\,. \tag{4.17}$$

It is easy to see that $\langle v_i^\perp, v_j^\perp \rangle \xrightarrow{\text{a.s.}} 0$, which implies $\langle v_i^*, v_j^* \rangle = \langle v_i, v_j \rangle - \langle v_i^\perp, v_j^\perp \rangle \xrightarrow{\text{a.s.}} 0$, so, setting $\lambda = (1 - \eta^2 n/m)^{-1/2}$, the normalized vectors $\lambda v_1^*, \ldots, \lambda v_n^*$ almost surely tend to an orthonormal basis for $\mathcal{U}_0$. Thus, $U = \sqrt{m}\,[v_1, \ldots, v_n]$ is related to the "true" $U_0 = \sqrt{m}\,[v_1^0, \ldots, v_n^0]$ by

$$\lambda U \xrightarrow{\text{a.s.}} U_0 O + \lambda E' = (U_0 + \lambda E) O,$$

where O is some rotation and each column of the noise matrices E and E' has norm $\eta(n/m)^{1/2}$. Since multiplying U on the right by an orthogonal matrix does not affect $\mathcal{P}$, and the Kuhn–Munkres algorithm is invariant to scaling by a constant, this equation tells us that (almost surely) the effect of (4.15) is equivalent to setting $U = U_0 + \lambda E$. In terms of the individual P_{ji} blocks of $\mathcal{P} = UU^\top$, neglecting second order terms,

$$P_{ji} = (U_j^0 + \lambda E_j)(U_i^0 + \lambda E_i)^\top \approx P(\tau_{ji}) + \lambda U_j^0 E_i^\top + \lambda E_j U_i^{0\top},$$

where τ_{ji} is the ground truth matching and U_i^0 and E_i denote the appro-

priate $n \times n$ submatrices of U^0 and E. Conjecturing that in the limit E_i and E_j follow rotationally invariant distributions, almost surely

$$\lim \| U_j^0 E_i^\top + E_j U_i^{0\top} \|_{\text{Frob}} = \lim \| E_i + E_j \|_{\text{Frob}} \leq 2\eta n/m.$$

Thus, plugging in to our earlier result for the error tolerance of the Kuhn–Munkres algorithm, Permutation Synchronization will correctly recover τ_{ji} with probability one provided $2\lambda\eta n/m < 1$, or, equivalently,

$$\eta^2 < \frac{m/n}{1 + 4(m/n)^{-1}} \,.$$

This is much better than our $\eta < 1/n$ result for the naive algorithm, and remarkably only slightly stricter than the condition $\eta < (m/n)^{1/2}$ for recovering the eigenvectors with any accuracy at all. Of course, these results are asymptotic (in the sense of $nm \longrightarrow \infty$), and strictly speaking only apply to additive Gaussian Wigner noise. However, as Figures 4.7 to 4.9 show, in practice, even when the noise is in the form of corrupting entire permutations and nm is relatively small, qualitatively our algorithm exhibits the correct behavior, and for large enough m Permutation Synchronization does indeed recover *all* $(\tau_{ji})_{j,i=1}^m$ with no error even when the vast majority of the entries in $\mathcal{T}$ are incorrect.

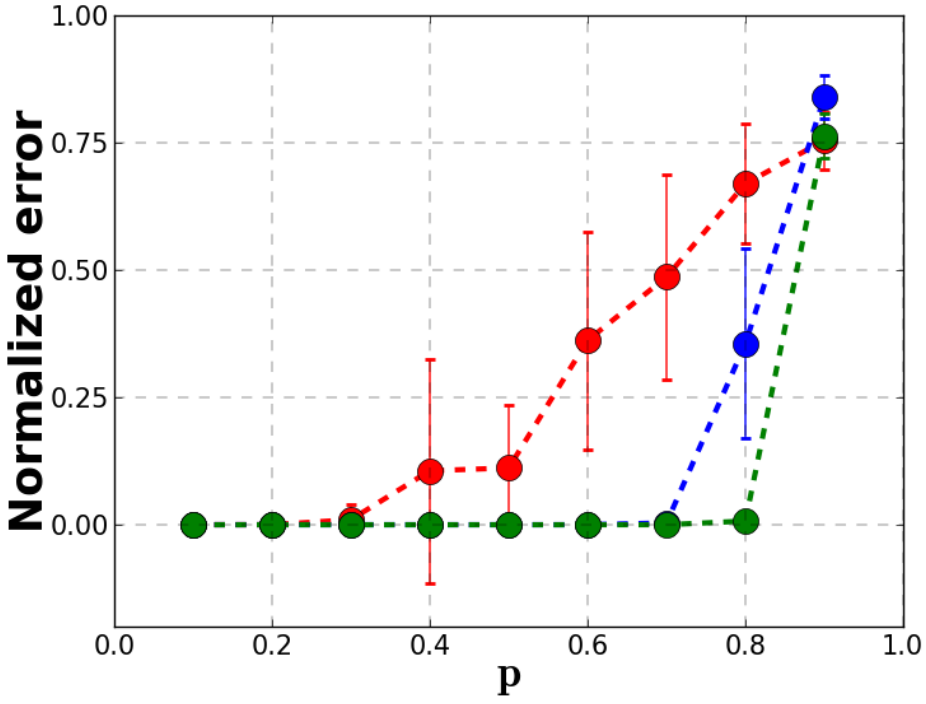


Figure 4.7: The fraction of $(\sigma_i)_{i=1}^m$ permutations that are incorrect when reconstructed by Permutation Synchronization from an array $(\tilde{\tau}_{ji})_{j,i=1}^m$, in which each entry, with probability p is replaced by a random permutation. The plots show the mean and standard deviation of errors over 20 runs as a function of p for $n = 10$. $m = 10$ (red), $m = 50$ (blue) and $m = 100$ (green).

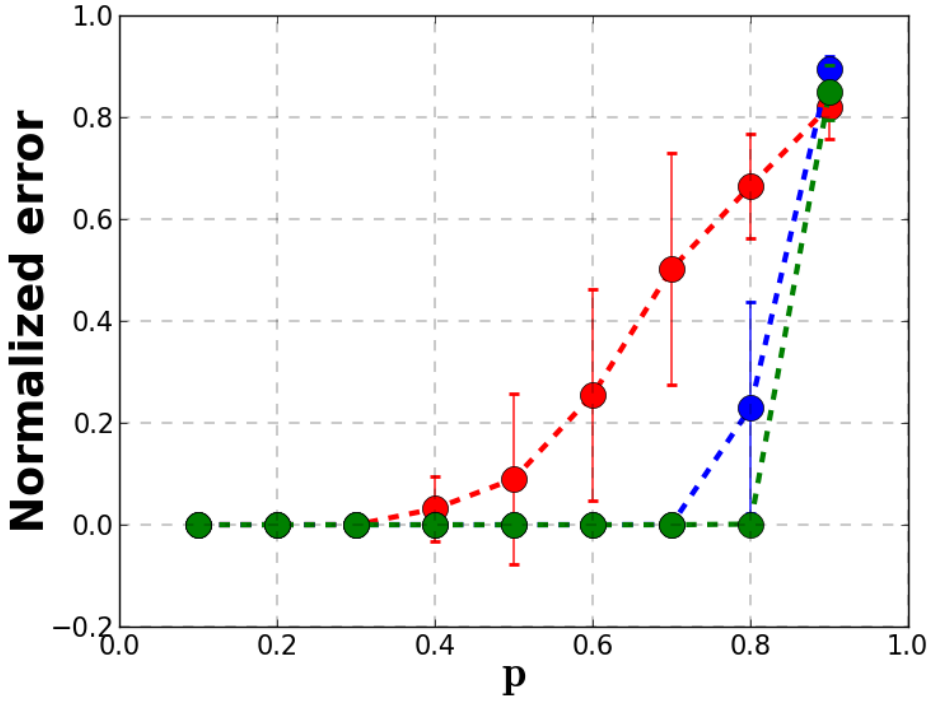


Figure 4.8: The fraction of $(\sigma_i)_{i=1}^m$ permutations that are incorrect when reconstructed by Permutation Synchronization from an array $(\tilde{\tau}_{ji})_{j,i=1}^m$, in which each entry, with probability p is replaced by a random permutation. The plots show the mean and standard deviation of errors over 20 runs as a function of p for $n = 25$. $m = 10$ (red), $m = 50$ (blue) and $m = 100$ (green).

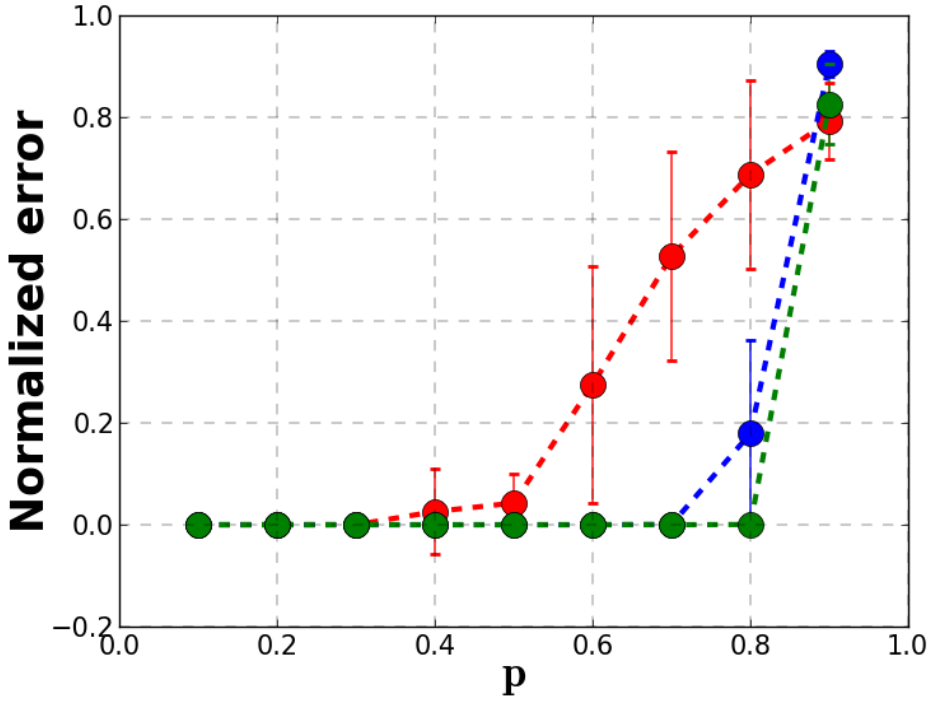


Figure 4.9: The fraction of $(\sigma_i)_{i=1}^m$ permutations that are incorrect when reconstructed by Permutation Synchronization from an array $(\tilde{\tau}_{ji})_{j,i=1}^m$, in which each entry, with probability p is replaced by a random permutation. The plots show the mean and standard deviation of errors over 20 runs as a function of p for $n = 30$. $m = 10$ (red), $m = 50$ (blue) and $m = 100$ (green).

4.3 Experiments

Since computer vision is one of the areas where improving the accuracy of multi-matching problems is the most pressing, our experiments focused on this domain.

4.3.1 2D Image Matching

As a proof of principle, we considered the task of aligning landmarks in 2D images of the same object taken from different viewpoints in the **CMU house** ($m = 111$ frames of a video sequence of a toy house with $n = 30$ hand labeled landmark points in each frame). The baseline method is to compute $(\tilde{\tau}_{ji})_{i,j=1}^{m}$ by solving $\binom{m}{2}$ independent linear assignment problems based on matching landmarks by their shape context features (Belongie et al., 2002). Our method takes the same pairwise matches and synchronizes them with the eigenvector based procedure. Figure 4.10 shows that this clearly outperforms the baseline, which tends to degrade progressively as the number of images increases. This is due to the fact that the appearance (or descriptors) of keypoints differ considerably for large offset pairs (which is likely when the image set is large), leading to many false matches. In contrast, our method improves as the size of the image set increases.

While simple, this experiment demonstrates the utility of Permutation Synchronization for multi-view stereo matching, showing that instead of heuristically propagating local pairwise matches, it can find a much more accurate globally consistent matching at little additional cost, Figure 4.11.

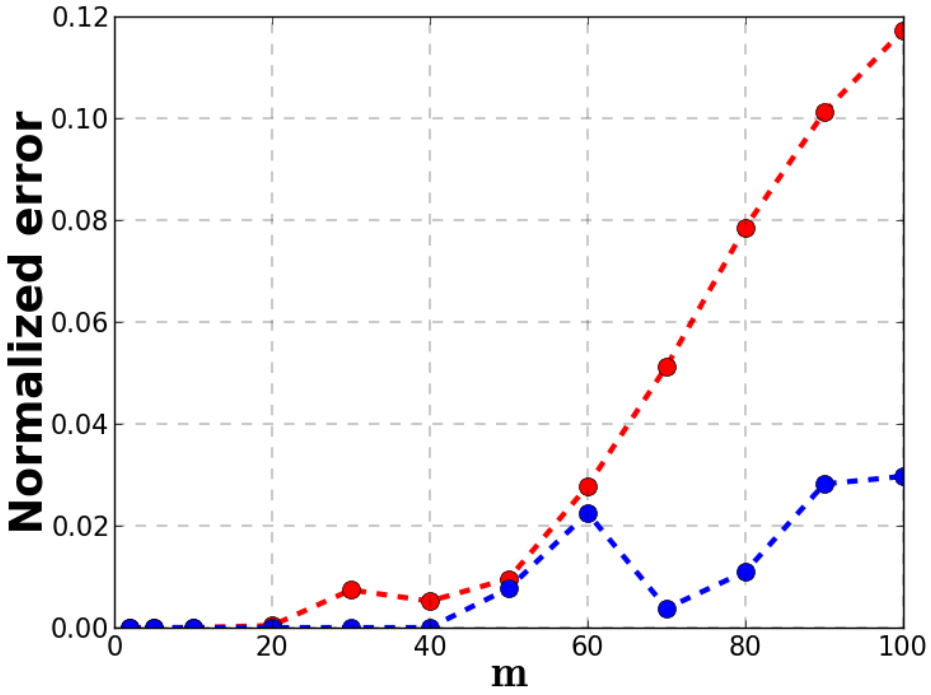
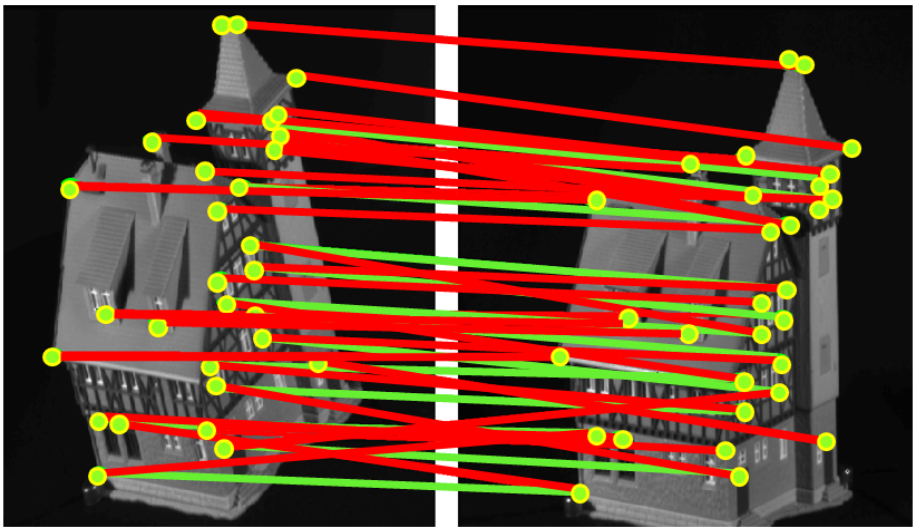
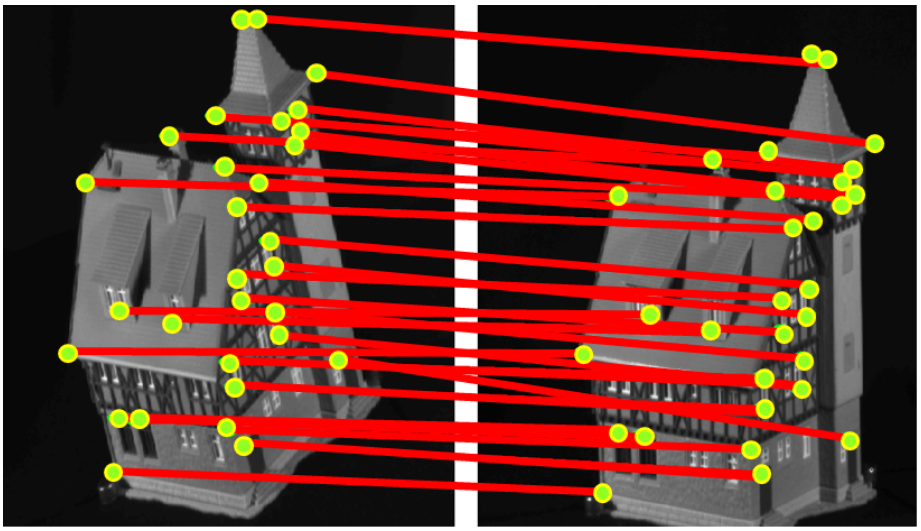


Figure 4.10: Normalized error as m increases on the House data set. Permutation Synchronization (blue) vs. the pairwise Kuhn-Munkres baseline (red).



(a)



(b)

Figure 4.11: (a-b) Matches found for a representative image pair. (Green circles) landmarks, (green lines) ground truth, (red lines) found matches. (a) Pairwise linear assignment, (b) Permutation Synchronization. Note that less visible green is good.

Repetitive Structures. Next, we considered a data set with severe geometric ambiguities due to repetitive structures. There is some consensus in the community that even sophisticated features (like SIFT) yield unsatisfactory results in this scenario, and deriving a good initial matching for structure from motion is problematic, see (Roberts et al., 2011) and references therein. Our evaluations included 16 images from the **Building** data set (Roberts et al., 2011), Figure 4.12. We identified 25 "similar looking"

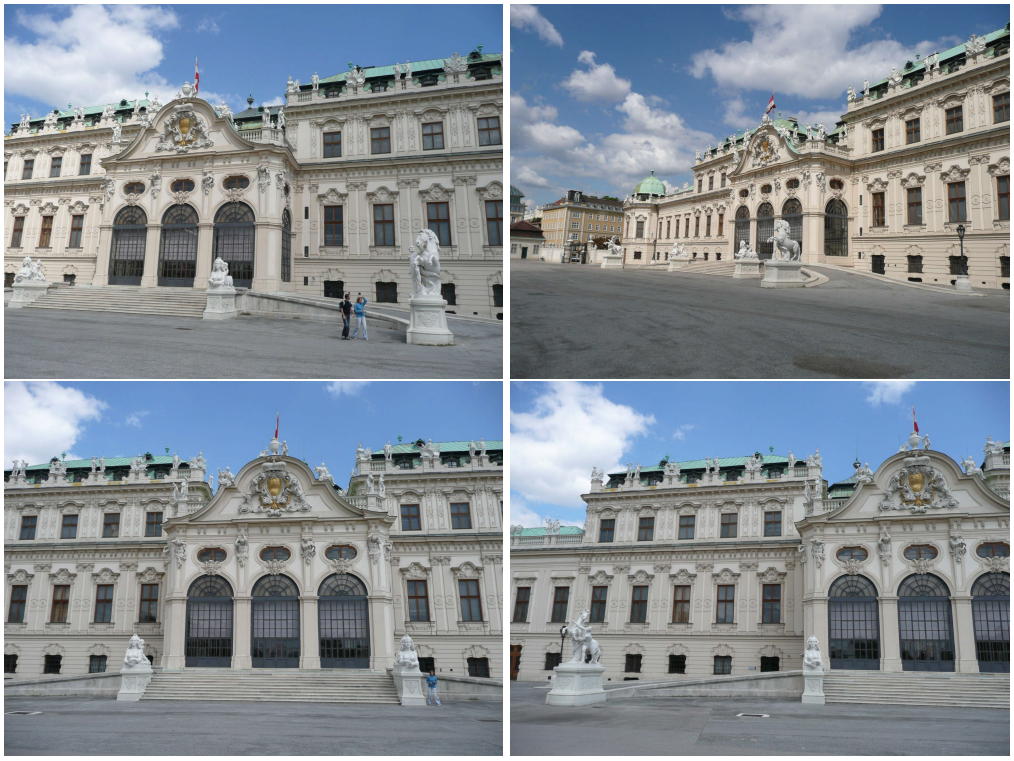
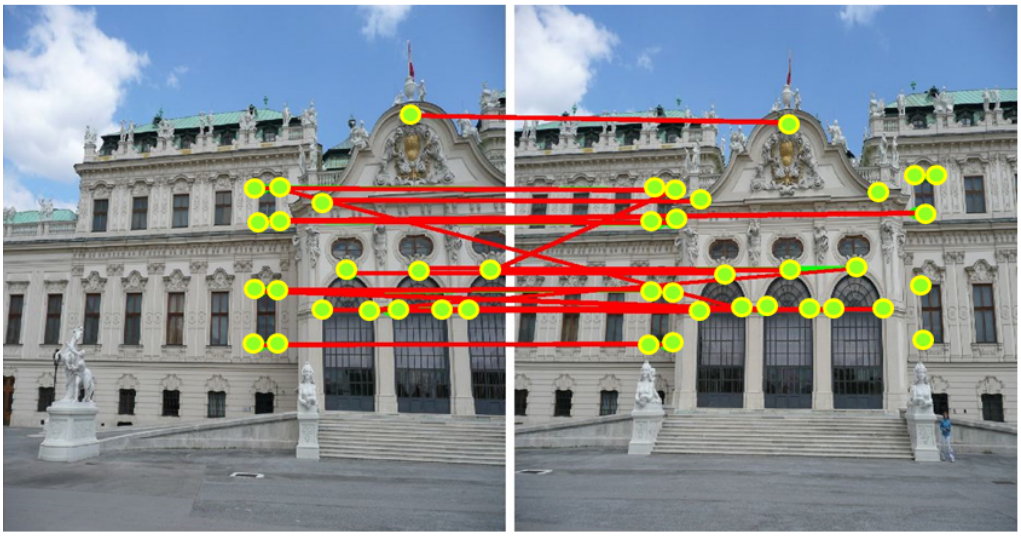


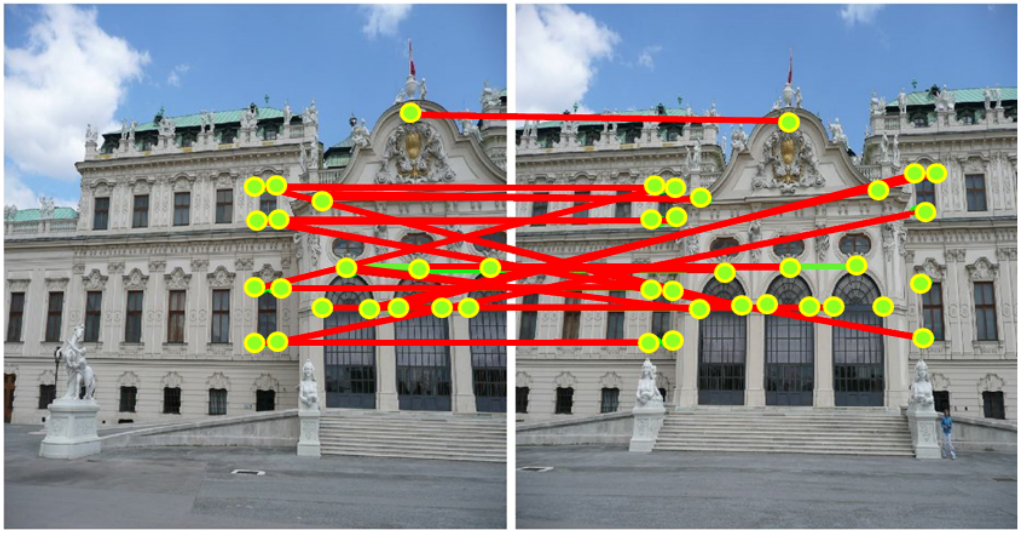
Figure 4.12: Subset of images from Building data set with multiple instances of similar structure.

landmark points in the scene, and hand annotated them across all images. Many landmarks were occluded due to the camera angle. Qualitative results for pairwise matching and Permutation Synchronization are shown in Figure 4.13. We highlight two important observations. First, our method

resolved geometrical ambiguities by enforcing mutual consistency efficiently. Second, Permutation Synchronization robustly handles occlusion: landmark points that are occluded in one image are seamlessly assigned to null nodes in the other (see the set of unassigned points in Figure 4.13(b)) thanks to evidence derived from the large number of additional images in the data set. In contrast, pairwise matching struggles with occlusion in the presence of similar looking landmarks (and feature descriptors). For $n = 25$ and $m = 16$, the error from the baseline method (Pairwise Linear Assignment) was 0.74. Permutation Synchronization decreased this by 10% to 0.64.



(a)



(b)

Figure 4.13: Matches for a representative image pair from Building data set. (Green circles) landmark points, (green lines) ground truth matchings, (red lines) found matches. (a) Pairwise linear assignment, (b) Permutation synchronization. Note that less visible green is better.

We made a similar observation on a second data set called **Books** (Roberts et al., 2011). This data set contains $m = 20$ images of multiple books on a "L" shaped study table, and suffers geometrical ambiguities similar to the above with severe occlusion Figure 4.14. Here we identified $n = 34$ landmark points, many of which were occluded in most images. The error from the baseline method was 0.92, and Permutation Synchronization decreased this by 22% to 0.70. Qualitative results are shown in Figure 4.15.

Figure 4.14: Multiple views of a "L" shaped study table from the Books data set.

(a)

(b)

Figure 4.15: Matches for a representative image pair from Books data set. (Green circles) landmark points, (green lines) ground truth matchings, (red lines) found matches. (a) Pairwise linear assignment, (b) Permutation synchronization. Note that less visible green is better.

4.3.2 Keypoint matching with nominal user supervision

Our final experiment deals with matching problems where keypoints in each image preserve a common structure. In the literature, this is usually tackled as a graph matching problem, with the keypoints defining the vertices, and their structural relationships being encoded by the edges of the graph. Ideally, one wants to solve the problem for all images at once but most practical solutions operate on image (or graph) pairs. Note that in terms of difficulty, this problem is quite distinct from the ones discussed above. In stereo, the *same object* is imaged and what varies from one view to the other is the field of view, scale, or pose. In contrast, in keypoint matching, the background is not controlled and even sophisticated descriptors may go wrong. Recent solutions often leverage supervision to make the problem tractable (Caetano et al., 2009; Leordeanu et al., 2009). Instead of learning parameters (Caetano et al., 2009; Jebara et al., 2009), we utilize supervision directly to provide the correct matches on a small subset of randomly picked image pairs (e.g., via a crowd-sourced platform like Mechanical Turk). We hope to exploit this 'ground-truth' to significantly boost accuracy via Permutation Synchronization. For our experiments, we used the baseline method output to set up our objective matrix $\mathcal{T}$ but with a fixed "supervision probability", we replaced the T_{ji} block by the correct permutation matrix, and ran Permutation Synchronization. We considered the "Bikes" sub-class from the **Caltech 256 data set**, which contains multiple images of common objects with varying backdrops, and chose to match images in the "touring bike" class, Figure 4.16.

Our analysis included 28 out of 110 images in this data set that were taken "side-on". SUSAN corner detector was used to identify landmarks in each image. Further, we identified 6 interest points in each image that correspond to the frame of the bicycle, Figure 4.17. We modeled the matching cost for an image pair as the shape distance between interest points in the pair.

Figure 4.16: Touring bike data set.

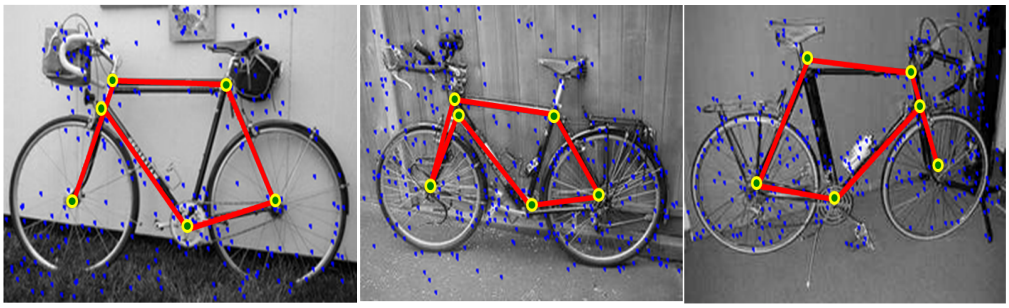


Figure 4.17: SUSAN detector output (blue). Bike frame (red).

For a fixed degree of supervision, we randomly selected image pairs for supervision and estimated matchings for the rest of the image pairs. We performed 50 runs for each degree of supervision. Mean error and standard deviation is shown in Figure 4.18 as supervision increases. Figure 4.19 demonstrates qualitative results of our method as the rate of supervision varies.

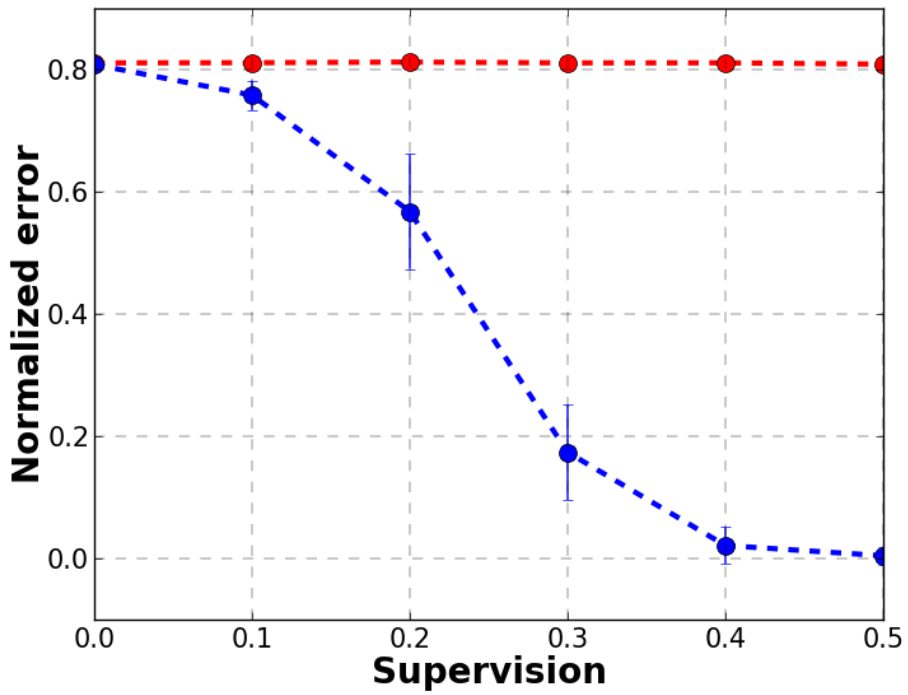
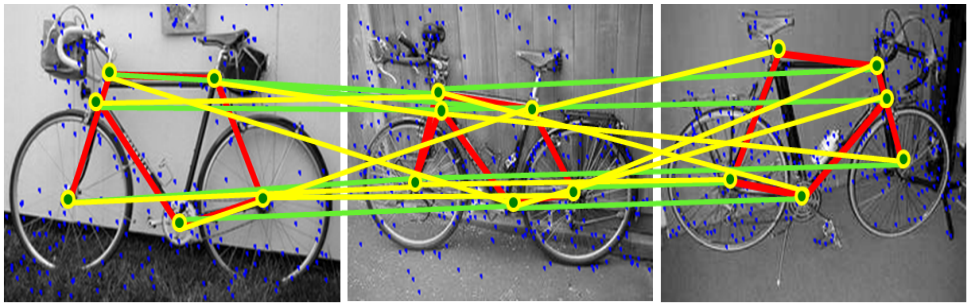
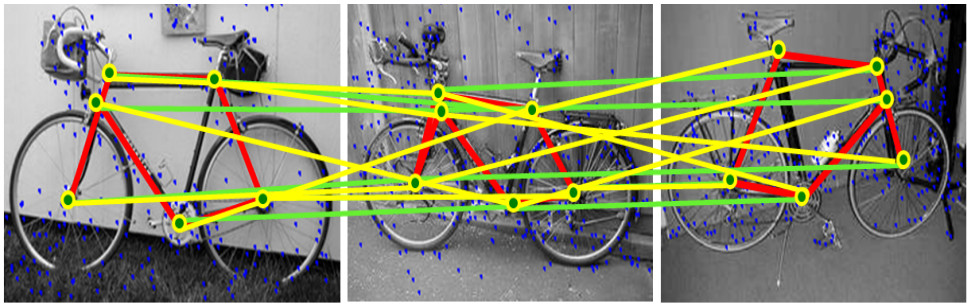


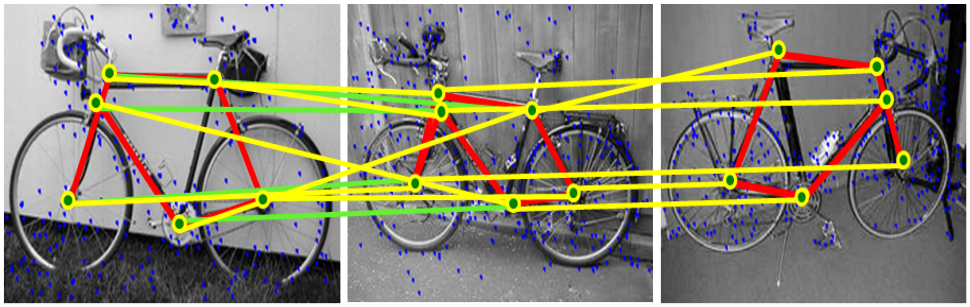
Figure 4.18: Normalized error as the degree of supervision varies. Baseline method PLA (red) and Permutation Synchronization (blue)



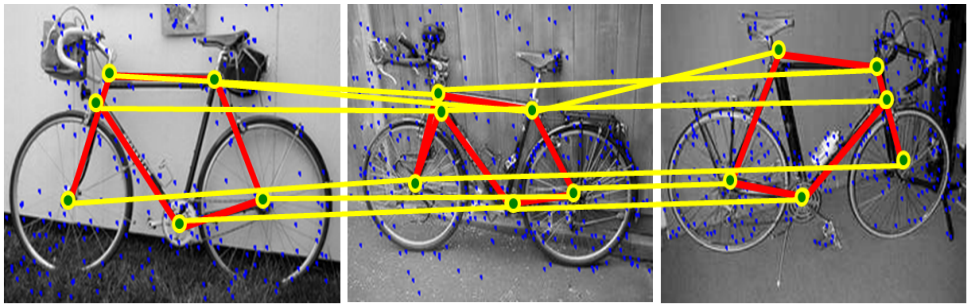
(a)



(b)



(c)



(d)

Figure 4.19: Recovered matches using permutation synchronization for a representative graph triplet. (Green circles) Landmark points, (Green lines) ground truth matchings, (Yellow lines) found matches.(a) Supervision = 0.1. (b) Supervision = 0.2. (c) Supervision = 0.3. (d) Supervision = 0.4.

4.4 Summary

Estimating the correct matching between two sets from noisy similarity data, such as the visual feature based similarity matrices that arise in computer vision is an error-prone process. However, when we have not just two, but m different sets, the consistency conditions between the $\binom{m}{2}$ pairwise matchings severely constrain the solution. In this chapter, we presented an eigenvector decomposition based algorithm, Permutation Synchronization which exploits this fact. The proposed algorithm is an *unsupervised* algorithm which pools information from all pairwise similarity matrices to jointly estimate a globally consistent array of matchings in a single shot. Moreover, we showed that the proposed framework is flexible to allow direct supervision effectively. Our theoretical results suggest that this approach is very robust to *random* noise. Experimental results confirm that in a range of computer vision tasks from stereo to keypoint matching, the method does indeed significantly improve performance (especially when m is large, as expected in video), and can get around problems such as occlusion that a pairwise strategy cannot handle.

Notice, our proposed synchronization method make certain assumptions – the sets presented for multi-matching are ordered i.e., every set must match every other set; the errors occurred in the putative matching procedure are random; the sets do not contain ambiguous objects and occlusion is structure-less. Our analysis showed that the multi-matching problem become far more obscure when any combination of these assumptions is grossly violated. For example, in vision applications, when there are several landmarks that look "alike" then the multi-way matching problem have appearance ambiguity. Further, large repetitive structure in the presence of severe occlusion manifest itself as coherent noise. Permutation synchronization method give sub-optimal results in these scenarios, indicating that several questions need to be investigated regarding the undordered multi-way matching problem in the presence of structured

noise. Motivated by these insights, we explore the benefits of a particularly elegant formalism, called graph connection Laplacian (Bandeira et al., 2013) in the next chapter, Chapter 5.

As described in the previous chapter, consistently matching interest points across images, is critical to various applications in computer vision. Typically, real-world vision tasks involve large datasets. One such application of interest is the large scale Structure from Motion (SfM) which utilizes images from the internet to reconstruct 3D scenes. Images harvested from the internet data are largely unordered, i.e., we have no prior knowledge how the images are related to each other. Keypoints or features extracted from each image provide correspondences between pairs of images, making it possible to estimate the relative camera pose. This gives rise to an association graph in which two images are connected by an edge if they share a sufficient number of corresponding keypoints, and the edge itself is labeled by the estimated matching between the two sets of keypoints. Starting with these putative image to image associations, one typically uses the bundle adjustment procedures (**?**) to simultaneously solve for the global camera pose parameters and 3D scene locations.

Despite their popularity, large scale bundle adjustment methods have well known limitations. In particular, due to the highly nonlinear nature of the objective function, they can get stuck in bad local minima. Therefore, starting with a good initial matching (i.e., an informative image association graph) is critical. Several papers have studied this behavior in detail (Crandall et al., 2011), and conclude that if one starts the numerical optimization from an incorrect "seed" (i.e., a subgraph of the image associations), the downstream optimization is unlikely to ever recover. Several authors (Wilson and Snavely, 2013; Roberts et al., 2011; Jiang et al., 2012) have recently described situations in large scale structure from motion where setting up a good image association graph is difficult, and consequently a direct application of bundle adjustment yields unsatisfactory results. One such situation is when the scene depicted in the images involves a

(a)

(b)

Figure 5.1: HOUSE sequence. (a) Representative images. (b) Folded reconstruction by traditional SfM pipeline (Wu, 2013; Wu et al., 2011).

large number of duplicate structures, see Figure 5.1. The preprocessing step in a standard pipeline will match visual features and set up the associations accordingly, but a key underlying assumption in most (if not all) approaches is that we observe only a single instance of any structure.

This assumption is problematic when scenes have repeating architectural components or recurring patterns, such as windows, bricks, and so on. In Figure 5.1(a), views that look exactly the same do not necessarily represent the same physical structure. Some (or all) points in one image are actually occluded in the other image. Typical SfM methods will not work well when initialized with such image associations, regardless of which type of bundle adjustment solver we use. In our example, the resulting reconstruction will be folded, see Figure 5.1(b). In other cases (Wilson and Snavely, 2013), we get errors ranging from phantom walls to severely superimposed structures yielding nonsensical reconstructions.

Similar challenges arise in other fields, ranging from machine learning (Nguyen et al., 2011) to computational biology. For instance, consider the *de novo* genome assembly problem in computational biology (Li et al., 2010). The goal here is to reconstruct the original DNA sequence from fragments without a reference genome. Because the genome may have many repeated structures, the alignment problem becomes very hard. In general, reconstruction algorithms start with two maximally overlapping sequences, and proceed by selecting subsequent fragment using a process not unlike bundle adjustment. This process is prone to similar issues with local minima (Pop et al., 2002). In both cases it would be preferable to have a model that reasons globally over all pairwise information. In this chapter, to make our presentation as concrete as possible, we restrict ourselves to describing such an algorithm in the context of Structure from Motion, while understanding that the underlying ideas apply more generally.

Related Work. The issue described above is variously known in the literature as the SfM disambiguation problem or the data/image association problem in structure from motion. Some of the strategies that have been proposed to mitigate it impose additional conditions, such as in (Schaffalitzky and Zisserman, 2002; Snavely et al., 2006; Martinec and Pajdla, 2007;

Havlena et al., 2009; Sinha et al., 2012), but this also breaks down in the presence of large coherent sets of incorrectly matched pairs. One creative solution in recent work is to use meta-data alongside images. "Geotags" or GIS data when available have been shown to be very effective in deriving a better initialization for bundle adjustment or as a post-processing step to stitch together different components of a reconstruction. In (Roberts et al., 2011), the authors suggest using image time-stamps to impose a natural association among images, which is valuable when the images are acquired by a single camera in a temporal sequence but difficult to deploy otherwise. Separate from the meta-data approach, in controlled scenes with relatively less occlusion, missing correspondences yield important local cues to infer potentially incorrect image pairs (Roberts et al., 2011; Jiang et al., 2012). Very recently, (Wilson and Snavely, 2013) formalized the intuition that incorrect feature correspondences result in anomalous structures in the so-called visibility graph of the features. By looking at a measure of local track quality (from local clustering), one can reason about which associations are likely to be erroneous. This works well when the number of points is very large, but the authors of (Wilson and Snavely, 2013) acknowledge that for data-sets like those shown in Figure 5.1, it may not help much.

In contrast to the above approaches, a number of recent algorithms for the association (or disambiguation) problem argue for *global* geometric reasoning. In (Enqvist et al., 2011), the authors used the number of point correspondences as a measure of certainty, which was then globally optimized to find a maximum-weight set of consistent pairwise associations. The authors in (Zach et al., 2008) seek consistency of epipolar geometry constraints for triplets, whereas (Zach et al., 2010) expands it over larger consistent cliques. The procedure in (Enqvist et al., 2011) takes into account loops of associations concurrently with a minimal spanning tree over image to image matches. In summary, the bulk of prior work

suggests that locally based statistics over chained transformations will run into problems if the inconsistencies are more global in nature. However, even if the objectives used are global, *approximate* inference is not known to be robust to *coherent* noise, which is exactly what we face in the presence of duplicate structures (Govindu, 2006).

If we take the idea of reasoning globally about association consistency using triples or higher order loops to an extreme, it implies deriving the likelihood of a specific image to image association conditioned on *all* other associations. This joint likelihood does not factor and explicit enumeration quickly becomes intractable. Here, we propose a new approach which operates on the association graph derived from image pairs but with a key distinguishing feature. The association relationships will now be denoted in terms of a 'certificate', that is, the *transformation* which justifies the relationship. The transformation may denote the pose parameters derived from the correspondences or the matching (between features) itself, see Figure 5.2. Other options are possible — as long as this transformation is a *group action* from one set to the other. If so, we can carry over the intuition of consistency over larger cliques of images desired in existing works and rewrite those ideas as invariance properties of functions defined on the *group*. In particular, when the transformation is a matching, each edge in the graph is a permutation, i.e., a member of the symmetric group $\mathbb{S}_n$, and a generalization of the Laplacian related to the representation theory of $\mathbb{S}_n$ encodes the associations. In this regard, the proposed approach is based on the recent work on synchronization by (Singer and Shkolnisky, 2011; Singer and Wu, 2012; Pachauri et al., 2013; Huang and Guibas, 2013).

5.1 Synchronization

Consider a collection of m images $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_m\}$ of the same object or scene taken from different viewpoints and possibly under different condi-

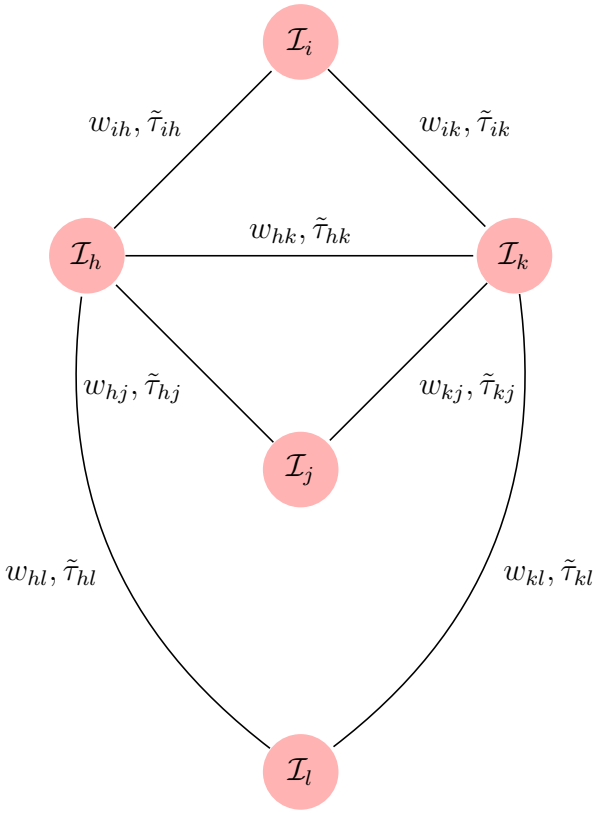


Figure 5.2: An association graph for images with scalar weights w_{ij} and putative matchings τ_{ij} along edges.

tions, and assume that in each image $\mathcal{I}_i$, a keypoint detector has detected n landmarks (keypoints) $\{x_1^i, x_2^i, \ldots, x_n^i\}$. Given two images $\mathcal{I}_i$ and $\mathcal{I}_j$, the landmark matching problem consists of finding pairs of landmarks $x_p^i \sim x_q^j$ (with x_p^i coming from image $\mathcal{I}_i$ and x_q^j coming from $\mathcal{I}_j$ which correspond to the same underlying physical feature). The matching between $\mathcal{I}_i$ and $\mathcal{I}_j$ may be described by the unique permutation $\tau_{ji} \colon \{1, 2, \ldots, n\} \longrightarrow \{1, 2, \ldots, n\}$ under which $x_p^i \sim x_{\tau_{ji}(p)}^j$. Typically, local image features, such as SIFT descriptors, can provide an initial guess for each τ_{ji}, but by itself each of these individual image-to-image matchings is highly error prone. A major clue to correcting these errors is the constraint that matchings must be

consistent, i.e., $\tau_{kj}\tau_{ji} = \tau_{ki}$. In the previous chapter, Chapter 4, we used this clue and described an algorithm which finds all the matchings jointly, in one shot, via a relaxation to eigenvector decomposition. In particular, the fact that the product of two permutations σ_1 and σ_2 in the usual way as

$$\sigma_3 = \sigma_2\sigma_1 \qquad \Longleftrightarrow \qquad \sigma_3(i) = \sigma_2(\sigma_1(i)) \qquad i = 1, 2, \ldots, n,$$

the $n!$ different permutations of $\{1, 2, \ldots, n\}$ form a group, reduced the problem to finding a consistent set of τ_{ji}'s to finding the m base permutations $\sigma_1, \ldots, \sigma_m$.

Problems of this general form, where given some (finite or continuous) group G, one must estimate a matrix $(g_{ji})_{j,i=1}^{m}$ of group elements obeying $g_{kj}g_{ji} = g_{ki}$ are called synchronization problems, (Singer and Shkolnisky, 2011; Singer, 2011; Pachauri et al., 2013; Huang and Guibas, 2013; Ozyesil et al., 2015; Chen et al., 2014). In the context of synchronizing three dimensional rotations for cryo-EM, Singer and Wu (Singer and Wu, 2012) proposed an elegant formalism, called Vector Diffusion Maps, which formulates synchronization as diffusing the base rotation Q_i from each image to its neighbors. Reconsider the graph shown in Figure 5.2 but with a distinction that the transformations on the edges are rotations, i.e., attribute of the edge between $\mathcal{I}_i$ and $\mathcal{I}_j$ is O_{ji}. The diffusion process diffuses the base rotation Q_i to $\mathcal{I}_j$ and the observed O_{ji} relative rotation of $\mathcal{I}_j$ to $\mathcal{I}_i$ changes Q_i to $O_{ji}Q_i$. If all the $(O_{ji})_{i,j}$ observations were perfectly synchronized, then no matter what path $i \longrightarrow i_1 \longrightarrow i_2 \longrightarrow \ldots \longrightarrow j$ we took from i to j, the resulting rotation $O_{j,i_p} \ldots O_{i_2,i_1} O_{i_1,i} Q_i$ would be the same. The paths

between $\mathcal{I}_i$ and $\mathcal{I}_j$ in Figure 5.2 are

$$
\begin{aligned}
i &\longrightarrow h \longrightarrow j \\
i &\longrightarrow k \longrightarrow j \\
i &\longrightarrow h \longrightarrow k \longrightarrow j \\
i &\longrightarrow k \longrightarrow h \longrightarrow j \\
i &\longrightarrow h \longrightarrow k \longrightarrow j \\
i &\longrightarrow h \longrightarrow l \longrightarrow k \longrightarrow j \\
i &\longrightarrow k \longrightarrow l \longrightarrow h \longrightarrow j
\end{aligned}
$$

.

If all the observations are perfect then we have

$$
\begin{aligned}
O_{j,h}\, O_{h,i}\, Q_i &= O_{j,k}\, O_{k,i}\, Q_i \\
&= O_{j,h}\, O_{h,k}\, O_{k,i} Q_i \\
&= O_{j,k}\, O_{k,h}\, O_{h,i} Q_i \\
&= O_{j,k}\, O_{k,l}\, O_{l,h}\, O_{h,i} Q_i \\
&= O_{j,k}\, O_{h,l}\, O_{l,k}\, O_{k,i} Q_i \ .
\end{aligned}
$$

However, if some (in many practical cases, the majority) of the O_{ji}'s are incorrect, then different paths from one vertex to another contribute different rotations, which one then needs to average in some appropriate sense.

A natural choice for the loss that describes the extent to which the $Q_1, \ldots, Q_m$ imputed base rotations (playing the role of the σ_i's in the permutation case) satisfy the O_{ji} observations is

$$
\mathcal{E}(Q_1, \ldots, Q_m) = \frac{1}{2} \sum_{i,j=1}^{m} w_{ij} \| Q_j - O_{ji} Q_i \|_{\text{Frob}}^2 = \frac{1}{2} \sum_{i,j=1}^{m} w_{ij} \| Q_j Q_i^\top - O_{ji} \|_{\text{Frob}}^2, \tag{5.1}
$$

where the w_{ij} edge weight describes our confidence in rotation O_{ji}. Note that this loss function is more robust to the noisy edges in the graph as compared to the synchronization loss proposed in Chapter 4 because the diffusion process on a graph reveals consistent cluster properties of the graph.

A crucial observation is that loss in (5.1) can be rewritten in the form $\mathcal{E}(Q_1, \ldots, Q_m) = V^\top \mathcal{L} V$, where

$$V = \begin{pmatrix} Q_1 \\ \vdots \\ Q_m \end{pmatrix}, \qquad \mathcal{L} = \begin{pmatrix} d_i I & -w_{1,2} O_{1,2} & \ldots & -w_{1,m} O_{1,m} \\ \vdots & \ddots & & \vdots \\ -w_{m,1} O_{m,1} & -w_{m,2} O_{m,2} & \ldots & d_m I \end{pmatrix}, \tag{5.2}$$

and $d_i = \sum_{j \neq i} w_{ij}$. Note that since $w_{ij} = w_{ji}$, and $O_{ij} = O_{ji}^{-1} = O_{ji}^\top$, the matrix $\mathcal{L}$ is symmetric. Furthermore, the above is exactly analogous to the way in which in spectral graph theory, (Chung, 1997), the functional $\mathcal{E}(f) = \frac{1}{2} \sum_{i,j} w_{ij} (f(i) - f(j))^2$ describing the "smoothness" (with respect to the graph topology) of a function f defined on the vertices of a graph can be written as $f^\top L f$ in terms of the usual graph Laplacian

$$L_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{k \neq i} w_{ik} & i = j. \end{cases}$$

As it is well known, constraining f to have unit norm and excluding the subspace of constant functions, the function minimizing $\mathcal{E}(f)$ is the eigenvector of L with (second) smallest eigenvalue. Analogously, in synchronizing rotations, the steady state of the diffusion system, which minimizes (5.1), can be computed by forming the $3m \times 3$ dimensional matrix V from the 3 lowest eigenvalue eigenvectors of $\mathcal{L}$, and appropriately rounding each 3×3 block V_i of V to the nearest orthogonal matrix Q_i. The resulting array $(Q_j Q_i^\top)_{i,j}$ of imputed relative rotations is guaranteed to be consistent, and minimizes the loss (5.1).

5.2 Permutation Diffusion

Its elegance notwithstanding, the vector diffusion formalism of the previous section seems ill suited to our present purposes of improving the SfM pipeline for two reasons: (1) synchronizing over $\mathbb{S}_n$, which is a finite group, seems much harder than synchronizing over the continuous group of rotations; (2) rather than getting an actual synchronized array of matchings, what is critical to SfM is to estimate the association graph that captures the extent to which any two images are related to one-another. In this chapter, we show that both of these problems have natural solutions in the formalism of group representations.

Our first key observation (already alluded to in (Singer, 2011)) is that the critical step of rewriting the loss Equation (5.1) in terms of the Laplacian Equation (5.2) does not depend on any special properties of the rotation group other than the facts that (a) rotation matrices are unitary (in fact, orthogonal) (b) if we follow one rotation by another, their matrices simply multiply. In general, for any group G, a complex valued function $\rho\colon G \longrightarrow \mathbb{C}^{d_\rho \times d_\rho}$ which satisfies $\rho(g_2\, g_1) = \rho(g_2)\, \rho(g_1)$ is called a representation of G. The representation is unitary if $\rho(g^{-1}) = (\rho(g))^{-1} = \rho^\dagger$, where $M^\dagger$ denotes the Hermitian conjugate (conjugate transpose) of M. Thus, we have the following proposition.

Proposition 5.1. *Let G be any compact group with identity e and $\rho\colon G \longrightarrow \mathbb{C}^{d_\rho \times d_\rho}$ be a unitary representation of G. Then given an array of possibly noisy and unsynchronized group elements, $(g_{ji})_{i,j}$, and corresponding positive confidence weights $(w_{ji})_{i,j}$, the synchronization loss (assuming $g_{ii} = e$ for all i)*

$$\mathcal{E}(h_1, \ldots, h_m) = \frac{1}{2} \sum_{i,j=1}^{m} w_{ji} \left\| \rho(h_j h_i^{-1}) - \rho(g_{ji}) \right\|_{Frob}^2 \qquad h_1, \ldots, h_m \in G$$

can be written in the form $\mathcal{E}(h_1, \ldots, h_m) = V^\dagger \mathcal{L} V$*, where*

$$V = \begin{pmatrix} \rho(h_1) \\ \vdots \\ \rho(h_m) \end{pmatrix}, \qquad \mathcal{L} = \begin{pmatrix} d_i\, I & -w_{1,2}\, \rho(g_{1,2}) & \ldots & -w_{1,m}\, \rho(g_{1,m}) \\ \vdots & \ddots & & \vdots \\ -w_{m,1}\, \rho(g_{m,1}) & -w_{m,2}\, \rho(g_{m,2}) & \ldots & d_m\, I \end{pmatrix}. \tag{5.3}$$

To synchronize matchings between images using this proposition, one plugs in the appropriate unitary representation of the symmetric group. Similar to Chapter 4, we use permutation matrix representation

$$\rho_{\text{def}}(\sigma) = P(\sigma) \qquad [P(\sigma)]_{q,p} = \begin{cases} 1 & \sigma(p) = q \\ 0 & \text{otherwise,} \end{cases}$$

to express the loss function in (5.3) as

$$\mathcal{E}(\sigma_1, \ldots, \sigma_m) = \frac{1}{2} \sum_{i,j=1}^{m} w_{ji}\, \| P(\sigma_j \sigma_i^{-1}) - P(\tau_{ji}) \|_{\text{Frob}}^2. \tag{5.4}$$

The squared Frobenius norm in this expression simply counts the number of mismatches between the observed but noisy permutation τ_{ji}, and the inferred permutation $\sigma_j \sigma_i^{-1}$. For this choice of ρ, letting $P_i := P(\sigma(i))$ and $P_{ji}^{\text{obs}} := P(\tau_{ji})$, $\mathcal{E}(\sigma_1, \ldots, \sigma_m) = V^\top \mathcal{L} V$, with

$$V = \begin{pmatrix} P_1 \\ \vdots \\ P_m \end{pmatrix}, \qquad \mathcal{L} = \begin{pmatrix} d_i\, I & -w_{1,2}\, P_{1,2}^{\text{obs}} & \ldots & -w_{1,m}\, P_{1,m}^{\text{obs}} \\ \vdots & \ddots & & \vdots \\ -w_{m,1}\, P_{m,1}^{\text{obs}} & -w_{m,2}\, P_{m,2}^{\text{obs}} & \ldots & d_m\, I \end{pmatrix}. \tag{5.5}$$

Consequently, just as in the rotation case, synchronization over $\mathbb{S}_n$ can be solved by forming V from the first $d_{\rho_{\text{def}}} = n$ lowest eigenvectors of $\mathcal{L}$, and extracting each P_i from its i'th $n \times n$ block, V_i. Here we must take a little care because unless the τ_{ji}'s are already synchronized, it is not a priori

guaranteed that the resulting block will be a valid permutation matrix. Therefore, analogously to the procedure described in Chapter 4, we first multiply V_i by $V_1^\top$, and then use a linear assignment procedure to find the permutation $\widehat{\sigma}_i$, whose permutation matrix is closest to $V_i V_1^\top$. The resulting algorithm we call Synchronization by Permutation Diffusion.

5.3 Uncertain Matches and Permutation Diffusion Affinity

The limitation of our framework, as described so far, is the assumption that each keypoint in each image will have a single counterpart in every other image that the local matching procedure with some error can identify. In realistic scenarios this is far from satisfied, due to occlusion, repetitive structures, and noisy detections. Most algorithms, including (Huang and Guibas, 2013) and (Pachauri et al., 2013), deal with the problem simply by turning the P_{ij} block in Equation (5.5) into a weighted sum of all possible permutations. For example, if landmarks number $1 \ldots 20$ are present in both images, but landmarks $21 \ldots 40$ are not, then the P_{ij} block in Equation (5.5) will have a corresponding 20×20 block of all ones, rescaled by a factor of $1/20$.

This approach effectively amounts to replacing τ_{ji} by an appropriate *distribution* $t_{ji}(\tau)$ over matchings. Correspondingly, when we form V from the first d_ρ eigenvectors of $\mathcal{L}$, each resulting V_i block will stand for a distribution $p_i(\sigma)$, rather than a single base permutation σ_i. Moreover, if some set of k landmarks $U = (u_1, \ldots, u_k)$ are occluded in $\mathcal{I}_i$, then t_{ij} (for any j) will be agnostic to their assignment, and consequently p_i will be invariant to what is mapped to $u_1, \ldots, u_k$. Let $\sigma \sim_U \sigma'$ denote the relation that two permutations σ and σ' differ *only* in what numbers they map to $u_1, \ldots, u_k$, but fully agree on what they assign to any landmark *not* in U (i.e., $\sigma(i) = \sigma(j) \ \ \forall\, i \notin U$). Clearly, $\sim_U$ is an equivalence relation on $\mathbb{S}_n$, and

it is not difficult to see that letting μ_U be some reference permutation that maps $1 \mapsto u_1, \ldots, k \mapsto u_k$, and $\mathbb{S}_k$ be the subgroup of permutations that permute $1, 2, \ldots, k$ amongst themselves but leave $k+1, \ldots, n$ fixed, the equivalence classes of $\sim_U$ are the sets

$$\mu_U \mathbb{S}_k \nu := \{\mu_U \gamma \nu \mid \gamma \in \mathbb{S}_k\} \qquad \nu \in \mathbb{S}_n. \tag{5.6}$$

These sets are called (two-sided) $\mathbb{S}_k$–cosets. Note that while $|\mathbb{S}_n| = n!$, there are only $n!/k!$ distinct equivalence classes, so not all possible values of ν yield a distinct coset.

What is important is that uncertainty in the synchronization process with respect to a given set of landmarks $\{u_1, \ldots, u_k\}$ (typically due to occlusion) has a clear algebraic signature, namely the inferred p_i being constant on each of the cosets in Equation (5.6). Conversely, if we find that p_i is constant on these cosets, that is a strong indication that $u_1, \ldots, u_k$ are occluded, which is an important clue to estimating $\mathcal{I}_i$'s viewpoint, sometimes even more informative than the synchronized matchings themselves.

The invariance structure of p_i is most easily detected from its so-called auto-correlation function

$$a_i(\sigma) = \sum_{\omega \in \mathbb{S}_n} p_i(\sigma\omega)\, p_i(\omega). \tag{5.7}$$

Clearly, Equation (5.7) attains its maximum at the identity permutation, where $a_i(e) = \sum_{\omega \in \mathbb{S}_n} p_i(\omega)^2$. However, when p_i has invariances, the same maximum will be attained over a wider plateau of permutations. Note, in particular, that ω and $\sigma\omega$ always fall in the same $\mu_U \mathbb{S}_k \nu$ coset when $\sigma \in \mu_U \mathbb{S}_k \mu_U^{-1}$. Therefore, if p_i happens to be a function that is constant on $\mu_U \mathbb{S}_k \nu$ cosets, then any $\sigma \in \mu_U \mathbb{S}_k \mu_U^{-1}$ will maximize $a_i(\sigma)$.

Of course, in synchronization problems p_i is not directly accessible to us, rather we only have access to the weighted sum $\widehat{p}_i(\rho) := \sum_{\sigma \in \mathbb{S}_n} p_i(\sigma)\, \rho(\sigma) = V_i V_1^\top$. We observe that the form of $\widehat{p}_i(\rho)$ is *similar* to the Fourier expan-

sion of the distribution $p_i(\sigma)$ in terms irreducible reprentation of $\mathbb{S}_n$, as described in Chapter 2. While $\widehat{p}_i(\rho)$ is not exactly a Fourier component of p_i, it can be expressed as one by rewriting ρ as a direct sum of smaller irreducible representations. This means,

$$\widehat{p}_i(\rho) = C^\dagger \Big[\bigoplus_{\lambda \in \Lambda} \widehat{p}_i(\lambda) \Big] C$$

for some unitary matrix C. This is effectively just a basis transform as we described in Chapter 2. One of the properties of the Fourier transform is that if h is the cross-correlation of two functions f and g (i.e., $h(\sigma) = \sum_{\mu \in \mathbb{S}_n} f(\sigma\mu)\, g(\mu)$), then $\widehat{h}(\lambda) = \widehat{f}(\lambda)\, \widehat{g}(\lambda)^\dagger$. Consequently, assuming that V_1 has been normalized to ensure that $V_1^\top V_1 = I$, and using the fact that in our setting all matrices are real,

$$\widehat{a}_i(\rho) := C^\dagger \Big[\bigoplus_{\lambda \in \Lambda} \widehat{a}_i(\lambda) \Big] C = C^\dagger \Big[\bigoplus_{\lambda \in \Lambda} \widehat{p}_i(\lambda)\, \widehat{p}_i(\lambda)^\dagger \Big] C = (V_i V_1^\top)\, (V_i V_1^\top)^\top = V_i V_i^\top$$

is an easily computable matrix that captures essentially all the coset invariance structure encoded in the inferred distribution p_i.

To compute an affinity score between two images $\mathcal{I}_i$ and $\mathcal{I}_j$ reflecting how many occluded landmarks they share, it remains to compare their coset invariance structures, for example, by computing $(\sum_{\sigma \in \mathbb{S}_n} a_i(\sigma)\, a_j(\sigma))^{1/2}$. Again using the correlation theorem, one finds that this reduces to

$$\Pi(i,j) = \operatorname{tr}(V_i V_i^\top V_j V_j^\top)^{1/2}, \tag{5.8}$$

which we call Permutation Diffusion Affinity (PDA). Remarkably, PDA is closely related to the notion of diffusion similarity derived in (Singer and Wu, 2012) for rotations, using entirely different, differential geometric tools. Our experiments show that PDA is surprisingly informative about the actual distance between image viewpoints in physical space, and, as easy it is to compute, can greatly improve the performance of the SfM

pipeline.

5.4 Experiments

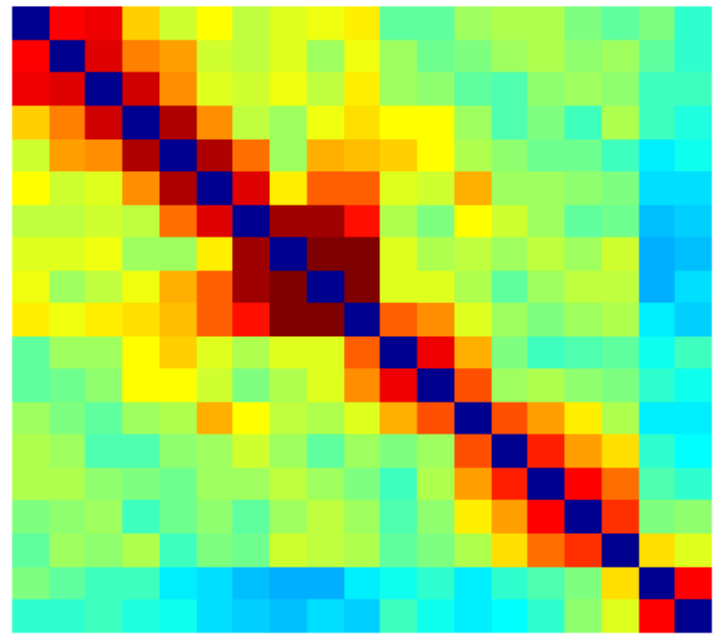
As described in Chapter 4, we want to match interest points across multiple images but here, the image data set is unordered and contains large repetitive structure. So the first logical step is to organize the images based on consistent associations between them. We used the proposed Permutation Diffusion Maps to extract image association matrices for various data sets described in the literature. Geometric ambiguities due to large duplicate structures are evident in each of these data sets, in up to 50% of the matches (Roberts et al., 2011; Zhu et al., 2010). As we will show shortly, even sophisticated SfM pipelines run into difficulties for such data sets and our machinery in Chapter 4 only solves this partially because of the underlying single instance assumption. While our primary interest is SfM, to illustrate the utility and applicability of PDM, we also present experimental results for scene summarization for a set of images (Simon et al., 2007; Aner and Kender, 2002).

5.4.1 Structure from Motion (SfM)

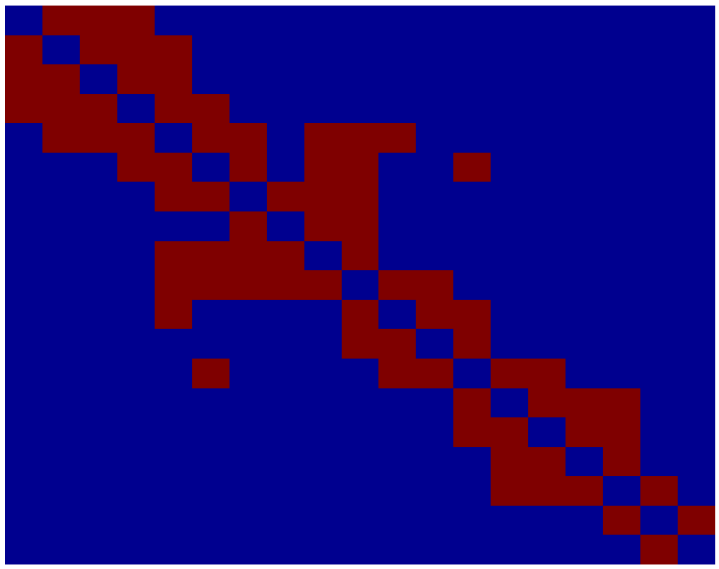
In the SfM experiments we used PDM to generate an image match matrix which is then fed to a state-of-the-art SfM pipeline for 3D reconstruction (Wu, 2013; Wu et al., 2011). The baseline was a Bundle Adjustment procedure which uses visual features for matching and has a built-in heuristic outlier removal module. Several other papers have used a similar comparisons (Roberts et al., 2011). For each data set, SIFT was used to detect and characterize landmarks (Lowe, 2004; Mikolajczyk and Schmid, 2004). We compute putative pairwise matchings $(\tau_{ij})_{i,j=1}^{m}$ by solving $\binom{m}{2}$ linear independent assignments (Kuhn, 1955) based on their SIFT features. The permutation matrix representation is used for putative matchings

$(\tau_{ij})_{i,j=1}^{m}$ as in (5.5). Here, n is relative large, on the order of 1000. Ideally, n is the total number of distinct keypoints in the 3D scene, but is not directly observable, so we set n to be the maximum number of keypoints detected in any single image in the data set. Eigenvector based procedure computes weighted affinity matrix. We used a binary match matrix as the input to an SfM library (Wu, 2013; Wu et al., 2011). Note that we only provide this library the image association hypotheses, leaving all other modules unchanged. With (potentially) good image association information, the SfM modules can sample landmarks more densely and perform bundle adjustment, leaving everything else unchanged. The baseline 3D reconstruction is performed using the same SfM pipeline without intervention.

The "HOUSE" sequence has three instances of similar looking houses Figure 5.1. The diffusion process accumulates evidence and eventually provides strongly connected images in the data association matrix Figure 5.3(a). Warm colors correspond to high affinity between pairs of images. The binary match matrix was obtained by applying a threshold on the weighted matrix Figure 5.3(b). We used this matrix to define the image matching for feature tracks. This means that features are *only* matched between images that are connected in our matching matrix. The SfM pipeline was given these image matches as a hypotheses to explain how the images are "connected". The resulting reconstruction correctly gives three houses Figure 5.4. In contrast, the same SfM pipeline when allowed to track features automatically with an outlier removal heuristic resulted in a folded reconstruction Figure 5.1(b). One may ask if more specialized heuristics will do better, such as time stamps, as suggested in (Roberts et al., 2011). However, experimental results in (Wilson and Snavely, 2013) and elsewhere strongly suggests that these data sets still remain challenging.



(a)



(b)

Figure 5.3: House sequence. (a) Weighted image association matrix. (b) Binary image match matrix.

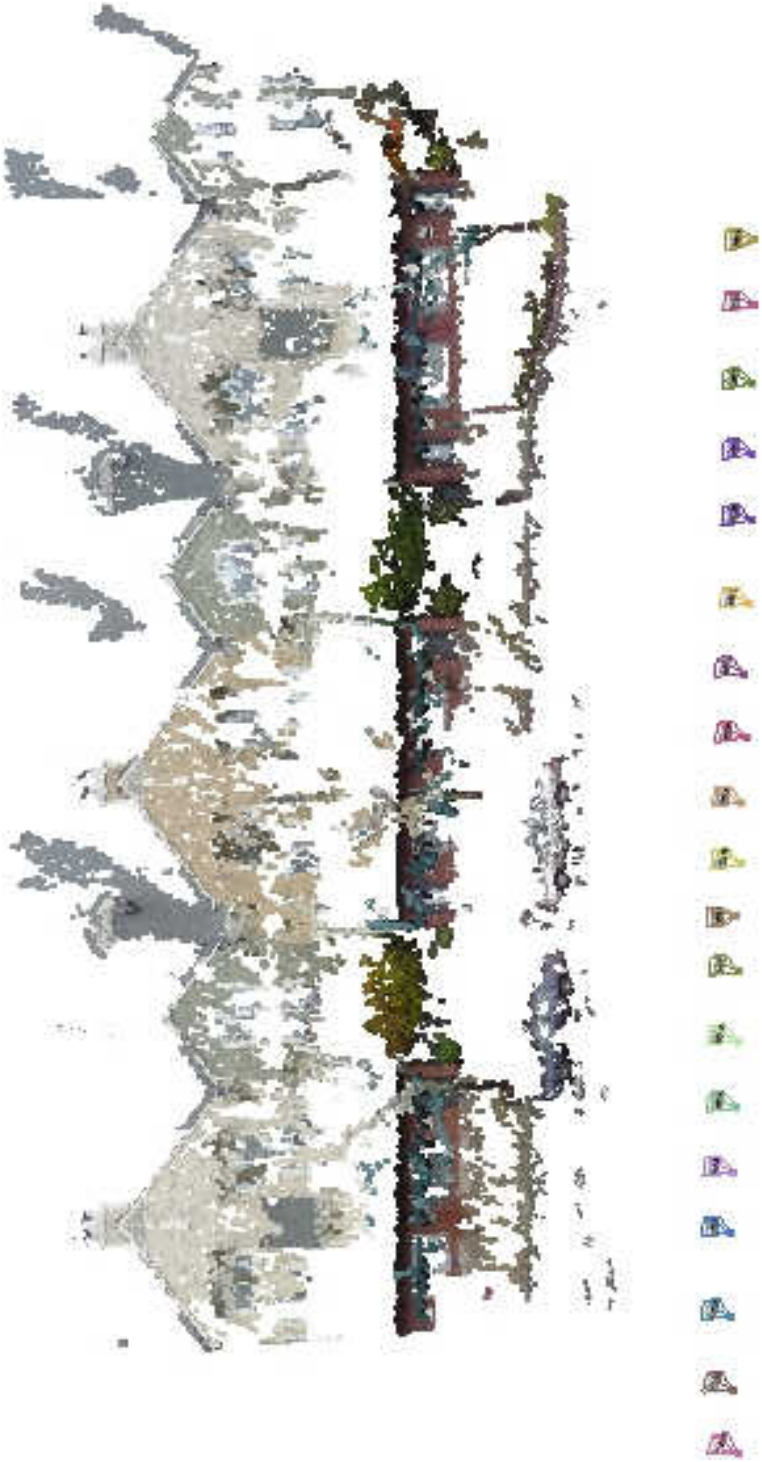
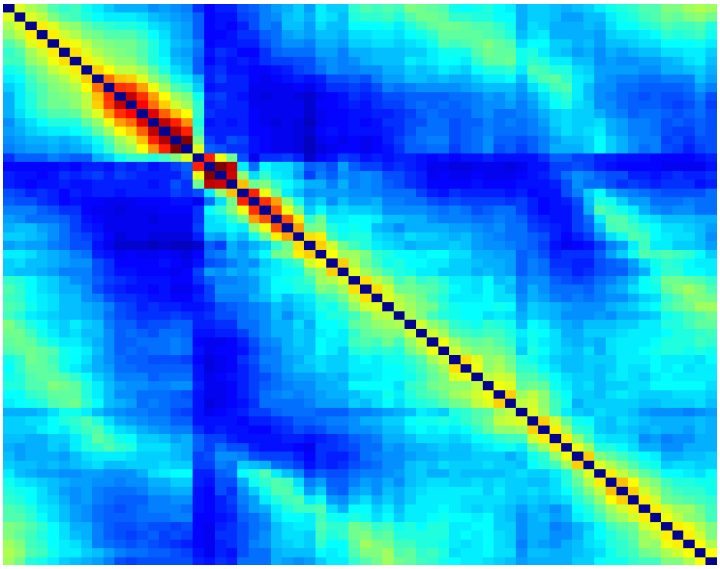


Figure 5.4: House sequence. PDM based 3D reconstruction.

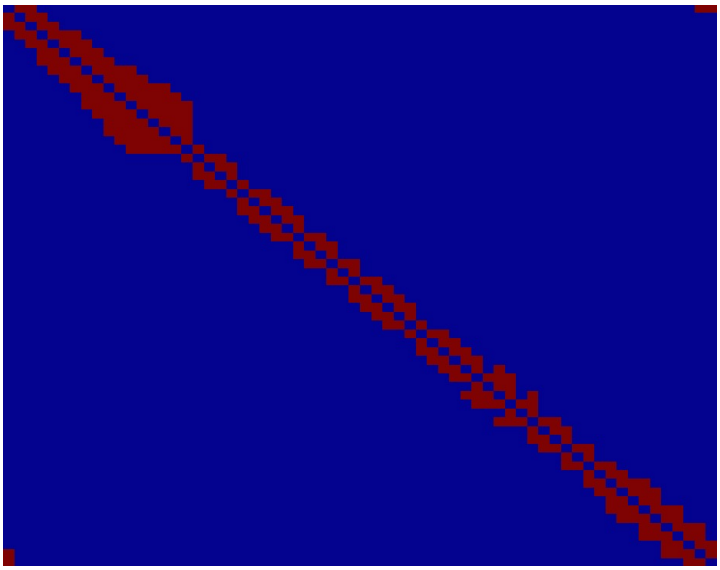
The "CUP" data set has multiple images of a 180 degree symmetric cup from all sides Figure 5.5. PDM reveals a strongly connected component along the diagonal for this data set, shown in warm colors in Figure 5.6a. Our global reasoning over the space of permutations substantially mitigates coherent errors. The binary match matrix was obtained by thresholding the weighted matrix Figure 5.6(b).

Figure 5.5: Representative images from CUP data set.

As is evident from the reconstructions, the baseline method only reconstruct a "half cup". Due to the structural ambiguity, it also concludes that the cup has two handles Figure 5.7(b). In contrast, the PDM reconstruction gives a perfect reconstruction of the full cup with a single handle Figure 5.7(a).



(a)



(b)

Figure 5.6: CUP data set. (a) Weighted data association matrix. (b) Binary data association matrix.

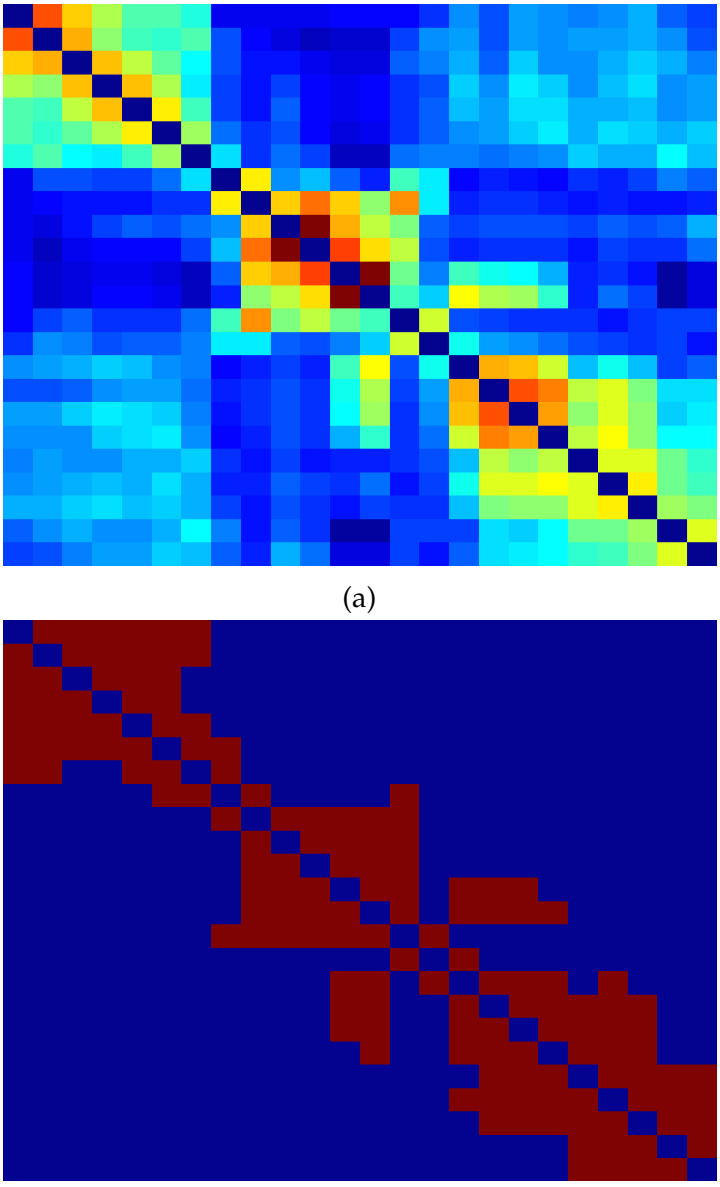
(a)

(b)

Figure 5.7: CUP data set. (a) PDM dense reconstruction. (b) Baseline dense reconstruction.

The "OAT" data set contains two instances of a red oat box, one on the left of a box of "Wheat Thins", and another on the right Figure 5.8. The PDM weighted match matrix and binary match matrix successfully discover strongly connected components, Figure 5.9(a) and Figure 5.9(b). The baseline method confused the two oat boxes as one, and reconstructs only a single box, Figure 5.10(b). Moreover, the structural ambiguity splits the Wheat Thins into two pieces. On the other hand, PDM gives a nice reconstruction of the two oat boxes with the entire Wheat Thins in the middle, Figure 5.10(a). Videos of 3D reconstructions are available on the project website: `http://pages.cs.wisc.edu/~pachauri/pdm/`.

Figure 5.8: Representative images from OAT data set.



(a)

(b)

Figure 5.9: OAT data set. (a) Weighted data association matrix. (b) Binary data association matrix.

(a)

(b)

Figure 5.10: OAT data set. (a) PDM dense reconstruction. (b) Baseline dense reconstruction.

5.4.2 Scene summarization for a set of images

In the second set of experiments, we considered the problem of selecting a small number of images from a large data set to summarize a given image data set. Such problems have been studied by others (Simon et al., 2007; Aner and Kender, 2002). The **Hyundai** data set has 35 images of the 2002 Hyundai Santa Fe model (Zhu et al., 2010). There are 54 landmark points, such as center of right headlight, center of left headlight, and so on. Each view captures only a subset of these interest points Figure 5.11. Putative

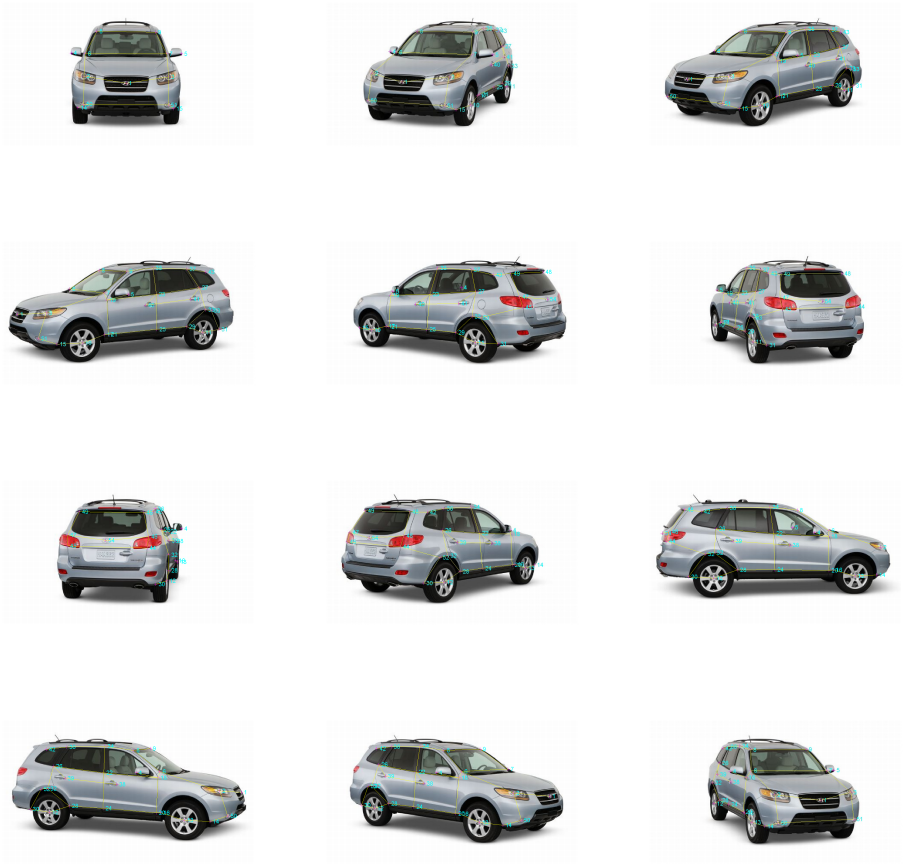


Figure 5.11: Representative images from HYUNDAI data set.

matches are computed for pair of images using the SIFT descriptor of interest points in each image. The *k-medoids* algorithm on permutation diffusion distance matrix, see Figure 5.12, selects summary images that are simultaneously "diverse" and "most centered", as expected from an ideal summary set, see Figure 5.13.

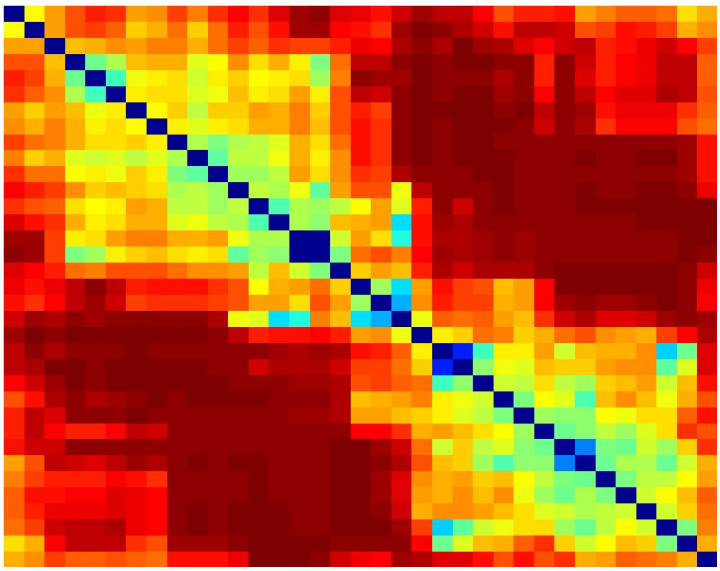


Figure 5.12: HYUNDAI data set. PDM based distance matrix.

Figure 5.13: HYUNDAI data set. Summary images found by the *k-medoids* algorithm for $k = 4$.

5.5 Summary

Inspired by the Vector Diffusion formalism of Singer and Wu (2012), we have proposed a new algorithm called Permutation Diffusion Maps for solving permutation synchronization problems, and an associated new affinity measure called Permutation Diffusion Affinity (PDA). Experiments show that the latter, in particular, can significantly improve the quality of Structure from Motion reconstructions of difficult scenes. Interestingly, PDA has an interpretation in terms of the inner product between two auto-correlation functions expressed in Fourier space, which, we believe, is a new approach to detecting hidden symmetries, with many potential applications even outside the realm of permutation problems.

In this chapter, we reiterate the main themes and contributions of this thesis. The road map for the future directions that could likely follow from the results described here, reveal the multifold relevance of this thesis.

6.1 Main ideas and contributions

With all the technical details and experiments in place, we are now in a position to return our attention to the key motivation that is at the center of this thesis. As summarized in Section 1.3, the aim of this thesis is to understand the connection between the combinatorial structure of matching problems, and the algebra. More concretely, we seek to understand *how* to capitalize on the high degree of regularity of this structure, as we approach real world matching problems; *how* to elucidate important matching tasks and generalize them on multiple instances. Each chapter in this thesis advances our understanding of these questions. We focused on core challenges in several computer vision problems where existing formulations run into difficulties which makes the ideas/algorithms described here immediately applicable. Looking more closely at the analysis tasks that motivated much of this work in machine learning helps identify new applications which can leverage the ideas proposed in this thesis, towards (potentially) more efficient algorithms. Recall that some of the main contributions of this thesis are:

- An efficient parameter learning framework for a class of combinatorial problems where the solution is a candidate in the symmetric group $\mathbb{S}_n$, (Pachauri et al., 2012).
- A Fourier spectra based framework to solve multi-way matching problem from noisy pairwise information. Our proposed framework

is capable of handling large random noise by allowing nominal user supervision, (Pachauri et al., 2013).

- A global inference method to investigate multi-way matching problem in the presence of large coherent noise. Our proposed method interprets noisy and incomplete matchings as functions defined on the discrete group $\mathbb{S}_n$, and uses the invariance properties of these functions for analysis, (Pachauri et al., 2014).

- In a collaborative role, an open source software library, $\mathbb{S}_n$Julia, to facilitate harmonic analysis of functions defined on the $\mathbb{S}_n$, (Plumb et al., 2014). The library is developed in a new programming language called *Julia*, led by my undergraduate mentee, Greg Plumb.

6.2 Future Directions

Computer vision. Matching is a critical component of various computer vision applications, and our experiments demonstrated that improvements in the matching results directly improves the quality of the downstream analysis for vision applications. For instance – a PDM based image association matrix leads to *good* quality 3D reconstruction of scenes with large duplicate structures. A natural question is: what are some other vision applications which may benefit from some of the proposed methods? The list may include:

- Image retrieval from a potentially large database. Earlier approaches have demonstrated the utility of diffusion processes for retrieval, where the edges are scalar attributes of similarity (Donoser and Bischof, 2013). A framework that equips the underlying graph with an appropriate transformation may provide better results;

- Representative selection problem to summarize the large image data set for recognition task. As the modern data sets are growing exponentially every day, this application has a lot of potential. Analyzing large data sets as a whole has many disadvantages: hard to elucidate the complete data set when presented at once; significant cost of storing information on the entire data. Computational efficiency of various recognition algorithms can improve when presented with a diverse exemplar set.

Matching problems. The main contribution of this thesis is to the state-of-the-art of matching problems. Therefore, it is not surprising that various interesting extensions of matching problems immediately follow. Some potential applications include:

- Exploring additional structure in domain specific matching problems using other forms of parameterization. For instance – weight updates on the frequency components instead of the features which may provide deeper insights;

- Multi-way matching problem to enforce higher order *consistency*. For example – second order consistency among multiple QAP instances.

Alzheimer's disease progression and $\mathbb{S}_n$. In the context of understanding progression of a neurological disease such as Alzheimer's disease, one is interested in the following question: how does the brain deteriorate as the disease develops and progresses. In particular, the characteristic sequence (ordering) in which cognitive abilities become abnormal due to the AD, is of great interest to construct disease staging systems that can be effectively used in diagnosis and treatment.

Previous studies described the disease progression using an *event based model* (EBM), (Fonteijn et al., 2012). Events in this context refer to clinical

events which correspond to the changes in the patient state that can be either a significant change in symptoms or abnormality in pathology. The bio-marker list include cerebrospinal fluid proteins, imaging based atrophy measures and so on. The disease progression is then defined as a single universal ordering in which these bio-markers become abnormal. Initial methods defined bio-marker "abnormality" by selecting a cut point value (Jack et al., 2011). For each bio-marker, cut points were established by comparing the bio-marker measurements between controls and AD subjects in an independent autopsy cohort. Notice that cut points are hard to validate for a general cohort.

Recently, various groups have proposed a probabilistic approach. This approach learns the bio-marker distributions for controls and AD from the data independently (Young et al., 2014). This at least partially settles the debate on cut point methods. In particular, these methods estimate two independent likelihood functions from the measurements: one for the normal state of the bio-marker, and another for the abnormal state of the bio-marker. Though the particular choice for likelihood function is directly informed by the type of bio-marker and clinically labeled control/AD subjects in the cohort, it has been shown that the choice of likelihood function does not generalize readily, i.e., clinical events and atrophy events require different treatments. Further, a complete independence assumption in the bio-marker measurement matrix is an over simplification of a rather complex concept of brain degeneration. Nevertheless, the probabilistic EBM model use a likelihood function to compute the probability density function (PDF) of a particular event ordering given the data. As the final form of the PDF becomes intractable, sampling methods such as MCMC are used. Finally, the maximum likelihood estimate (MLE) of the PDF is obtained by averaging the sequences in the most recent sampled set which gives the characteristic event ordering and additional information regarding the uncertainty around MLE ordering.

An interesting line of future research, is to investigate a new framework for disease progression – employing a group theoretic interpretation of the clinical events. In an ongoing project, we are using neuropsychological tests to asses cognitive abilities of individuals in large scale heterogeneous population cohort. The data is taken from the WRAP investigation in which middle aged asymptomatic population is studied to evaluate various levels of AD risks (Carlsson et al., 2014). At present, we have data for 1245 participants. Clinical diagnosis is not available for this data, and therefore fitting data likelihood functions for control/AD is not applicable.

Instead, we propose a new method to use measurement matrix for estimating the disease progression. We extract putative ordering of cognitive abilities for each subject using entire measurement matrix, i.e., individual subject ordering is conditioned on all other subjects. In particular, we use a percentile ranking procedure to extract an ordering of tests (cognitive domains) for each subject. Thus, each subject is represented as a permutation over cognitive markers in order of worst to best performance. We used a simple averaging method to find ordering patterns in our preliminary data set of 116 subjects for which we are provided with all measurements and no-missing data. Interestingly, the proposed pre-processing step which requires no involved modeling or cut point, indicates a clear pattern in the sequence, see Figure 6.1. The preliminary analysis is cross-sectional, i.e., the method is comparing measurements at a single time point. Notice, individual performance rank can be expressed as a probability distribution over $\mathbb{S}_n$. Formally, each subject is represented as a probability distribution centered around its putative percentile ranking. This way we capture the variability in the observations due to noise or missing data. The main idea is to use this probabilistic representation of the measurements on $\mathbb{S}_n$ and reason about the underlying sequence in which cognitive markers become abnormal in subjects that are at high risk of developing AD. The proposed method is distinctly different from the existing methods for disease pro-

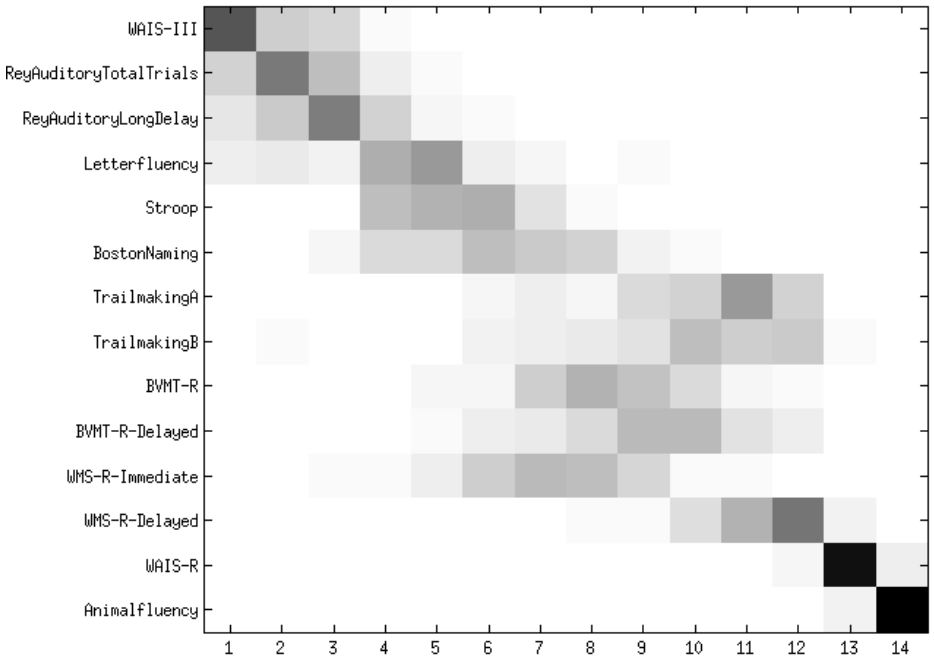


Figure 6.1: Positional variance diagrams showing the distribution of event sequences in population estimated by rank percentile method on a small subset of bio-markers. The rows correspond to different types of measurements related to various cognitive domains. Each subject appears at a percentile location for each bio-marker. Now, if we were to fix a subject and evaluate the "ranking" of each bio-marker for that individual, it provides a notion of association between different subjects. We expect that participants with similar disease progression trajectories will manifest similar such rankings.

gression. For the first time, we are proposing to only use non-invasive neuropsychological tests to model the disease progression without any cut points (which is not possible to obtain in an asymptomatic population). Our model is data-driven which will identify groups of subjects showing similar cognitive decline without any restrictive assumptions.

Open problem. Another interesting line for future work would be the direct extensions of Chapter 4 and Chapter 5 to other groups. As mentioned in Proposition 5.1, the proposed procedures for multi-reference problem, on ordered or unordered data set, can be generalized to any *compact* group with identity and a unitary representation. However, there are various applications that involve *non-compact* groups. For instance – the population atlas construction for brain image analysis (Thompson and Toga, 1996). The key idea behind deformable registration applied to brain images is to superimpose images from different subjects to enable very precise localization of morphological characteristics in population studies. It has numerous important clinical applications such as spatial normalization of functional images, group analysis and statistical parametric mapping (Friston et al., 1994).

Currently, the population-wide brain alignment problem is solved sequentially. A candidate brain from the cohort is randomly selected as a template. Then, the template brain is iteratively deformed to match all subjects in the data set, (Shen and Davatzikos, 2002; Thirion, 1996; Christensen et al., 1996). Such procedures are prone to the same type of issues as described in Chapter 4 and Chapter 5. It should be evident that if all of the pairwise information is used at once, most of these issues can be mitigated. This thesis suggests a natural framework for this problem – a multi-reference problem where the transformations between pair of brain images is a diffeomorphism.

Note that a diffeomorphism is a smooth, differentiable, invertible map, which under the composition operation is a group. This implies, diffeomorphism satisfy the group properties: closure, associative, inverse and identity. But there is very little known about the representation theory of the infinite dimensional diffeomorphism group. In practice, deformations are represented discretely with a definite number of parameters. Additional constraints are used on these discrete representations to en-

sure diffeomorphism (i.e., nonzero Jacobian determinant). Though the parametric representation is widely used in the community, these representations are not suitable to exploit the underlying group structure. This highlights the gap between theory and practice. Therefore, a better understanding of the representation theory of non-compact groups will enable interesting applications in the years to come.

Agarwal, S., Y. Furukawa, N. Snavely, I. Simon, B. Curless, S.M. Seitz, and R. Szeliski. 2011. Building Rome in a day. *Communications of the ACM* 54.

Aner, A., and J. R. Kender. 2002. Video summaries through mosaic-based shot and scene clustering. *ECCV*.

Balas, E., and J. B. Mazzola. 1980. Quadratic 0-1 programming by a new linearization. *Joint ORSA/TIMS National Meeting*.

Bandeira, A. S., A. Singer, and D. A. Spielman. 2013. A Cheeger inequality for the graph connection Laplacian. *Journal of Matrix Analysis and Applications* 34.

Bazaraa, M. S., and H. D. Sherali. 1980. Benders' partitioning scheme applied to a new formulation of the quadratic assignment problem. *Naval Research Logistics Quarterly* 27.

Belongie, S., J. Malik, and J. Puzicha. 2002. Shape matching and object recognition using shape contexts. *Transactions on Pattern Analysis and Machine Intelligence* 24.

Benaych-Georges, F, and R.R. Nadakuditi. 2011. The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices. *Advances in Mathematics* 227.

Berg, A.C., T.L. Berg, and J. Malik. 2005. Shape matching and object recognition using low distortion correspondences. *CVPR*.

Burkard, R. E. 1984. Quadratic assignment problems. *European Journal of Operational Research* 15.

Burnside, W. 1911. *Theory of groups of finite order*. Cambridge University Press.

Caelli, T., and T. Caetano. 2005. Graphical models for graph matching: Approximate models and optimal algorithms. *Pattern Recognition Letters* 26.

Caetano, T., J. J. McAuley, L. Cheng, Q. V. Le, and A. Smola. 2009. Learning graph matching. *Transactions on Pattern Analysis and Machine Intelligence* 31.

Carlsson, C. M., S. Johnson, et al. 2014. CSF biomarker profiles in middle-aged adults with parental history of AD: The Wisconsin ADRC cohorts. *Alzheimer's & Dementia: The Journal of the Alzheimer's Association* 10.

Cela, E. 1998. *The quadratic assignment problem: Theory and algorithms*. Kluwer Academic.

Chen, J-Q., J. Ping, and F. Wang. 1989. *Group representation theory for physicists*, vol. 7. World Scientific.

Chen, Y., L. Guibas, and Q. Huang. 2014. Near-optimal joint object matching via convex relaxation. *ICML*.

Christensen, G. E., R. D. Rabbitt, and M. I. Miller. 1996. Deformable templates using large deformation kinematics. *Transactions on Image Processing* 5.

Chung, F. R. K. 1997. *Spectral graph theory*, vol. 92. American Mathematical Society.

Clausen, M. 1989. Fast generalized Fourier transforms. *Theoretical Computer Science* 67.

Conte, D., P. Foggia, C. Sansone, and M. Vento. 2004. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence* 18.

Crandall, D., A. Owens, N. Snavely, and D. P. Huttenlocher. 2011. Discrete continuous optimization for large scale structure from motion. *CVPR*.

Dantzig, G. B. 1951. Application of the simplex method to a transportation problem. *Activity Analysis of Production and Allocation* 13.

Demirci, M.F., A. Shokoufandeh, Y. Keselman, L. Bretzner, and S. Dickinson. 2006. Object recognition as many-to-many feature matching. *International Journal of Computer Vision* 69.

Diaconis, P. 1988. Group representations in probability and statistics. *Lecture Notes-Monograph Series*.

———. 1989. A generalization of spectral analysis with application to ranked data. *The Annals of Statistics* 17.

Donoser, M., and H. Bischof. 2013. Diffusion processes for retrieval revisited. *CVPR*.

Duan, K., D. Parikh, D. Crandall, and K. Grauman. 2012. Discovering localized attributes for fine-grained recognition. *CVPR*.

Duchenne, O., A. Joulin, and J. Ponce. 2011. A graph-matching kernel for object categorization. *ICCV*.

Enqvist, O., F. Kahl, and C. Olsson. 2011. Non-sequential structure from motion. *ICCV Workshops*.

Finley, T., and T. Joachims. 2008. Training structural SVMs when exact inference is intractable. *ICML*.

Fischler, M. A., and R. A. Elschlager. 1973. The representation and matching of pictorial structures. *Transactions on Computers* 22.

Fonteijn, H. M, et al. 2012. An event-based model for disease progression and its application in familial Alzheimer's disease and Huntington's disease. *NeuroImage* 60.

Forsyth, D. A., and J. Ponce. 2003. A modern approach. *Computer Vision: A Modern Approach*.

Friston, K.J., A.P. Holmes, K.J. Worsley, J-P Poline, C.D. Frith, and R.S.J Frackowiak. 1994. Statistical parametric maps in functional imaging: A general linear approach. *Human Brain Mapping* 2.

Füredi, Z., and J. Komlós. 1981. The eigenvalues of random symmetric matrices. *Combinatorica* 1.

Gilmore, P. C. 1962. Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of Applied Mathematics* 10.

Goesele, M., N. Snavely, B. Curless, H. Hoppe, and S.M. Seitz. 2007. Multi-view stereo for community photo collections. *ICCV*.

Govindu, V. M. 2006. Robustness in motion averaging. *ACCV*.

Hadani, R., and A. Singer. 2011a. Representation theoretic patterns in three dimensional cryo-electron microscopy I–The intrinsic reconstitution algorithm. *Annals of mathematics* 174.

———. 2011b. Representation theoretic patterns in three-dimensional cryo-electron microscopy II – The class averaging problem. *Foundations of Computational Mathematics* 11.

Hancock, E., and R. Wilson. 2009. Graph-based methods for vision: A Yorkist manifesto. *Structural, Syntactic, and Statistical Pattern Recognition* 2396.

Harris, C., and M. Stephens. 1988. A combined corner and edge detector. *AVC* 15.

Hartley, R., and A. Zisserman. 2000. *Multiple view geometry in computer vision*, vol. 2. Cambridge University Press.

Hauagge, D. C., and N. Snavely. 2012. Image matching using local symmetry features. *CVPR*.

Havlena, M., A. Torii, J. Knopp, and T. Pajdla. 2009. Randomized structure from motion based on atomic 3d models from camera triplets. *CVPR*.

Hitchcock, F. L. 1941. The distribution of a product from several sources to numerous localities. *Journal of Mathematical Physics* 20.

Huang, B. C., and T. Jebara. 2011. Fast b-matching via sufficient selection belief propagation. *AISTATS*.

Huang, J., C. Guestrin, and L. Guibas. 2009. Fourier theoretic probabilistic inference over permutations. *Journal of Machine Learning Research* 10.

Huang, Q-X., and L. Guibas. 2013. Consistent shape maps via semidefinite programming. *Computer Graphics Forum* 32.

Jack, C. R., P. Vemuri, et al. 2011. Evidence for ordering of Alzheimer disease biomarkers. *Archives of Neurology* 68.

James, G. D. 1987. *The representation theory of the Symmetric groups*. Reading, Mass.

Jebara, T., J. Wang, and S.F. Chang. 2009. Graph construction and b-matching for semi-supervised learning. *ICML*.

Jiang, N., P. Tan, and L. F. Cheong. 2012. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. *CVPR*.

Joachims, T., T. Finley, and C. N. Yu. 2009. Cutting-plane training of structural SVMs. *Machine Learning* 77.

Jonker, R., and A. Volgenant. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* 38.

Kaufman, L., and F. Broeckx. 1978. An algorithm for the quadratic assignment problem using Bender's decomposition. *European Journal of Operational Research* 2.

Kondor, R. 2008. Group theoretical methods in machine learning. Ph.D. thesis, Columbia University.

———. 2010. A Fourier space algorithm for solving quadratic assignment problems. *SODA*.

Kondor, R., A. Howard, and T. Jebara. 2007. Multi-object tracking with representations of the Symmetric group. *AISTATS*.

Koopmans, T. C., and M. Beckmann. 1957. Assignment problems and the location of economic activities. *Journal of the Econometric Society*.

Kuhn, H.W. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2.

Lawler, E. L. 1963. The quadratic assignment problem. *Management Science* 9.

Lee, Y., and J. B. Orlin. 1994. *On very large scale assignment problems*. Springer.

Leordeanu, M., and M. Hebert. 2005. A spectral technique for correspondence problems using pairwise constraints. *ICCV*.

———. 2009. Unsupervised learning for graph matching. *CVPR*.

Leordeanu, M., M. Hebert, and R. Sukthankar. 2009. An integer projected fixed point method for graph matching and map inference. *NIPS*.

Li, R., H. Zhu, et al. 2010. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research* 20.

Lowe, D.G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60.

Martinec, D., and T. Pajdla. 2007. Robust rotation and translation estimation in multiview reconstruction. *CVPR*.

Maslen, D. 1998. The efficient computation of Fourier transforms on the Symmetric group. *Mathematics of Computation* 67.

Messing, R., C. Pal, and H. Kautz. 2009. Activity recognition using the velocity histories of tracked keypoints. *ICCV*.

Mikolajczyk, K., and C. Schmid. 2004. Scale & affine invariant interest point detectors. *International Journal of Computer Vision* 60.

Nguyen, A., M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas. 2011. An optimization approach to improving collections of shape maps. *Computer Graphics Forum* 30.

Ozyesil, O., A. Singer, and R. Basri. 2015. Stable camera motion estimation by convex programming. *Journal of Imaging Science*.

Pachauri, D., M. Collins, V. Singh, and R. Kondor. 2012. Incorporating domain knowledge in matching problems via Harmonic analysis. *ICML*.

Pachauri, D., R. Kondor, G. Sargur, and V. Singh. 2014. Permutation diffusion maps with application to the image association problem in computer vision. *NIPS*.

Pachauri, D., R. Kondor, and V. Singh. 2013. Solving the multi-way matching problem by permutation synchronization. *NIPS*.

Pardalos, P., F. Rendl, and H. Wolkowicz. 1994. *The quadratic assignment problem: A survey and recent developments*, vol. 16. American Mathematical Society.

Petterson, J., T. Caetano, J. McAuley, and J. Yu. 2009. Exponential family graph matching and ranking. *NIPS*.

Plumb, G., D. Pachauri, R. Kondor, and V. Singh. 2014. $\mathbb{S}_n$Julia: A Julia toolkit for Harmonic analysis on the Symmetric group.

Pop, M., S. L. Salzberg, and M. Shumway. 2002. Genome sequence assembly: Algorithms and issues. *Computer* 35.

Rao, B. S. Y., H. F. Durrant-Whyte, and J. A. Sheen. 1993. A fully decentralized multi-sensor system for tracking and surveillance. *The International Journal of Robotics Research* 12.

Roberts, R., S. Sinha, R. Szeliski, and D. Steedly. 2011. Structure from motion for scenes with large duplicate structures. *CVPR*.

Rudin, W. 1962. *Fourier analysis on groups*, vol. 2808. Interscience Publishers.

Sagan, B. E. 2001. *The Symmetric group: Representations, combinatorial algorithms, and symmetric functions*, vol. 203. Springer.

Schaffalitzky, F., and A. Zisserman. 2002. Multi-view matching for unordered image sets, or "how do I organize my holiday snaps?". *ECCV*.

Schrjiver, A. 2002. On the history of combinatorial optimization. *Preprint available at www. cwi. nl* $\setminus \sim$ *lex*.

Serre, J. P. 1977. *Linear representations of finite groups*, vol. 42. Springer.

Shalev-Shwartz, S., Y. Singer, and N. Srebro. 2007. Pegasos: Primal estimated sub-gradient solver for SVM. *ICML*.

Shen, D., and C. Davatzikos. 2002. Hammer: Hierarchical attribute matching mechanism for elastic registration. *Transactions on Medical Imaging* 21.

Shi, J., and C. Tomasi. 1994. Good features to track. *CVPR*.

Simon, I., N. Snavely, and S.M. Seitz. 2007. Scene summarization for online image collections. *ICCV*.

Singer, A. 2011. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis* 30.

Singer, A., and Y. Shkolnisky. 2011. Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming. *Journal on Imaging Sciences* 4.

Singer, A., and H.-T. Wu. 2012. Vector diffusion maps and the connection Laplacian. *Communications of Pure and Applied Mathematics* 65.

Sinha, S. N., D. Steedly, and R. Szeliski. 2012. A multi-stage linear approach to structure from motion. *Trends and Topics in Computer Vision*.

Snavely, N., S. M. Seitz, and R. Szeliski. 2006. Photo tourism: Exploring photo collections in 3D. *SIGGRAPH*.

Snavely, N., S.M. Seitz, and R. Szeliski. 2008. Modeling the world from internet photo collections. *International Journal of Computer Vision* 80.

Taskar, B., C. Guestrin, and D. Koller. 2003. Max-margin markov networks. *NIPS*.

Terras, A. 1999. *Fourier analysis on finite groups and applications*, vol. 43. Cambridge University Press.

Thirion, J. P. 1996. Non-rigid matching using demons. *CVPR*.

Thompson, P., and A. W. Toga. 1996. A surface-based technique for warping three-dimensional images of the brain. *Transactions on Medical Imaging* 15.

Torr, P. HS. 2003. Solving markov random fields using semidefinite programming. *AISTATS*.

Tsochantaridis, I., T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. 2006. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 6.

Umeyama, S. 1988. An eigendecomposition approach to weighted graph matching problems. *Transactions on Pattern Analysis and Machine Intelligence* 10.

Vishwanathan, S. V. N., K. M. Borgwardt, and N. N. Schraudolph. 2007. Fast computation of graph kernels. *NIPS*.

Volkovs, M., and R. Zemel. 2012. Efficient sampling for bipartite matching problems. *NIPS*.

Weyl, H. 1997. *The classical groups: Their invariants and representations*, vol. 1. Princeton University Press.

Wigner, E.P. 1958. On the distribution of the roots of certain symmetric matrices. *The Annals of Mathematics* 67.

Wilson, K., and N. Snavely. 2013. Network principles for SfM: Disambiguating repeated structures with local context. *ICCV*.

Wu, C. 2013. Towards linear-time incremental structure from motion. *3D Vision - 3DV*.

Wu, C., S. Agarwal, B. Curless, and S. M. Seitz. 2011. Multicore bundle adjustment. *CVPR*.

Wu, Y., J. Lim, and M-H Yang. 2013. Online object tracking: A benchmark. *CVPR*.

Xu, Y., A. Fern, and S. Yoon. 2007. Discriminative learning of beam-search heuristics for planning. *IJCAI*.

Young, A. L., N. P. Oxtoby, et al. 2014. A data-driven model of biomarker changes in sporadic Alzheimer's disease. *Brain*.

Zach, C., A. Irschara, and H. Bischof. 2008. What can missing correspondences tell us about 3d structure and motion? *CVPR*.

Zach, C., M. Klopschitz, and M. Pollefeys. 2010. Disambiguating visual relations using loop constraints. *CVPR*.

Zhu, S., L. Zhang, and B. M Smith. 2010. Model evolution: An incremental approach to non-rigid structure from motion. *CVPR*.