

Synchronisation of Partial Multi-Matchings via Non-negative Factorisations

Florian Bernard¹, Johan Thunberg^{2,3}, Jorge Goncalves³, Christian Theobalt¹

¹MPI Informatics, Saarland Informatics Campus ²Halmstad University ³LCSB Luxembourg

Abstract

In this work we study permutation synchronisation for the challenging case of partial permutations, which plays an important role for the problem of matching multiple objects (e.g. images or shapes). The term synchronisation refers to the property that the set of pairwise matchings is cycle-consistent, i.e. in the full matching case all compositions of pairwise matchings over cycles must be equal to the identity. Motivated by clustering and matrix factorisation perspectives of cycle-consistency, we derive an algorithm to tackle the permutation synchronisation problem based on non-negative factorisations. In order to deal with the inherent non-convexity of the permutation synchronisation problem, we use an initialisation procedure based on a novel rotation scheme applied to the solution of the spectral relaxation. Moreover, this rotation scheme facilitates a convenient Euclidean projection to obtain a binary solution after solving our relaxed problem. In contrast to state-of-the-art methods, our approach is guaranteed to produce cycle-consistent results. We experimentally demonstrate the efficacy of our method and show that it achieves better results compared to existing methods.

Keywords: partial permutation synchronisation, multi-matching, spectral decomposition, non-negative matrix factorisation

1. Introduction

The problem of matching features across images or shapes is a fundamental topic in pattern recognition and vision and has a high relevance in a wide range of problems. Potential applications include shape deformation model learning [17, 26], object tracking, 3D reconstruction, graph matching, or image registration. The fact that many tasks that seek for a matching between a pair of objects can be formulated as the NP-hard quadratic assignment problem (QAP) [45] illustrates the difficulty of matching problems. The more general problem of matching an entire collection of objects, rather than a pair of objects, is referred to as *multi-matching*. In general, such multi-matching problems are computationally at least as difficult as pairwise matching problems, as they can be phrased in terms of simultaneously solving multiple pairwise matching problems that are coupled via consistency constraints. Using such couplings of pairwise problems is a common approach for solving multi-matching problem in practice [30, 66, 63, 7].

Due to the importance and practical relevance of making use of pairwise matchings to solve multi-matching problems, in this work we focus on studying *permutation synchronisation* methods. The aim of

these methods is to process a given set of “noisy” pairwise matchings such that *cycle-consistency* is achieved. In the case of full matchings, cycle-consistency refers to the property that compositions of pairwise matchings over cycles must be equal to the identity matching. Synchronisation methods have been studied extensively both in the context of multi-matching (e.g. [43, 44, 28, 16, 50, 59, 40, 47]) as well as for general transformations (e.g. [24, 25, 52, 14, 8, 2, 56, 61]). One can interpret the synchronisation methods as a denoising procedure, where the wrong matchings (i.e. the noise) that account for cycle inconsistencies in the set of pairwise matchings are to be filtered out.

Most commonly, the synchronisation of pairwise matchings is formulated as an optimisation problem over permutation matrices. In the works by Pachauri et al. [44] and Shen et al. [50], solutions for the synchronisation of permutation matrices based on a spectral factorisation are presented. A major limitation of these works is that the method is only suitable for *full* permutation matrices, i.e. it is assumed that all features are present in all objects (cf. Sec. 3.2). While this limitation has recently been addressed in the work by Maset et al. [40], in their work they do not aim for cycle-consistency. Since the (unknown) true matchings must

be cycle-consistent, we argue that cycle-consistency is essential and should be strived for.

The main objective of this work is to present a novel approach for the synchronisation of pairwise matchings that addresses the mentioned shortcomings of existing methods. To this end, we present an improved formulation for the permutation synchronisation problem that finds a non-negative approximation of the range space of the pairwise matching matrix. In contrast to [44], our approach can handle *partial* pairwise matchings. Moreover, unlike [40, 73], our approach guarantees cycle-consistent matchings.

Main contributions: The main contributions of our work on the synchronisation of partial permutations can be summarised as follows: (i) Motivated by clustering and matrix factorisation perspectives of cycle-consistency in the set of pairwise matchings, we derive an *improved algorithm for permutation synchronisation* based on non-negative factorisations. (ii) While the proposed formulation is non-convex, we propose a *novel procedure for initialising* the variables. (iii) Moreover, we present a *novel projection procedure* to obtain a binary solution from the relaxed formulation. (iv) Experimentally we demonstrate that our method achieves *superior results* on synthetic and real datasets, while addressing the aforementioned shortcomings.

2. Related Work

In this section we discuss prior work that is most relevant to our approach.

Transformation synchronisation: Synchronisation methods have been studied for various kinds of transformations. The synchronisation of (special) orthogonal transformations has been considered based on spectral methods [52, 5, 61], semidefinite programming [52, 15, 61], or Lie-group averaging [24, 14]. The case of rigid-body transformations, which is particularly relevant in the context of vision, has been studied in semidefinite programming frameworks [15, 4], as well as in the context of spectral approaches [8, 3]. In general, spectral approaches are more scalable compared to semidefinite programming methods. In addition to centralised methods, distributed synchronisation methods have also been presented, both for the case of undirected graphs [58], as well as for the more general case of directed graphs [56].

Permutation synchronisation: Since permutation matrices are a subset of the orthogonal matrices, one could consider permutation synchronisation as a special case of the orthogonal synchronisation methods. However, in general the permutation synchronisation prob-

lem appears to be more difficult due to the additional binary constraints. Moreover, if one considers partial permutations, this interpretation as special case is no longer valid. The synchronisation of full permutation matrices has been presented by Pachauri et al. [44], with follow-up works that consider partial matchings [2, 40]. We devote Sec. 3.2 to an in-depth explanation of these approaches, where we also identify their main weaknesses upon which our approach improves.

Matching problems: Matching problems between two objects are commonly formulated in terms of the linear assignment problem (LAP) [13, 42] or the quadratic assignment problem (QAP) [31, 32, 13, 37]. When one matches graphs, the LAP corresponds to matching node attributes only, whereas the QAP matches node attributes as well as edge attributes [72]. Computationally, the difference between both is that the LAP is solvable in polynomial time (e.g. via the Hungarian method [42] or the Auction algorithm [12]), whereas the QAP is NP-hard [45]. Hence, for solving QAPs in practice, existing approaches either resort to (expensive) branch and bound methods [6], or to approximations, e.g. based on spectral methods [35, 18], dual decomposition [57], linear relaxations [54, 55], convex relaxations [71, 46, 23, 22, 30, 1, 21, 7], path following [70, 72, 29], or alternating directions [33].

Multi-matching problems: The problem of matching more than two objects can be phrased as multi-graph matching (MGM) problems [62, 65, 67, 28, 67, 30, 51, 7, 27], which in general are computationally very challenging. If one uses first-order terms only, so that geometric relations between the features are not explicitly taken into account, multi-matching can efficiently be solved as (constrained) clustering problem [64, 59]. The approaches described in [30, 66, 63, 7] phrase MGM in terms of multiple pairwise matchings. The work in [73] is closely related to the permutation synchronisation methods [44, 2, 40], as the authors formulate the multi-matching problem directly in terms of a low-rank optimisation problem for a given set of pairwise matchings. However, the so-obtained matchings are generally not cycle-consistent.

3. Background

Notation: Let $\mathbf{1}_{pq}$ and $\mathbf{0}_{pq}$ denote $p \times q$ matrices comprising of ones and zeros, and we write $\mathbf{1}_p$ and $\mathbf{0}_p$ for $q = 1$. We use X_+ to denote that all negative elements in the matrix X are replaced by 0. For an integer $i \in \mathbb{N}$, we define $[i] := \{1, \dots, i\}$. For a $p \times q$ matrix X , and the index sets $A \subseteq [p]$, $B \subseteq [q]$, we denote by $X_{A,B}$ the $|A| \times |B|$ submatrix of X that is formed

from the rows with indices in A and the columns with indices in B . We use the colon notation to denote the full index set, e.g. $X_{:,B} = X_{A,B}$ for $A = [p]$. For matrices $A_{ij}, i \in [p], j \in [q]$ of appropriate sizes, we use the shorthand notation $[A_{ij}]_{ij}$ to denote the block matrix

$$[A_{ij}]_{ij} := \begin{bmatrix} A_{11} & \dots & A_{1q} \\ \vdots & \ddots & \vdots \\ A_{p1} & \dots & A_{pq} \end{bmatrix}. \quad (1)$$

The set of (full) permutation matrices is defined as

$$\mathbb{P}_p := \{X \in \{0, 1\}^{p \times p} : X\mathbf{1}_p = \mathbf{1}_p, \mathbf{1}_p^T X = \mathbf{1}_p^T\}. \quad (2)$$

The set of $p \times q$ partial permutation matrices \mathbb{P}_{pq} is defined as

$$\mathbb{P}_{pq} := \{X \in \{0, 1\}^{p \times q} : X\mathbf{1}_q \leq \mathbf{1}_p, \mathbf{1}_p^T X \leq \mathbf{1}_q^T\}. \quad (3)$$

3.1. Partial Permutation Synchronisation

Let $k \in \mathbb{N}, k > 2$ be the total number of objects (e.g. images or shapes) that are to be matched. We assume that in object $i \in \mathbb{N}$, with $i \in [k]$, there are $m_i \in \mathbb{N}$ features, where the total number of features is denoted as $m = \sum_{i=1}^k m_i$. Moreover, we assume that there is a total number of $d \in \mathbb{N}$ distinct features across all objects $i \in [k]$ in the *universe*. We use $P_{ij} \in \mathbb{P}_{m_i, m_j}$ to denote a (partial) permutation that encodes the matching between the i -th and the j -th object (Fig. 1(i)). The element $(P_{ij})_{pq} \in \{0, 1\}$ at position (p, q) , $p \in [m_i], q \in [m_j]$ of matrix P_{ij} is 1 iff the p -th feature of object i is matched to the q -th feature of object j . For $P_{ij} \in \mathbb{P}_{m_i, m_j}$, $W := [P_{ij}]_{i,j \in [k]} \in [\mathbb{P}_{m_i, m_j}]_{i,j \in [k]}$ is the $m \times m$ matrix of pairwise (partial) matchings.

Cycle-consistency of partial matchings: In contrast to full matchings, where cycle-consistency refers to the property that compositions of pairwise matchings over cycles must be *equal* to the identity matching, in the case of partial matchings one only requires that compositions of pairwise matchings over cycles must be *a subset of the identity matching*. Due to potential pairwise *non-matchings* (i.e. zero rows or columns in P_{ij}) along a cyclic path, some of the original matchings may vanish. A convenient way to define *cycle-consistency* of partial matchings is based on universe features:

Definition 1. *The matrix of pairwise (partial) matchings $W = [P_{ij}]_{i,j \in [k]}$ is said to be cycle-consistent (or synchronised) iff there exists a set $\{P_i \in \mathbb{P}_{m_i, d} : i \in [k], P_i \mathbf{1}_d = \mathbf{1}_{m_i}\}$ such that for all $i, j \in [k]$ it holds that $P_{ij} = P_i P_j^T$.*

The *object-to-universe matching* matrices $P_i \in \mathbb{P}_{m_i, d}$ can be interpreted as assignments of each feature of the i -th object to one of the features in the universe (Fig. 1(ii)), where the ℓ -th row of P_i is the assignment of the ℓ -th feature of object i to a particular feature in the universe. The requirement $P_i \mathbf{1}_d = \mathbf{1}_{m_i}$ ensures that each feature of object i is assigned to exactly one feature of the universe. For $\mathcal{U} := \{U \in [\mathbb{P}_{m_i, d}]_{i \in [k]} : U \mathbf{1}_d = \mathbf{1}_m\} \subset \mathbb{R}^{m \times d}$, one can characterise cycle-consistency of partial matchings in terms of a low-rank factorisation [40], which is also illustrated in Fig. 1(iii):

Lemma 2. *The pairwise (partial) matching matrix W is cycle-consistent iff there exists a matrix $U \in \mathcal{U}$, such that $W = UU^T$.*

Proof. To prove the statement we identify $U = [P_1^T \ P_2^T \ \dots \ P_k^T]^T \in \mathbb{R}^{m \times d}$. One can easily see, cf. Def. 1, that cycle-consistency implies that there exists a U that has the desired properties. Likewise, if a $U \in \mathcal{U}$ with $W = UU^T$ is given, one can see that the blocks $\{P_i\}$ of U satisfy $P_i \mathbf{1}_d = \mathbf{1}_{m_i}$, as well as $P_{ij} = P_i P_j^T$. \square

Optimisation problem: Lemma 2 shows that in the noise-free case, the matrix of pairwise matchings W can be factorised as $W = UU^T$. Given a *noisy* W , a straightforward way to phrase the permutation synchronisation problem is to consider the constrained nonlinear least-squares problem

$$\arg \min_{U \in \mathcal{U}} \|W - UU^T\|_F^2. \quad (4)$$

Since Problem (4) is non-convex, finding an exact solution is intractable for reasonably large instances. Hence, various simplifications have been considered in the literature, as we describe next.

3.2. Spectral Relaxations

In this section we summarise the key ideas of existing spectral relaxations, where we also identify their shortcomings when synchronising *partial* permutations. In order to avoid confusion, we explicitly mention that the reader should carefully distinguish between the $d \times d$ matrix $U^T U$ and the $m \times m$ matrix UU^T , as both terms will appear below.

Full matchings: In the case of (cycle-consistent) full matchings, it holds that $U^T U = k\mathbf{I}_d$. Thus, $\|W - UU^T\|_F^2 = \langle W, W \rangle - 2\langle W, UU^T \rangle + \langle UU^T, UU^T \rangle = \text{const} - 2\langle W, UU^T \rangle$. Hence, for full matchings, the authors of [44] relax the constraint $U \in \mathcal{U}$ to $U^T U = k\mathbf{I}_d$, and then solve Problem (4) with the relaxed constraints by eigendecomposition, followed by a projection step.

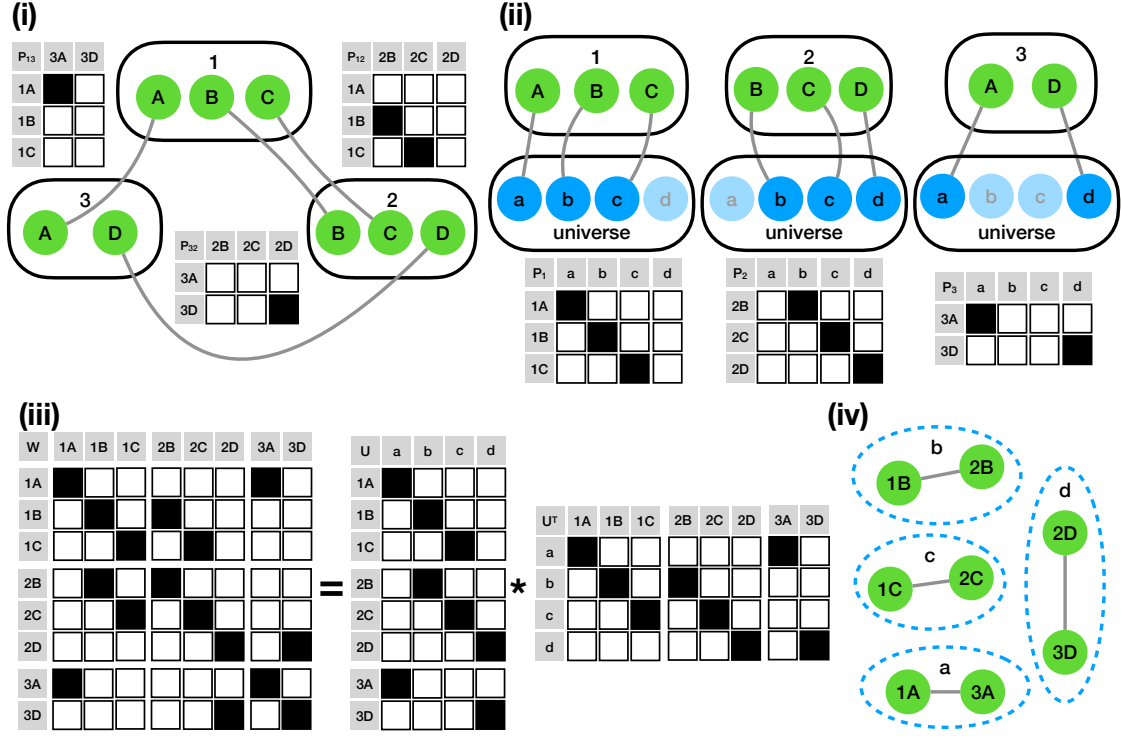


Figure 1: Conceptual illustration of (i) relative matchings, (ii) absolute matchings, (iii) the matrix factorisation perspective, and (iv) the graph of pairwise matchings. The objects are denoted by 1, 2 and 3, where corresponding features across objects are labelled by the same letter from A to D. The *relative* matchings are represented by the permutation matrices P_{ij} (e.g. P_{12}), and the *absolute* matchings are represented by the permutation matrices P_i (e.g. P_1) that match each feature to one of the universe features a, b, c, d. Since cycle-consistency holds in this case, the matrix W in (iii) can be factorised into UU^T (Lemma 2); and the graph of pairwise matchings in (iv) is a union of the disconnected cliques a, b, c and d (Lemma 3).

Partial matchings: For partial matchings, the authors of [40] propose to maximise $\langle W, UU^T \rangle$ based on eigendecomposition. However, in the *partial* matchings case, in general $U^T U \neq k\mathbf{I}_d$, so that the objective $\langle W, UU^T \rangle$ differs from the objective in Problem (4). Instead, for $U \in \mathcal{U}$ the objective $\langle W, UU^T \rangle$ counts the number of *equal matchings* between the matrices P_{ij} and $P_i P_j^T$ for all i, j . A further difficulty with *partial* matchings is related to the necessary projection due to the relaxation of the constraints, as we describe next.

Projection: When the constraint $U \in \mathcal{U}$ is replaced by $U^T U = k\mathbf{I}_d$, after obtaining U based on the spectral decomposition of W , one needs to project U onto the set \mathcal{U} . Since for any orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ it holds that $(UQ)(UQ)^T = UQQ^T U = UU^T$, the factorisation UU^T is only determined up to such a matrix Q . Hence, for projecting the blocks of U , one can choose a suitable orthogonal matrix Q in order to simplify the projection. For the full matching case, the authors of [44] suggest to perform Euclidean projections of the $d \times d$ blocks of UQ for the choice $Q = P_1^T$. Under the assumption that

W is relatively close to the form UU^T , the matrix P_1 is near-orthogonal, such that the first block of UQ is close to the identity matrix, while the remaining blocks of UQ shall become close to permutation matrices.

Since for partial permutations the matrices P_i are of dimension $m_i \times d$, where generally $m_i < d$, the assumption that the P_i are near-orthogonal breaks, and thus such a procedure is not applicable anymore (cf. Sec. 4.1 for details). As workaround, instead of projecting the blocks of U onto \mathcal{U} , the authors of [40] perform a projection of the blocks of UU^T , such that the $m \times m$ matrix $\text{proj}(UU^T)$ is obtained. While it is reasonable (under small noise assumptions) to assume that the blocks of UU^T are close to being (partial) permutation matrices, in this approach one cannot guarantee that the matrix $\text{proj}(UU^T)$ satisfies the conditions in Lemma 2, and thus, cycle-consistency is violated.

Another approach for the projection is pursued by the authors of [73, 2], where a greedy strategy is employed for obtaining blocks of partial permutations from the matrix of eigenvectors U .

3.3. Clustering Perspective

Here, we summarise the clustering perspective of synchronisation (cf. [2, 59]), which will become useful to motivate our approach in Sec. 4. For that, we consider the *graph of pairwise matchings* $\mathcal{G} := \mathcal{G}(W)$ (cf. Fig. 1(iv) for an illustration). The (non-negative) $m \times m$ matrix W is considered as the adjacency matrix of \mathcal{G} , so that \mathcal{G} comprises m nodes (recall that $m = \sum_i m_i$). The value $(W)_{pq} \in \mathbb{R}$ at position (p, q) of W denotes the edge weight that represents the affinity of nodes $p \in [m]$ and $q \in [m]$, where $(W)_{pq} = 0$ means that there is no edge. Note that w.l.o.g. we assume $(W)_{pp} = 1$ for all $p \in [m]$. As shown by Tron et al. [59], and illustrated in Fig. 1(iv), cycle-consistency can compactly be formulated in terms of the graph of pairwise matchings:

Lemma 3. *The graph of pairwise matchings $\mathcal{G}(W)$ is cycle-consistent iff it is a union of disconnected cliques.*

Proof. See Prop. 2 in [59]. \square

Lemma 4. *Let the graph of pairwise matchings $\mathcal{G}(W)$ be cycle-consistent so that it is a union of the disconnected cliques $C_i \subseteq [m]$, $i \in [d]$. It holds that all columns of the matrix $W_{:,C_i} \in \{0, 1\}^{m \times |C_i|}$ are equal for a given $i \in [d]$.*

Proof. We denote by c_i , $i \in [d]$, the number of elements in the i -th clique. Since \mathcal{G} is a union of d disconnected cliques, there is a permutation $P \in \mathbb{P}_m$ such that PWP^T is the block-diagonal matrix $PWP^T = \text{diag}(\mathbf{1}_{c_1 c_1}, \dots, \mathbf{1}_{c_d c_d})$. Moreover, for P it holds that $\mathbf{1}_{:,C_i} = P^T \mathbf{1}_{:,A_i}$ for $A_i = \{d_i+1, d_i+2, \dots, d_i+c_i\}$ with $d_i = \sum_{\ell=1}^{i-1} c_\ell$. From $\mathbf{1}_{:,C_i} = P^T \mathbf{1}_{:,A_i}$ it follows that $(PW)_{:,C_i} = (PW)\mathbf{1}_{:,C_i} = (PWP^T)\mathbf{1}_{:,A_i} = \text{diag}(\mathbf{1}_{c_1 c_1}, \dots, \mathbf{1}_{c_d c_d})\mathbf{1}_{:,A_i} = [\mathbf{0}_{c_1 c_1}^T, \dots, \mathbf{0}_{c_i c_i-1}^T, \mathbf{1}_{c_i c_i}^T, \mathbf{0}_{c_i c_i+1}^T, \dots, \mathbf{0}_{c_i c_d}^T]^T$, which shows that the columns of $(PW)_{:,C_i}$ are equal. Hence, with PW being a permutation of the rows of W , the columns of $W_{:,C_i}$ must also be equal. Since cycle-consistency implies symmetry of W , the analogous statement also holds for the rows of W . \square

Lemma 4 illustrates that one can cluster the columns (or rows) of W to identify to which *universe feature* they belong (cf. Fig. 1(iv)).

4. Proposed Approach

A key idea of our approach is to formulate the permutation synchronisation problem in terms of a *non-negative matrix factorisation* (NMF) [34]. To be more specific, we propose to solve

$$\arg \min_{V \geq 0, H \geq 0} \|W - VH\|_F^2, \quad (5)$$

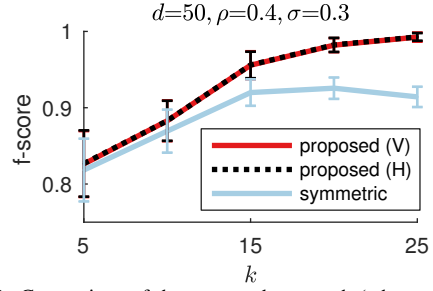


Figure 2: Comparison of the proposed approach (when using either V or H to obtain the cycle-consistent matchings) with a *symmetric* NMF [60]. While using an *unsymmetric factorisation* is clearly advantageous, both V and H can be used equivalently.

where $V \in \mathbb{R}^{m \times d}$ and $H \in \mathbb{R}^{d \times m}$. Problem (5) is a relaxation of Problem (4), where the constraints $V = H^T$ are dropped, and the constraint set \mathcal{U} is replaced by non-negativity constraints. At first sight it may appear unnatural that one aims for an *unsymmetric* factorisation VH of the symmetric matrix W . However, we have found that this is advantageous compared to a symmetric factorisation (see Fig. 2), which we believe is due to the following reasons: (i) On the one hand, from a theoretical perspective the factorisation VH enables to get a better rank- d approximation of W (cf. Lemma 2) compared to enforcing H^T to be equal to V . (ii) On the other hand, the unsymmetric NMF optimises over a higher-dimensional space, such that it has more freedom during the optimisation and is thus less prone to unwanted local optima of the non-convex Problem (5). (iii) Furthermore, with the inherent clustering properties of NMF [20, 36, 19, 69, 39, 68], Problem (5) can also be understood from the clustering point-of-view (cf. Sec. 3.3). In the clustering perspective, the columns of the matrix V can be seen as the *cluster centres*, where each column of W is a conic combination of the columns of V , and the corresponding column of H contains the coefficients. Since swapping the roles of V and H is equivalent to factorising W^T in place of W , using either V or H for obtaining the cycle-consistent partial matchings from the *unsymmetric factorisation* VH are equivalent, as also demonstrated in Fig. 2. Note that due to points (i) and (ii) it nevertheless is important that the factorisation is *unsymmetric* (cf. Fig. 2).

The motivation for enforcing both V and H to be non-negative is as follows: when cycle-consistency holds, the columns of V should be non-negative and mutually orthogonal, so that each row in V can contain at most one non-zero element. Thus, if the factor matrix H is such that $W = VH$, then, since W is non-negative, H needs to be non-negative.

Next, we introduce our rotation scheme that is used

for the initialisation of V and H , as well as for the projection of V onto \mathcal{U} .

4.1. Rotation Scheme

For $X_i \in \mathbb{R}^{m_i \times d}$, $i \in [k]$, let $X = [X_1^T, \dots, X_k^T]^T \in \mathbb{R}^{m \times d}$ be a rank- d matrix that comprises a low-rank approximation of W , i.e. $W \approx XX^T$. For any orthogonal matrix Q we have that $XX^T = (XQ)(XQ)^T$, so that we can freely choose Q and use $(XQ)(XQ)^T$ as low-rank approximation of W in place of XX^T . The purpose of this section is to describe a procedure to find a Q , such that XQ is closer to the set \mathcal{U} compared to X , which is for example beneficial for performing a Euclidean projection of X onto \mathcal{U} . To this end, we generalise the full-matching rotation scheme in [44], which will be explained in the next paragraph, such that one can find a suitable orthogonal matrix Q for the case of *partial* matchings.

Challenges: As discussed in Sec. 3.2, in the case of full matchings, i.e. $m_1 = \dots = m_k = d$, the authors of [44] set $Q = X_i^T$ for one of the block indices $i \in [k]$, e.g. $Q = X_1^T$, so that XQ is close to a matrix that comprises blocks of permutations. This is based on the assumption that the pairwise matchings XX^T are close to the ground truth, which in turn implies that (i) each X_i^T is near-orthogonal, so that $X_i X_i^T \approx \mathbf{I}_d \in \mathbb{P}_d$ for all $i \in [k]$; and that (ii) there exists an orthogonal $Q \in \mathbb{R}^{d \times d}$ such that XQ is close to comprising blocks of permutation matrices, so that for all $j \in [k]$ there exists a $P_j \in \mathbb{P}_d$ such that $X_j X_j^T \approx P_j$. Essentially, due to (i) and (ii) it is ensured that XQ is close to \mathcal{U} whenever $Q = X_i^T$ for any $i \in [k]$.

For partial matchings, point (i) is not valid anymore, because generally not *all* the universe features are present in each object $i \in [k]$. Hence, the $X_i \in \mathbb{R}^{m_i \times d}$ are (generally) not orthogonal (as they are not even square matrices), from which it follows that $XX_i^T (XX_i^T)^T = XX_i^T X_i X^T \neq XX^T$. For partial matchings, it is not sufficient to consider only a single block X_i of X for constructing Q . Instead, one needs to aggregate information from rows of X that come from different blocks X_1, \dots, X_k . We tackle this using the *Successive Block Rotation Algorithm* (SBRA), as we describe next.

Successive Block Rotation Algorithm: Similarly as in [44], we assume that a given X forms a sufficiently good approximation XX^T to the (unknown) ground truth matchings. With that, there must exist an orthogonal Q such that XQ is close to an element of \mathcal{U} , in which case each row of XQ has a single element that is close to one, with all other elements being close to zero. When we make particular elements in XQ close to one by rotating X by Q , we say that we *activate* these elements.

For finding a suitable orthogonal matrix Q in the case of partial matchings, we successively select elements of X that shall be activated. Moreover, we ensure that at most one element in each row in XQ is activated, so that all other elements in these rows become small (based on the above assumption). To this end, we employ an $(m \times d)$ -dimensional binary matrix C , which has the purpose of masking those elements that shall become activated in the rotated XQ . For now, let us assume that we are given a $C \in \{0, 1\}^{m \times d}$. With that, we consider the problem

$$Q := \arg \max_{\bar{Q}^T \bar{Q} = \mathbf{I}_d} \langle C, X\bar{Q} \rangle, \quad (6)$$

so that the orthogonal matrix Q is chosen such that the elements of the rotated XQ are as large as possible at the active positions C . This problem can be solved by setting $Q = \bar{U}\bar{V}^T$, for $\bar{U}\bar{\Sigma}\bar{V}$ being the singular value decomposition (SVD) of $X^T C$. For example, in the case of full matchings, when using $C = [\mathbf{I}_d, \mathbf{0}_{d, m-d}]^T$, the diagonal elements of the first block X_1 of X are activated. With such a choice of C we obtain $X^T C = X_1^T$, which corresponds to the rotation approach in [44] with an additional SVD-based orthogonalisation of X_1^T . The important difference that makes our approach applicable to *partial* matchings is that we successively construct the matrix C , rather than activating elements of a single block X_i for some fixed i . The *Successive Block Rotation Algorithm* (SBRA) is summarised as follows:

- (i) First, we initialise C to contain an $m_\ell \times m_\ell$ identity matrix in the ℓ -th block, where $\ell = \arg \max_i m_i$. All other elements of C are zero.
- (ii) Given C , we obtain Q by solving Problem (6).
- (iii) Based on $X_{\text{rot}} = XQ$, we update C so that the inactive rows of X_{rot} chosen from the block with the largest number of inactive rows will be activated in the next step, as well as all active elements remain active.

Step (ii) and (iii) are repeated until there are no further elements of X_{rot} that shall become activated. We illustrate our algorithm in Fig. 3.

4.2. Initialisation

Since Problem (5) is non-convex, the initialisation of the matrices V and H plays a crucial role. We propose to initialise V and H based on a rotation of the spectral factorisation of the pairwise matching matrix W . Hence, we first compute the best rank- d approximation of W using eigendecomposition, so that $W \approx XX^T$, where

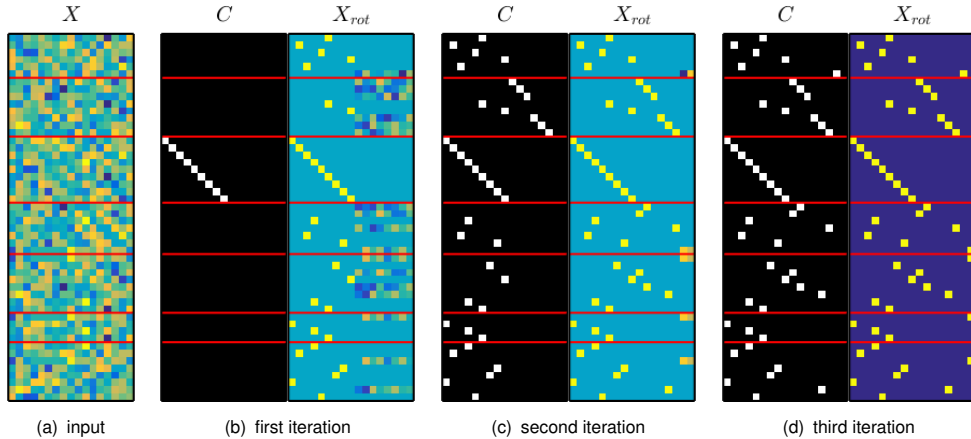


Figure 3: Illustration of the *Successive Block Rotation Algorithm*. In each subimage the $k=7$ blocks are separated by red horizontal lines. (a) The input matrix X . (b) In the first iteration, the masking matrix C is initialised so that the third block (which is the largest one) contains an $(m_3 \times m_3)$ -dimensional identity matrix. Solving Problem (6) results in the rotated X_{rot} . (c) In the second iteration, C is updated such that on the one hand all active elements of the previous X_{rot} remain active, and on the other hand all the inactive rows in the block with the largest number of inactive rows will be activated (in this case the second block). (d) The third iteration produces X_{rot} , where $X_{rot} \in \mathcal{U}$.

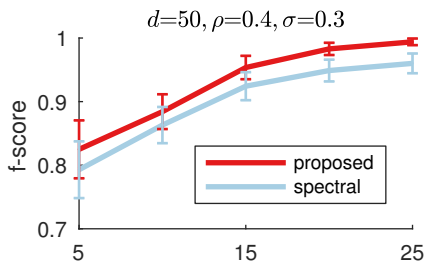


Figure 4: Comparison of proposed vs. spectral initialisation (cf. Sec. 5.1 for details).

$X \in \mathbb{R}^{m \times d}$ is the matrix of the (scaled) most dominant eigenvectors of W . Subsequently, we rotate the columns of X with Q , so that it becomes closer to \mathcal{U} , as described in Sec. 4.1. Since we use an NMF algorithm based on multiplicative updates (cf. Sec. 4.4) that requires a non-negative initialisation, we set $V = (XQ)_+$ and $H = (XQ)_+^T$. In Fig. 4 we demonstrate that the proposed initialisation procedure is superior compared to using the spectral initialisation X_+ .

4.3. Projection onto \mathcal{U}

After solving Problem (5) (with the algorithm described in Sec. 4.4), we perform a projection-after-rotation, i.e. we find Q based on the SBRA (Sec. 4.1), and then project VQ onto \mathcal{U} to obtain U . This is done by solving k (independent) linear assignment problems via the Auction algorithm [12, 9]. Moreover, similarly to existing approaches (e.g. [73, 40]), we prune bad matchings. To this end, we define a threshold $\theta \geq 0$ and re-

move all multi-matchings in U where $VQ \odot U$ is smaller than θ , for \odot denoting the Hadamard product. In order to ensure that $U\mathbf{1}_d = \mathbf{1}_m$, for each individual matching that is removed from a column of U , we add a new column to U that contains all zeros apart from a single element being one—as such, this feature is now matched to its own universe feature (in the clustering perspective, it is a cluster comprising a single element, cf. Fig. 1).

4.4. Algorithm

We call the overall synchronisation procedure NMF-SYNC, which is summarised in Algorithm 1. NMF-SYNC comprises the following main steps: (i) initialisation of V and H (Sec. 4.2), (ii) minimisation of Problem (5), (iii) projection of V onto \mathcal{U} to obtain $U \in \mathcal{U}$ (Sec. 4.3), and (iv) computation of the synchronised $W^{\text{sync}} = UU^T$.

5. Experiments

In this section we evaluate the robustness of NMF-SYNC and compare it against existing permutation synchronisation approaches. To be more specific, we consider the SPECTRAL method [44], as implemented by the authors of [73] to handle partial matchings based on a greedy rounding procedure, the MATCHEIG method [40], and the MATCHALS method [73]. In our experiments we first consider synthetic data in a wide range of different configurations, followed by experiments on real data. We

Algorithm 1: NMF_{SYNC}

Input: $W \in \mathbb{R}^{m \times n}$, d, θ
Output: synchronised W^{sync}
// find best rank- d approximation of W (spectral method [44, 40])

- 1 $[X, \Lambda] \leftarrow \text{eig}(W, d)$, $X \leftarrow X\Lambda^{0.5}$
// initialise according to Secs. 4.1 and 4.2
- 2 $Q \leftarrow \text{SBRA}(X)$, $V \leftarrow (XQ)_+$, $H \leftarrow V^T$
- 3 **Repeat**
 - // multiplicative updates of NMF [10], $\epsilon > 0$ is a small number (numerics)
 - 4 $H \leftarrow H \odot ((V^T W) \oslash ((V^T V)H + \epsilon))$ // \oslash is element-wise division
 - 5 $V \leftarrow V \odot ((WH^T) \oslash (W(HH^T) + \epsilon))$
 - // normalise so that the columns of V and H^T have the same ℓ_2 -norms
- 6 $T \leftarrow \text{diag}(\mathbf{1}_m^T (V \odot V))^{0.5}$, $V \leftarrow VT^{-1}$, $H = TH$
// project onto \mathcal{U} according to Sec. 4.3
- 7 $Q \leftarrow \text{SBRA}(V)$
- 8 $U \leftarrow \text{proj}_{\mathcal{U}}(VQ)$ // project VQ onto \mathcal{U} by solving k independent LAPs
- 9 $U \leftarrow \text{prune}(VQ, U, \theta)$ // prune uncertain matchings
// compute synchronised W
- 10 $W^{\text{sync}} \leftarrow UU^T$

quantify the consistency of the pairwise matchings using the *cycle-error*

$$e_{\text{cycle}}(W) = \frac{1}{k^3} \sum_{i,j,\ell \in [k]} \|(P_{i\ell})_{R_{i\ell}} \cdot (P_{\ell j})_{:,C_{\ell j}} - (P_{ij})_{R_{ij},C_{ij}}\|_F, \quad (7)$$

where for $i, j \in [k]$, the sets $R_{ij} \subseteq [m_i]$ and $C_{ij} \subseteq [m_j]$ denote the indices of non-zero rows and columns of P_{ij} , respectively. We use the ground truth error e_{gt} (*gt-error*) to measure the discrepancy between a given W and the ground truth pairwise matchings W_{gt} , which we define as $e_{\text{gt}}(W) = \|W - W_{\text{gt}}\|_F$. The *f-score* = $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ summarises the *precision* and *recall*.

5.1. Synthetic Data

For our synthetic data experiments we generate the pairwise matchings W for a given number of objects k , the universe size d , the observation rate ρ , and the error rate σ as follows: For each $i \in [k]$, we first sample a random (full) permutation matrix $P_i \in \mathbb{P}_d$. To obtain a *partial* permutation, we remove each row of P_i with probability $1 - \rho$. As such, the number m_i is implicitly determined by ρ , where the average of the m_i is $\bar{m} = \rho d$. Eventually, the ground truth matrix of cycle-consistent matchings is obtained as $W_{\text{gt}} = [P_{ij}]_{i,j \in [k]} = [P_i P_j^T]_{i,j \in [k]}$. We obtain the noisy matrix of pairwise matchings W by perturbing each block P_{ij} of W_{gt} individually by randomly selecting a proportion of σ of the rows of P_{ij} , and then shuffle the selected rows. Note that we perturb

W_{gt} in a symmetric fashion. For each evaluated configuration, we draw 100 samples of W and report the averaged results.

Sensitivity Analysis: In Fig. 5 we present results of our sensitivity analysis with respect to the choice of the threshold parameter θ , as well as to the choice of the estimate of the universe size d that is used as additional input to all the methods. For a wide range of thresholds θ our method results in a smaller *gt-error* compared to the other methods while providing *cycle-consistent* results. Moreover, our method outperforms the other methods for varying universe sizes d .

Comparison to Other Methods: The results of this experiments are shown in Fig. 6, where the rows show the *cycle-error*, the *gt-error*, the *f-score*, and the number of matchings (*#matchings*); and the columns show four different evaluation scenarios where in each scenario a different parameter varies along the horizontal axis. While MATCH_{EIG} and MATCH_{ALS} generally result in a non-zero *cycle-error*, i.e. the matchings are *not cycle-consistent*, the NMF_{SYNC} method guarantees cycle-consistent matchings. It can be seen that the overall result quality of NMF_{SYNC} is superior compared to the other methods.

5.2. Real Data

In our second set of experiments we consider real-world matching problems based on the GRAFFITI [41], EPFL [53] and the MIDDLEBURY [49] datasets, all of which come with ground truth registrations. Our evaluations are based on the well-established protocol of [73], which was for example also used in [59]. To obtain the pairwise matchings W , we first extract SIFT features [38] from the images, and then obtain the pairwise matchings based on simple nearest neighbour matching. Then, we use the so-obtained pairwise matchings as input to the synchronisation methods. We consider the *fraction of correct matchings* (FCM), which indicates the fraction of matchings that have an error less than a specified threshold. Since the true number of correct matchings is unknown (cf. [59]), the FCM is computed relative to the number of image features, as done in [73].

Results: In Fig. 7 we show quantitative results. The first three rows show the FCM for the individual problem instances #1 to #18, where the solid lines indicate cycle-consistent results (NMF_{SYNC}) and the dashed lines indicate cycle-inconsistent matchings (all other methods, with the exception of SPECTRAL in a few instances). Note that the multi-image matching problems in the GRAFFITI dataset are easier compared to the EPFL and MIDDLEBURY datasets, as the overlap of the visible object parts in the GRAFFITI images are much larger. This also

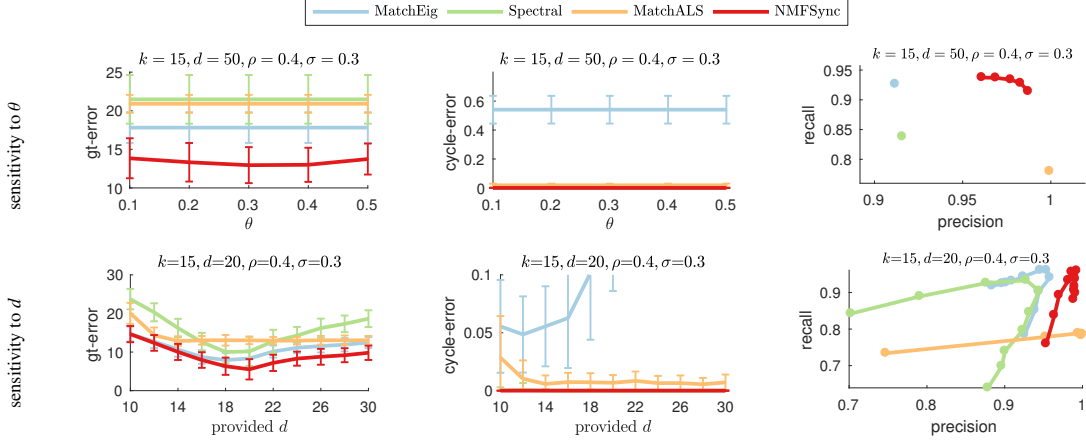


Figure 5: Analysis of the sensitivity of our method to the choice of the threshold parameter $\theta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ (top row), and the sensitivity of all methods to the provided universe size $d \in \{10, 12, \dots, 30\}$ (bottom row, in this case the true d is 20).

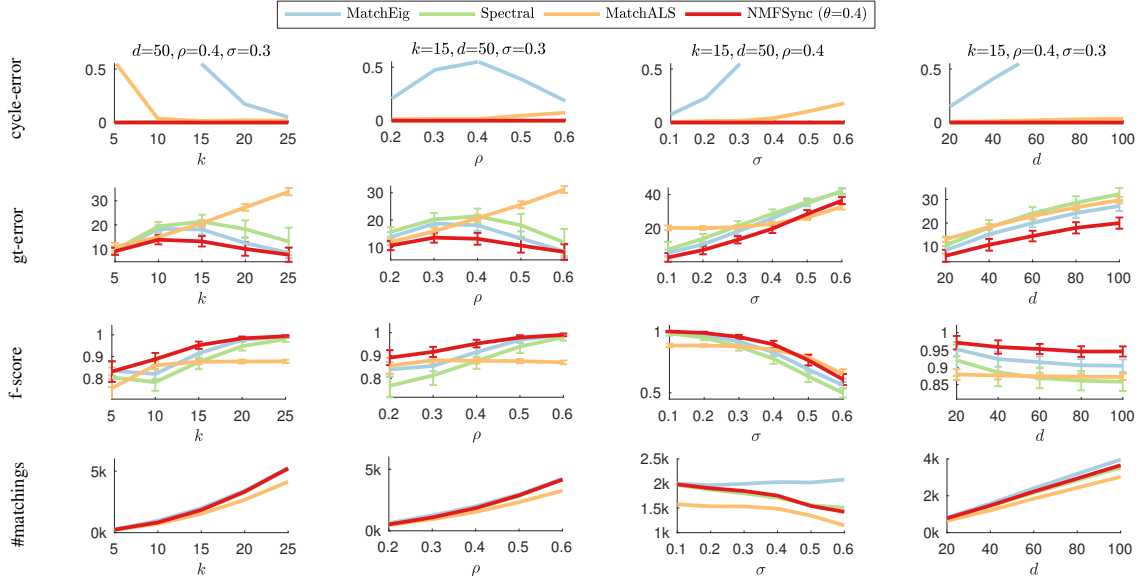


Figure 6: Quantitative results for synthetic data for different varying parameters on the horizontal axis (the number of objects k , the observation rate ρ , the error rate σ , and the universe size d). The *cycle-error* of NMFsync is always 0. For the synthetic data experiments, the *cycle-error* of SPECTRAL is also 0. Considering the *gt-error* (or analogously the *f-score*) and the *cycle-error*, NMFsync is clearly superior compared to its competitors.

explains that the values of the (relative) FCM scores in the other two datasets are lower (in this case the number of features in an image is an overly conservative upper bound for the true number of matchings). Considering the FCM and cycle-consistency, NMFsync clearly outperforms the other methods. For the moderately-sized problem instances #1 to #16, where m is between 372 and 12,238, all methods have comparable runtimes, with the exception of MATCHALS being substantially slower. Note that MATCHALS cannot be used for processing the very large instances #17 and #18 due to its unscalability in terms of memory (cf. Sec. 5.3).

5.3. Discussion & Limitations

Due to the pruning of uncertain matchings in NMFsync based on θ (Sec. 4.3), the total number of obtained matchings of NMFsync varies depending on the input quality. For example, the third column in Fig. 6 illustrates that when increasing the error rate while keeping other parameters fixed, the number of matchings returned by NMFsync decreases. This reflects that our method implicitly takes into account the larger input corruption. Note that other methods also prune uncertain matches.

While the Auction algorithm [12] for solving the LAP has (roughly) cubic *worst-case* complexity [11],

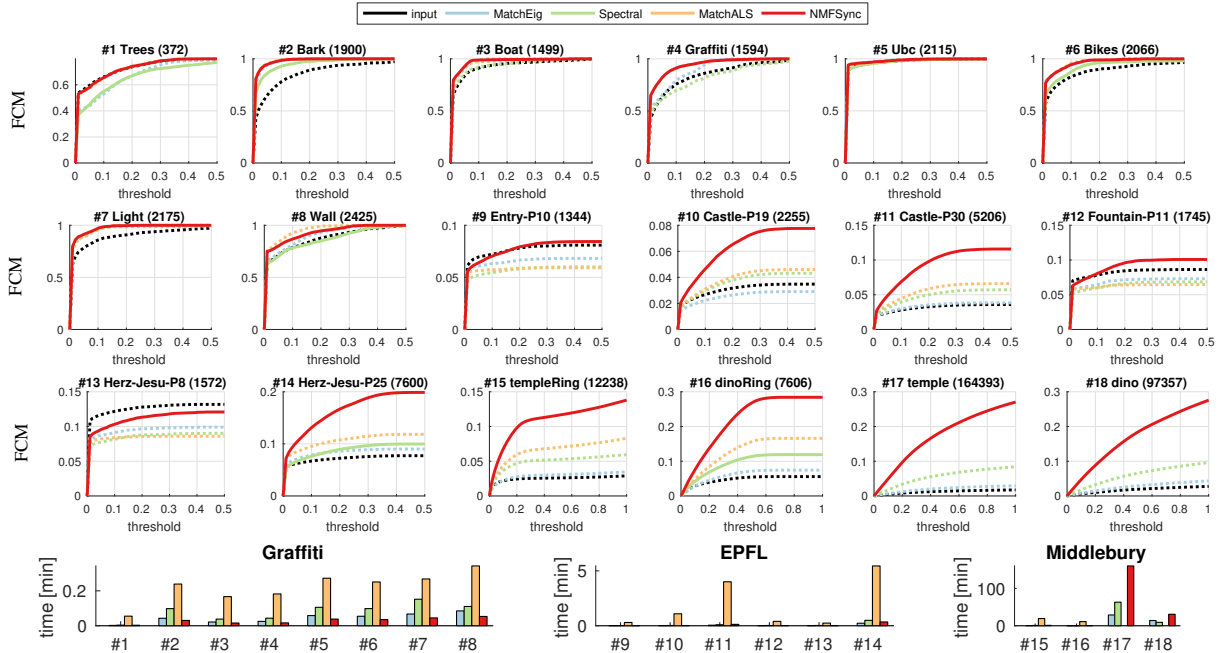


Figure 7: Results for the datasets GRAFFITI (#1–#8, $\theta=0.4$), EPFL (#9–#14, $\theta=0$), and MIDDLEBURY (#15–#18, $\theta=0$). Each plot shows the fraction of correct matchings (FCM) that have an error smaller than or equal to the threshold on the vertical axis (relative to the largest image dimension). The solid lines show results that are cycle-consistent, whereas the dashed lines show results that do not exhibit cycle-consistency. The title of each plot shows the size of the pairwise matching matrix m in parentheses. Considering FCM and cycle-error at the same time, NMF SYNC is superior compared to other approaches.

the analysis in [48] suggests that the *average* complexity is in the regime $O(d^2 \log d)$. Our rotation scheme involves the computation of an SVD with complexity $O(d^3)$. Both, the LAP and the SVD are solved $O(k)$ times. We have observed that the spectral decomposition and the NMF algorithm, with *per-iteration* complexity $O(m^2 d)$, usually dominate the overall runtime. In contrast to MARCHALS, our method never requires the computation of the dense and large $m \times m$ matrix VH (cf. Alg. 1), such that NMF SYNC is much more memory efficient. With that, our method is able to handle very large problem instances, as we show in Fig. 7 for instances #17 and #18, where m goes up to $\approx 160,000$.

One property that is common to all existing synchronisation methods is that they only consider given (partial) matchings without explicitly incorporating any *higher-order information* (such as distances between pairs of features positions). While in certain applications ignoring higher-order information is desirable (e.g. when it is simply not available), in other cases such information could be leveraged to obtain more reliable matchings. Hence, albeit being computationally challenging, we believe that the incorporation of higher-order terms (e.g. in the spirit of the QAP) into synchronisation problem formulations is an interesting direction

for future work.

6. Conclusions

Based on a non-negative factorisation of the matrix of pairwise matchings, we have presented the NMF SYNC method for the synchronisation of partial permutation matrices. We have found that even though the ground truth pairwise matching matrix W is symmetric, from a computational perspective it is actually beneficial to perform an unsymmetric factorisation (cf. Fig. 2). In order to deal with the non-convexity of our formulation, we have proposed a novel scheme for rotating the solution of the spectral relaxation such that it provides a suitable initialisation for the NMF. Moreover, we have generalised the projection-after-rotation approach of the SPECTRAL method [44], so that it can handle *partial* matchings (Sec. 4.3). In contrast to the MARCHALS method [73], and the more recent MATCHEIG method [40], our approach is guaranteed to produce a cycle-consistent solution. Since cycle-consistency is an intrinsic property of the (unknown) true matchings, we argue that it is important to achieve. Furthermore, we have demonstrated that NMF SYNC is comparable to existing methods in terms of scalability, and that it quantitatively outperforms existing approaches on various

datasets. Due to these favorable properties, we believe that NMF SYNC is a significant contribution towards the (sub)field of (partial) permutation synchronisation.

Acknowledgements: This work was funded by the ERC Starting Grant CapReal (335545), the ERC Consolidator Grant 4DRepLy (770784), and by the Luxembourg National Research Fund (FNR, C14/BM/8231540).

References

References

- [1] Aflalo, Y., Bronstein, A., Kimmel, R., 2015. On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences* 112 (10), 2942–2947.
- [2] Arrigoni, F., Maset, E., Fusiello, A., 2017. Synchronization in the Symmetric Inverse Semigroup. In: *ICIAP*. pp. 70–81.
- [3] Arrigoni, F., Rossi, B., Fusiello, A., 2016. Spectral Synchronization of Multiple Views in SE(3). *SIAM J Imaging Sciences* 9 (4), 1963–1990.
- [4] Bandeira, A. S., Charikar, M., Singer, A., Zhu, A., 2014. Multireference alignment using semidefinite programming. In: *Proceedings of the 5th conference on Innovations in theoretical computer science*. pp. 459–470.
- [5] Bandeira, A. S., Singer, A., Spielman, D. A., 2013. A Cheeger inequality for the graph connection Laplacian. *SIAM J Matrix Analysis and Applications* 34 (4), 1611–1630.
- [6] Bazaraa, M. S., Elshafei, A. N., Mar. 1979. An exact branch-and-bound procedure for the quadratic-assignment problem. *Naval Research Logistics (NRL)* 26 (1), 109–121.
- [7] Bernard, F., Theobalt, C., Moeller, M., 2018. DS*: Tighter Lifting-Free Convex Relaxations for Quadratic Matching Problems. In: *CVPR*. pp. 4310–4319.
- [8] Bernard, F., Thunberg, J., Gemmar, P., Hertel, F., Husch, A., Goncalves, J., 2015. A Solution for Multi-Alignment by Transformation Synchronisation. In: *CVPR*. pp. 2161–2169.
- [9] Bernard, F., Vlassis, N., Gemmar, P., Husch, A., Thunberg, J., Goncalves, J., Hertel, F., 2016. Fast correspondences for statistical shape models of brain structures. In: *Proc. SPIE Medical Imaging*. p. 97840R.
- [10] Berry, M. W., Browne, M., Langville, A. N., Pappas, V. P., Plemmons, R. J., 2007. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis* 52 (1), 155–173.
- [11] Bertsekas, D. P., 1988. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research* 14 (1), 105–123.
- [12] Bertsekas, D. P., 1998. *Network Optimization: Continuous and Discrete Models*. Athena Scientific.
- [13] Burkard, R., Dell’Amico, M., Martello, S., 2009. *Assignment problems*. SIAM.
- [14] Chatterjee, A., Govindu, V. M., 2013. Efficient and robust large-scale rotation averaging. In: *ICCV*. pp. 521–528.
- [15] Chaudhury, K. N., Khoo, Y., Singer, A., 2015. Global registration of multiple point clouds using semidefinite programming. *SIAM J Optimization* 25 (1), 468–501.
- [16] Chen, Y., Guibas, L. J., Huang, Q.-X., 2014. Near-Optimal Joint Object Matching via Convex Relaxation. In: *ICML*. pp. 100–108.
- [17] Cootes, T. F., Taylor, C. J., 1992. Active Shape Models - Smart Snakes. In: *British Machine Vision Conference*. pp. 266–275.
- [18] Cour, T., Srinivasan, P., Shi, J., 2006. Balanced graph matching. In: *NIPS*. pp. 313–320.
- [19] Ding, C. H. Q., 0001, T. L., Jordan, M. I., 2008. Nonnegative Matrix Factorization for Combinatorial Optimization - Spectral Clustering, Graph Matching, and Clique Finding. In: *ICDM*. pp. 183–192.
- [20] Ding, C. H. Q., He, X., 2005. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In: *SDM*. pp. 606–610.
- [21] Dym, N., Maron, H., Lipman, Y., 2017. DS++: A flexible, scalable and provably tight relaxation for matching problems. *ACM Transactions on Graphics (TOG)* 36 (6), 184:1–184:14.
- [22] Fiori, M., Sapiro, G., 2015. On spectral properties for graph matching and graph isomorphism problems. *Information and Inference* 4 (1), 63–76.
- [23] Fogel, F., Jenatton, R., Bach, F., d’Aspremont, A., 2013. Convex relaxations for permutation problems. In: *NIPS*. pp. 1016–1024.
- [24] Govindu, V. M., 2004. Lie-algebraic averaging for globally consistent motion estimation. In: *CVPR*. pp. 684–691.
- [25] Hadani, R., Singer, A., 2011. Representation Theoretic Patterns in Three-Dimensional Cryo-Electron Microscopy II—The Class Averaging Problem. *Foundations of computational mathematics* 11 (5), 589–616.
- [26] Heimann, T., Meinzer, H.-P., 2009. Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis* 13 (4), 543–563.
- [27] Hu, N., Huang, Q., Thibert, B., Guibas, L., 2018. Distributable Consistent Multi-Object Matching. In: *CVPR*. pp. 2463–2471.
- [28] Huang, Q.-X., Guibas, L., Jul. 2013. Consistent shape maps via semidefinite programming. In: *Symposium on Geometry Processing*. pp. 177–186.
- [29] Jiang, B., Tang, J., Ding, C., Luo, B., 2017. Binary Constraint Preserving Graph Matching. In: *CVPR*. pp. 4402–4409.
- [30] Kezurer, I., Kovalsky, S. Z., Basri, R., Lipman, Y., 2015. Tight Relaxation of Quadratic Matching. *Computer Graphics Forum* 34 (5), 115–128.
- [31] Koopmans, T. C., Beckmann, M., Jan. 1957. Assignment Problems and the Location of Economic Activities. *Econometrica* 25 (1), 53–76.
- [32] Lawler, E. L., 1963. The quadratic assignment problem. *Management science* 9 (4), 586–599.
- [33] Lê-Huu, D. K., Paragios, N., 2017. Alternating Direction Graph Matching. In: *CVPR*. pp. 6253–6261.
- [34] Lee, D. D., Seung, H. S., 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401 (6755), 788–791.
- [35] Leordeanu, M., Hebert, M., 2005. A Spectral Technique for Correspondence Problems Using Pairwise Constraints. In: *ICCV*. pp. 1482–1489.
- [36] Li, T., Ding, C. H. Q., 2006. The Relationships Among Various Nonnegative Matrix Factorization Methods for Clustering. In: *ICDM*. pp. 362–371.
- [37] Loiola, E. M., de Abreu, N. M. M., Netto, P. O. B., Hahn, P., Querido, T. M., 2007. A survey for the quadratic assignment problem. *European Journal of Operational Research* 176 (2), 657–690.
- [38] Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60 (2), 91–110.
- [39] Lu, H., Fu, Z., Shu, X., 2014. Non-negative and sparse spectral clustering. *Pattern Recognition* 47 (1), 418–426.
- [40] Maset, E., Arrigoni, F., Fusiello, A., 2017. Practical and Efficient Multi-View Matching. In: *ICCV*. pp. 4578–4586.
- [41] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A.,

- Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L. J., 2005. A Comparison of Affine Region Detectors. *International Journal of Computer Vision* 65 (1-2), 43–72.
- [42] Munkres, J., Mar. 1957. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics* 5 (1), 32–38.
- [43] Nguyen, A., Ben-Chen, M., Welnicka, K., Ye, Y., Guibas, L. J., 2011. An Optimization Approach to Improving Collections of Shape Maps. *Computer Graphics Forum* 30 (5), 1481–1491.
- [44] Pachauri, D., Kondor, R., Singh, V., 2013. Solving the multi-way matching problem by permutation synchronization. In: *NIPS*. pp. 1860–1868.
- [45] Sahni, S., Gonzalez, T., Jul. 1976. P-Complete Approximation Problems. *Journal of the ACM* 23 (3), 555–565.
- [46] Schellewald, C., Schnörr, C., 2005. Probabilistic subgraph matching based on convex relaxation. In: *EMMCVPR*. pp. 171–186.
- [47] Schiavinato, M., Torsello, A., 2017. Synchronization over the birkhoff polytope for multi-graph matching. In: *International Workshop on Graph-Based Representations in Pattern Recognition*. pp. 266–275.
- [48] Schwartz, B. L., 1994. A computational analysis of the Auction algorithm. *European Journal of Operational Research* 74 (1), 161–169.
- [49] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R., 2006. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In: *CVPR*. pp. 519–528.
- [50] Shen, Y., Huang, Q., Srebro, N., Sanghavi, S., 2016. Normalized Spectral Map Synchronization. In: *NIPS*. pp. 4925–4933.
- [51] Shi, X., Ling, H., Hu, W., Xing, J., 2016. Tensor power iteration for multi-graph matching. In: *CVPR*. pp. 5062–5070.
- [52] Singer, A., Shkolnisky, Y., Jun. 2011. Three-Dimensional Structure Determination from Common Lines in Cryo-EM by Eigenvectors and Semidefinite Programming(). *SIAM J Imaging Sciences* 4 (2), 543–572.
- [53] Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U., 2008. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: *CVPR*. pp. 1–8.
- [54] Swoboda, P., Kuske, J., Savchynskyy, B., 2017. A dual ascent framework for lagrangean decomposition of combinatorial problems. In: *CVPR*. pp. 1596–1606.
- [55] Swoboda, P., Rother, C., Alhajja, H. A., Kainmüller, D., Savchynskyy, B., 2017. Study of lagrangean decomposition and dual ascent solvers for graph matching. In: *CVPR*. pp. 1607–1616.
- [56] Thunberg, J., Bernard, F., Gonçalves, J. M., 2017. Distributed methods for synchronization of orthogonal matrices over graphs. *Automatica* 80, 243–252.
- [57] Torresani, L., Kolmogorov, V., 2013. A dual decomposition approach to feature correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2), 259–271.
- [58] Tron, R., Vidal, R., 2014. Distributed 3-d localization of camera sensor networks from 2-d image measurements. *Transactions on Automatic Control* 59 (12), 3325–3340.
- [59] Tron, R., Zhou, X., Esteves, C., Daniilidis, K., 2017. Fast Multi-Image Matching via Density-Based Clustering. In: *ICCV*. pp. 4057–4066.
- [60] Vandaele, A., Gillis, N., Lei, Q., Zhong, K., Dhillon, I., 2016. Efficient and non-convex coordinate descent for symmetric non-negative matrix factorization. *IEEE Transactions on Signal Processing* 64 (21), 5571–5584.
- [61] Wang, L., Singer, A., 2013. Exact and stable recovery of rotations for robust synchronization. *Information and Inference* 2 (2), 145–193.
- [62] Williams, M. L., Wilson, R. C., Hancock, E. R., 1997. Multiple graph matching with Bayesian inference. *Pattern Recognition Letters* 18 (11–13), 1275–1281.
- [63] Yan, J., Cho, M., Zha, H., Yang, X., Chu, S. M., 2016. Multi-graph matching via affinity optimization with graduated consistency regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (6), 1228–1242.
- [64] Yan, J., Ren, Z., Zha, H., Chu, S., 2016. A constrained clustering based approach for matching a collection of feature sets. In: *International Conference on Pattern Recognition (ICPR)*. pp. 3832–3837.
- [65] Yan, J., Tian, Y., Zha, H., Yang, X., Zhang, Y., Chu, S. M., 2013. Joint Optimization for Consistent Multiple Graph Matching. In: *ICCV*. pp. 1649–1656.
- [66] Yan, J., Wang, J., Zha, H., Yang, X., Chu, S., 2015. Consistency-driven alternating optimization for multigraph matching: A unified approach. *IEEE Transactions on Image Processing* 24 (3), 994–1009.
- [67] Yan, J., Xu, H., Zha, H., Yang, X., Liu, H., 2015. A matrix decomposition perspective to multiple graph matching. In: *ICCV*. pp. 199–207.
- [68] Yang, Z., Corander, J., Oja, E., 2016. Low-Rank Doubly Stochastic Matrix Decomposition for Cluster Analysis. *Journal of Machine Learning Research* 17 (187), 1–25.
- [69] Yang, Z., Oja, E., 2012. Clustering by Low-Rank Doubly Stochastic Matrix Decomposition. In: *ICML*. pp. 707–714.
- [70] Zaslavskiy, M., Bach, F., Vert, J.-P., 2009. A Path Following Algorithm for the Graph Matching Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (12), 2227–2242.
- [71] Zhao, Q., Karisch, S. E., Rendl, F., Wolkowicz, H., 1998. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization* 2 (1), 71–109.
- [72] Zhou, F., De la Torre, F., 2016. Factorized Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (9), 1774–1789.
- [73] Zhou, X., Zhu, M., Daniilidis, K., 2015. Multi-image matching via fast alternating minimization. In: *ICCV*. pp. 4032–4040.