Università
Ca'Foscari
Venezia

Scuola Dottorale di Ateneo
Graduate School

Dottorato di ricerca
in Informatica
Ciclo 30
Anno di discussione 2018

Transformation Synchronization with Applications in Computer Vision

SETTORE SCIENTIFICO DISCIPLINARE DI AFFERENZA: INF/01
Tesi di Dottorato di Michele Schiavinato, matricola 810469

Coordinatore del Dottorato	**Tutore del Dottorando**
Prof. Riccardo Focardi	**Prof. Andrea Torsello**

UNIVERSITÀ CA' FOSCARI VENEZIA

DOTTORATO DI RICERCA IN INFORMATICA, 30° CICLO

(A.A. 2014/2015 – 2016/2017)

Transformation Synchronization with Applications in Computer Vision

SETTORE SCIENTIFICO DISCIPLINARE DI AFFERENZA: INF/01

TESI DI DOTTORATO DI MICHELE SCHIAVINATO
MATR. 810469

TUTORE DEL DOTTORANDO
Andrea Torsello

COORDINATORE DEL DOTTORATO
Riccardo Focardi

Author's e-mail: michele.schiavinato@unive.it

Author's address:

Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348465
fax. +39 041 2348419
web: http://www.dais.unive.it

To my family, Andrea Torsello and the whole Computer Vision Group

- Michele Schiavinato

Abstract

In *Machine Learning* community the fundamental recognized problems are certainly related to *classification* and *regression* tasks, in which the way data are represented or analysed plays a crucial role to devise such applications. This research is focused on the study of solutions aimed to discover significant patterns which characterise data in a given domain and consists mainly in the ability of a machine to measure similarity relations between distinct objects, in order to identify a specific category or predict future outcomes. *Pattern Recognition* is the branch of science that treats these specific problems, which is supported by a rich literature of mathematical interpretations and practical approaches.

The *Matching Problem* is related to a well-known methodology whose contribution in this field is very influential, since the determination of correspondences between key substructures in the data can suggest a strong similarity information. In the contexts of *Graph Theory* and *Computer Vision*, such elements are nodes of two graphs or points of two shapes respectively, whose matches can yield to the intermediate transformation which aligns an object with respect to each other. Although, this is a task very difficult to solve generally, both for the complexity to compute the exact solution and for the presence of noise in real data. However, these limits can be overcome by formulating approximations of the primal infeasible models, which still provide sufficiently discriminative power for reasonable practical applications.

Furthermore, the matching problem has been always and mainly studied as a pairwise inference, but recently there is growing the interest to generalize this task in multiple setting too. Here, we are interested to solve a set of transformations between all the possible couples of objects in a given dataset, which has to support a global consistency criterion that leads the estimation towards the final transitive solution. The latter is a new fundamental notion in addition of the classic pairwise matching problem, which can be interpreted as a form of transformation synchronization. The main advantage consists to enforce the accurateness of the pairwise solution exploiting of the overall information in the set, reducing wrong correspondences and increasing the tolerance of the process with noisy data.

In this thesis we present novel works that exploit of synchronization to solve problems of Multi-Graph and Multi-Point Set Matching. As concern the former, we formulated the solution according different transformation domains proposing three specific approaches: in our first work, we treated orthogonal permutation matrices among graphs, creating a framework which is independent with respect to the initial solution from an external

Graph Matching algorithm, resulting to an off-line synchronization; in our second work, we formulated permutations as double-stochastic matrices in Birkhoff's Polytope, realizing a tool which can be integrated in any Graph Matching scheme in such space and that operates pushing actively the synchronized solution to the vertexes of the polytope during the learning; in our third work, we generalized the subgraph matching problem with numerous graphs of different size, we enforced consistency deriving partial permutations on the multi-simplex space that embraces the common universe of nodes. As concern the latter, we treated planar homographic transformations between 2D images, devising an optimization process which enforces consistency of the point correspondences and simultaneously learns a classifier to detect the plane contained in each view.

Sommario

Nella comunità scientifica del *Machine Learning* i fondamentali problemi riconosciuti sono sicuramente quelli rivolti ai compiti di *classificazione* e *regressione*, in cui il modo di rappresentate e analizzare le informazioni gioca un ruolo chiave per la loro risoluzione. La ricerca si concentra nello studio di soluzioni in grado di scoprire *pattern* significativi sui dati definiti in un certo dominio ed è indirizzata principalmente alla capacità di una macchina nel misurare aspetti di somiglianza tra oggetti, con lo scopo di identificare la loro appartenenza a una categoria o restituire un valore a cui dipendono. Il campo del *Pattern Recognition* è proprio indirizzato a studiare queste problematiche e nella letteratura scientifica si trova un vasto repertorio di formulazioni matematiche e relativi sviluppi pratici.

Il ben conosciuto *Matching Problem* si riferisce a una metodologia il cui contributo in questo campo è davvero influente, dal momento che il problema di determinare corrispondenze tra le caratteristiche intrinseche che descrivono gli oggetti possono suggerire significativi aspetti di somiglianza tra i dati. Nei contesti di *Graph Theory* e *Computer Vision*, solitamente si cercano rispettivamente le corrispondenze tra nodi di due grafi oppure punti di due immagini, le quali costituiscono una soluzione atta a trasformare la rappresentazione di un oggetto verso l'altro. Tuttavia questo compito risulta generalmente di rilevante difficoltà, sia per la richiesta computazionale e sia per la ricerca alla soluzione esatta in condizioni di rumore nei dati. Quindi si arginano queste limitazioni proponendo soluzioni approssimative rispetto alle formulazioni di partenza, ma che diano comunque una resa accettabile in termini di precisione nella messa in pratica.

In aggiunta, il problema di matching è sempre stato studiato principalmente come un tipo di analisi orientata alle sole due entità, ma di recente sta emergendo l'interesse nel generalizzare questo approccio disponendo di molteplici istanze. L'interesse verte a stimare un insieme di trasformazioni tra tutte le possibili combinazioni di oggetti facenti parte della collezione, ma a cui si impone un criterio di consistenza globale sulle corrispondenze. Quest'ultimo rappresenta un nuovo e fondamentale principio rispetto al classico approccio a coppie, il quale può essere interpretato come una forma di sincronizzazione delle trasformazioni. Il vantaggio primario consiste nel cercare di migliorare la precisione delle corrispondenze derivate su due entità servendosi però del contributo informativo portato da tutti gli altri oggetti in gioco, perciò riducendo possibili errate supposizioni delle soluzioni e rendendo il processo più tollerante al rumore nei dati.

In questa tesi si presentano dei lavori innovativi che fanno beneficio proprio della

sincronizzazione come paradigma per risolvere problemi di *Graph Matching* e *Point Set Matching* su insiemi di oggetti. Per quanto riguarda la prima categoria, si propongono soluzioni legate al dominio delle trasformazioni distinte però in tre approcci: il primo lavoro tratta matrici di permutazione ortogonali tra grafi, realizzando un framework ch'è indipendente rispetto alla soluzione di partenza non sincronizzata, la quale è ottenuta da un classico algoritmo a coppie di Graph Matching, quindi ne deriva una sincronizzazione di tipo *off-line*; nel secondo lavoro le permutazioni sono trattate come matrici bi-stocastiche appartenenti al politopo di Birkhoff, ottenendo quindi uno strumento che può essere integrato in qualsiasi algoritmo ordinario di Graph Matching e rendendolo capace di stimare attivamente la soluzione sincronizzata presente ai vertici del politopo; il terzo lavoro generalizza ulteriormente la ricerca di corrispondenze in sottografi, costruendo matrici parziali di permutazione sullo spazio del multi simplesso in un universo comune di nodi. Per quanto riguarda la seconda categoria, si propone un lavoro per trasformazioni omografiche planari tra immagini 2D, realizzando un processo di ottimizzazione atto a sostenere la consistenza delle corrispondenze tra punti e simultaneamente stimare un classificatore in grado di stabilire la presenza del piano nelle immagini.

Contents

List of Figures

List of Tables

Notation

Multidimensional Objects

- a: is any generic element (normally can assume nominal, scalar or vectorial form);
- $\mathbf{a} = (a_1, a_2, \ldots, a_m)^T$: is a column vector of m elements (or a $m \times 1$ matrix, with m rows and one column);
- $\mathbf{A} = (a_{ij})$: is a $m \times n$ matrix, with m rows and n columns where a_{ij} is the element in the matrix at the row i and column j (if a_{ij} is even a matrix the structure $\mathbf{A}$ is said *block matrix*);
- $\mathbf{0}$, $\mathbf{1}$: are constant column vectors or matrix whose all elements are respectively the scalars 0 or 1 (the dimensions are explicated by the context);
- $\mathbf{I}_m$: is an identity $m \times m$ matrix composed by all zero values and ones in the main diagonal;
- $\mathbf{e}^h$: is a binary vector where the unique value equals to 1 is the h-th component (the dimension is explicated by the context);.

Sets

- $S = \{s_1, s_2, \ldots, s_m\}$: is a finite set composed by a homogeneous collection of m different objects ($|S| = m$ denotes the *cardinality* of the set as the number of its elements), if $m = 0$ then S is expressed as the empty set $\emptyset = \{\}$), moreover, the style $\mathcal{S}$ refers to a special set such as problem space or domain of values;
- $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$: are the typical infinite numerical sets respectively for natural, integer, rational, real and complex numbers (to denote the subset of all the positive or negative numbers is added the superscript $+$ or $-$);
- $[\alpha, \beta] \subset \mathbb{R}$: is the *closed* interval for all the real numbers between two scalar ends with $\alpha \leq \beta$ (to denote that α is not included in the interval the symbol '[' is replaced with '(' and in the same manner for β where ']' becomes ')', in this case the interval is said *open* in one on both ends and infinite $\pm\infty$ ends are possible).
- $\Delta_m = \{\mathbf{x} \in \mathbb{R}^m \mid \forall i = 1 \ldots m : x_i \geq 0, \sum_{i=1}^{m} x_i = 1\}$: is the *Standard simplex* set of m variables and contains any probability distributions in $\mathbb{R}^m$ (let $\mathbf{x} \in \Delta_m$, if $\mathbf{x} = \mathbf{e}^h$ for some $h = 1 \ldots m$ then $\mathbf{x}$ is a *vertex point* in the boundary of the standard simplex, otherwise is any other point in the interior space).

Operations

- $S \times Q = \{(s, q) \mid s \in S, q \in Q\}$: the *Cartesian product* between the sets S and Q is a non commutative operation which returns a set composed by all the possible $|S||Q|$ ordered pairs from the elements contained in them;
- $S^n = \underbrace{S \times S \times \ldots \times S}_{n \text{ times}}$: is the *Cartesian power* of a set S as the Cartesian product of itself for n times;
- $S^{m \times n}$: the space in which are contained all the possible $m \times n$ matrices whose domain of the values is S;
- $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ (with $n, k \in \mathbb{N}; 0 \leq k \leq n$): the *binomial coefficient* which determines the number of subsets of k samples without repetitions extracted from a population of n different elements;
- $\exp(x) = e^x$: is a compact notation for a scalar exponentiation with the typical Euler's number e;
- $(\mathbf{a})_k = a_k$: is a shortcut notation to denote the k-th element from vector $\mathbf{a}$;
- $(\mathbf{A})_k = \mathbf{a}_k$: is a shortcut notation to denote the k-th column vector from matrix $\mathbf{A}$;
- $\|\mathbf{a}\|_2 = \sqrt{\sum_{k=1}^{m} a_k^2}$: is the *Euclidean norm* (or *length* in the Euclidean space) of a scalar vector of m dimensions (the shorter notation $\|\cdot\|$ can be equally used);
- $\|\mathbf{a} - \mathbf{b}\|_2 = \sqrt{\sum_{k=1}^{m} \Big(a_k - b_k\Big)^2}$: is the *Euclidean distance* between two scalar vectors of both m dimensions (you can observe that the norm of a vector is equal to its distance with respect to the origin, namely $\|\mathbf{a}\| = \|\mathbf{a} - \mathbf{0}\|$;
- $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}$: is the *Frobenius norm* of a $m \times n$ matrix;
- $\mathrm{Tr}(\mathbf{A}) = \sum_{i=1}^{m} a_{ii}$: is the linear *Trace* operator of a $m \times m$ square matrix, which returns the total sum of the elements in the main diagonal;
- $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \mathbf{b} = \sum_{k=1}^{m} a_k b_k = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos(\theta)$: is the *dot product* (also known as *scalar product* or *inner product* in Euclidean space) of two scalar vectors where θ denotes the angle between them;
- $det(\mathbf{A})$: is the determinant of a matrix;

- $\mathbf{A}^T$: the typical *transposition* for a matrix $\mathbf{A} = (a_{ij})$ where $\mathbf{A}^T = (a_{ji})$;
- $\mathrm{vec}(\mathbf{A}) = (a_{1,1}, \ldots, a_{m,1}, a_{1,2}, \ldots, a_{m,2}, \ldots, a_{1,n}, \ldots, a_{m,n})^T$: is the *vectorization* of a $m \times n$ matrix into a column vector;
- $\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$: given the $m \times n$ matrix $\mathbf{A}$ and the $p \times q$ matrix $\mathbf{B}$ the *Kronecker product* between them returns $\mathbf{C} = (\mathbf{C}_{ij})$ as a $m \times n$ block matrix (or an extended $mp \times nq$ matrix) where the element at row i and column j is a $p \times q$ matrix such that $\mathbf{C}_{ij} = a_{ij}\mathbf{B}$.

Any exception of the notation as above is locally explained in this thesis to avoid misunderstanding.

Published Papers

[1] MICHELE SCHIAVINATO, ANDREA GASPARETTO, ANDREA TORSELLO Transitive Assignment Kernels for Structural Classification *Third International Workshop on Similarity-Based Pattern Recognition, (SIMBAD)*, 2015

[2] FILIPPO BERGAMASCO; LUCA COSMO, MICHELE, SCHIAVINATO ANDREA ALBARELLI, ANDREA TORSELLO Dense multi-view homography estimation and plane segmentation *23rd International Conference on Pattern Recognition, (ICPR)*, 2016

[3] MICHELE SCHIAVINATO, ANDREA TORSELLO Synchronization over the Birkhoff Polytope for Multi-Graph Matching *15th International Workshop on Graph-based Representations In Pattern Recognition, (GbRPR)*, 2017

1

Introduction

In 1959, Arthur Lee Samuel in his very well-known survey about the game of checkers introduced for the first time in the history of Computer Science the term "*Machine Learning*" [142], which he defined as the "*field of study that gives computers the ability to learn without being explicitly programmed*". From this essential definition we can derive two fundamental branches in the evolution of the ample set of techniques we know nowadays. The general concept of "learning" by a machine is strictly related to the prospect that such ability may be totally autonomous or instead conditioned to follow some predetermined boundary. In the first class we relate to the well-known artificial neural network paradigm, where structures similar to human brain can potentially learn something without some external supervision. Despite the intensive efforts have been done to reach this goal, due to both the computational complexity and the unclear to describe precisely the dynamics in such learning scheme, the greater useful contribution has been given by applications reside in the second class, *i.e.*, solutions strictly conceived inside a specific problem space.

Indeed, concrete works in Machine Learning rely on foundations of Statistics and Maths through abstractions of real world problems. The majority of applications in this field are related to the well-known branch of *Pattern Recognition*. Here, the fundamental point of interest is focused on *data*, since the only way to transfer new *knowledge* in a machine consists to extract information from observations of some specific reality. The *learning* is just the ability to discover this information, which can be interpreted as the presence of recurring *patterns* or *regularities* in rough data. Obviously, this fundamental aspect is critically bound by the specific context of application, because according the origin nature whence are retrieved the samples, there are derived different types of patterns, as for instance, sound *versus* image. Therefore, it is necessary to define a reference *decisional* or *functional* model that is assumed to describe properly the nature of a particular *training set*. Afterwards the machine has acquired sufficiently information, then it learned and its abstracted representation of acquired knowledge is expressed as the set of fixed parameters for the reference model. Finally, the *validation* consists just asking to the machine what is the most likely learned pattern which recognises in a new unknown testing observation: the performance yields in the *accuracy* of its answer. Fundamental branches of problems in Pattern Recognition are surely related to *classification* and *regression*, which are aimed to define a model that predicts a categorical class or a real-valued signal dependent to the input instance. Although, we can generalize this

matter just considering essentially what may mean the presence of recurrent patterns in data. Without loss of generality, when two observations contain common features it is reasonable to assume they share similar properties, an aspect that should yield to assign analogous answers in the predictive model as well.

Data mining refers to another ample field of Machine Learning in which are included diversified techniques to infer salient patterns from the samples. The crucial efforts in this area reside on the study of the special real world nature, with particular focus to the data representation and the formal analysis of its structural domain. After that, machines can manage information relying on predetermined paradigms, which can be inspired from the most simple set of numbers to more complex structures, such as vectors or matrices and further combinations of the latter. Therefore, the salient features of real objects that could be relevant for a learning task have to be firstly measured and secondly organized in a suitable data structure. Even if these aspects may appear obvious and even uncorrelated by the concrete mining process, actually they involve heavily both the correctness of the acquired knowledge and the scheming of the reference model.

Measurements are often affected by bias due to the instrument sensitivity as well as errors in post-processing step of more complex devices. After that, the rooted assumption to devise robust learning strategies consists just that data contain always noise and machines should be able to deal with it autonomously. Nevertheless, this problem could be reduced increasing the resolution of the measurements, but such solution yields to the well-known *Curse of dimensionality* phenomenon [11]. Actually, the salient information necessary to acquire good knowledge is contained in far less features with respect to all those available. This is fundamental, in particular to speed up algorithms even operating on huge data set, but on the other hand, it suggests that enlarging the precision of the measure cannot always reduce the presence of noise.

Data structure has some determined properties, which can be expressed such as the domain of values, magnitude, scale, functional dependences, ordering and more articulate relations among the measures. Therefore, with the purpose to retrieve significant patterns, it is necessary to study the meaning of these properties with respect to the nature of the object which describe. In more technical terms, we have to establish that kind of knowledge to retrieve and those operations are required to perform a *comparison*. For example, we need to measure the distance between two 3D points represented as two real-valued vectors in $\mathbb{R}^3$. Trivially, exploiting of the spatial relation of the features in vectorial Euclidean space, there is not ambiguity to consider the Euclidean norm to solve this problem. After that, Pattern Recognition field is dominated by techniques which rely on the *vectorial paradigm*, just due to its relative simplicity to manage and discriminate data in this form.

Greater complexity emerges in case the objects of interest are represented by data structures with further arrangement of the features, *e.g.*, the classical rectangular representations or more generally feature collections, which are vastly employed to describe 2D images, 3D point clouds, graphs, metabolic networks and so on. Intuitively, one could divert the problem just devising an approach to project non-vectorial data against the vectorial paradigm and exploiting of ordinary techniques thus to derive a proper solution.

Unfortunately, this strategy is very difficult in general, since an injective mapping between such data spaces is not directly derivable, which is a problem typically in NP-hard class [145]. We may give two examples to highlight such difficulty: considering a 2D picture which is unrolled as a trivial sequence of colours, the visual regions of the original image are not spatially recognizable anymore; second, a data structure can be affected by *isomorphism*, which means that can describe with several different representations the same object, *e.g.*, the case of isomorphic graphs.

Sure enough, we can realize that the essential key issue is about the *ordering* or *indexing* of the features inside own data structure. Therefore, we need to deal with this problem looking for another interpretation of the content. Simply, since similar features can take arbitrary locations in own data representation, we may consider as salient patterns the *correspondences* between features of two distinct objects. In other terms, we need to learn how to "realign" (like a 15-puzzle game) the data representation of an object with respect to each other in order to enhance structural similarity: this special task is well-known as *Matching Problem*.

In this thesis we treat problems of matching in two specific contexts of Machine Learning, which are *Graph Theory* [85] and *Computer Vision* [8].

In the former context, we consider the analysis of general graph-based structure, which is expressed as a set of interconnected nodes with possibly further informative attributes both to nodes and edges. The related approach is well-known as *Graph Matching* [35, 45], where the goal consists to derive the binary mapping between the nodes of two graphs, that is a combinatorial problem. This strategy in Graph Theory is considered the most preferable way to retrieve similarity information, since takes in account both nodes and topological aspects in the substructures. Graph Matching yields to the well-known (sub)graph isomorphism, which emerges by the absence of ordering of nodes in the graph-based representation, whose exact recognition is proofed to be NP-complete [51].

In the latter context, our objects of interest refer to visual instances, which can be expressed by images, surfaces or more generally point clouds. Here, the correspondence estimation yields to the well-known problem of *Point Set Matching* [29, 135] or *Point Set Registration*, whose goal consists to infer a rigid or affine transformation that brings the points of an image to those of the other one. The fundamental aspect of this process is related on its real-valued domain, which relies on properties of Euclidean or Affine spaces. The general approach consists to extract subset of salient keypoints between the images whence estimating the optimization parameters of the reference model by minimizing the squared distance of the registered points with respect to a landmark subset. In general, this problem can be solved in polynomial time, since it can be treated as a classical least squares problem.

The majority of solutions in Machine Learning, as more generally in Computer Science, are diversified by their orders of complexity, whose greater interest relates to the fundamental distinction between *polynomial* and *non-polynomial* problems. The crucial point consists in the applicability of these methods in real world scenario, *i.e.*, the *feasibility* in terms of computational time required to solve the given tasks. Unfortunately,

there are many important problems whose exact solutions can not be obtained by polynomial algorithms, *e.g.*, graph (sub)isomorphism. Typically, these processes are modelled as *optimization problems*, in which the desirable solution is evaluated by a score *objective function* over the reference model parameters. Therefore, the main goal consists to maximize (minimize) the objective, but the optimal solution, *i.e.*, its *global* maximum (minimum), could require very effort to be solved. Although, this limit can be overcome by establishing a trade-off, which consists to focus the research in alternative solutions with the purpose to save computational time. Here, infeasible optimization problems can be relaxed by further approximate formulations, reducing in general the problem space and shifting the research towards *local* maximum (minimum). Fundamental contribution to treat these subjects is related to the well-known *Dynamical Programming* [11] field. The matching problems discussed in this thesis are just carried according such approaches.

Graph Matching and Registration may appear two very diversified applications in Machine Learning, but actually share both the fundamental purpose to estimate correspondences between data structures, in particular, representations of real world objects. Clearly, this application is encoded according different schemes, *i.e.*, matching between *nodes* (combinatorial) w.r.t. matching between *points* (geometrical), but the solution can be considered just a more general data "*transformation*" or "*alignment*" of two objects. Correspondences on hand, we can reorganise the features in the data structures to encompass a common structural domain, hence getting the representations being potentially comparable with classical distance metrics. Between the various benefits derived from this approach, we find the fundamental requirement to design desirable positive semidefinite kernels, which are functions that quantifies the similarity between two data entities. Indeed, kernel methods can be exploited in order to perform classification tasks even from non-vectorial data.

Furthermore, there is another interesting aspect that characterizes this kind of analysis, which is the simple fact that the inference aimed to estimate the final transformation relies exclusively to a couple of instances only. This thing is quite obvious, because a matching problem is in general a pairwise process, but since data contains noise for assumption its integrity may affect deeply the accuracy of the overall learning task. Unfortunately, it is not always possible to devise robust methodologies just trusting on this limited set of information as well. On other hand, matching techniques can be very useful tools to deal with noisy data as nestled processes inside more complex applications. After that, finding correspondences states strong data relations, which could suggest to further specialized solvers what elements to discard in the estimation of their reliable knowledge. In such vast field of techniques, this methodology is well-known as *Inlier Selection* [44], wherein the main goal consists to detect the best set of instances to fit a model filtering potential outliers in raw data.

Recently, in several contexts of Machine Learning is increasingly rising a new approach to overcome just this critical drawback, which consists in a generalization of the matching problem exploiting of multiple instances, *i.e.*, more than two objects. Indeed, there exist specific applications where for example sensors or cameras can take numer-

ous observations of a same reference object from different point of view, therefore an extension of this problem is reasonable in practical scenarios. The underlying principle in this novel scheme consists just to exploit of additional information brought from similar objects to enforce those uncertain correspondences due to noise in the pairwise analysis. In other terms, the local bias is spread globally over all the possible pairwise transformations, which should even the quality of the estimated solutions. This approach is very challenging, but introduces further complexity to the matching problem. We describe just the simple case with three objects x_a, x_b, x_c and all possible combinations of alignments as $\pi_{ab}, \pi_{ac}, \pi_{bc}$. The fundamental condition which must be supported is clearly that if the result of the transformation from x_a to x_c is the following estimated object $\overline{x}_c = \pi_{ac}(x_a)$ then $(\pi_{bc} \circ \pi_{ab})(x_a) = \overline{x}_c$. In other terms, the alignments has to be *transitive* and such status can be interpreted as the global *transformation synchronization* [13, 123]. Unfortunately, this consistency constraint is not in general guaranteed by classical pairwise solvers, but it has to be imported in multi-matching problems both to keep coherency of the transformations and to realize positive semidefinite kernels. The main contribution in this thesis regards several techniques aimed to synchronize transformations for multi-way matching, whose principal applications are addressed to generic graph and visual representations.

The contents in this work are organized in three fundamental macro-parts: for each one, we give a brief introduction presenting the fundamental topics and motivations.

I. **Related Work.** In the first part of this thesis we give an overall and detailed presentation of common theoretical concepts related to the works treated in this text. The main areas of the science covered are about Machine Learning, Graph Theory and Computer Vision. Clearly, we focus our attention to introduce just those notions which are relevant to an exhaustive comprehension for the reader, without considering the overall theoretic background which would deserve such fields. Moreover, the whole explanation is addressed to review in deep the general literature of all the well-known methodologies and concepts presented, with the purpose to highlight the current state-of-the-art in these related works. The *Chapter 2* is aimed to introduce our reference data structures underlining their canonical representations and the properties of the spaces where are defined. In particular, we cover the topic of Kernel Methods, presenting advantages, limits and some famous models. Furthermore, we give a general introduction of Computer Vision, presenting the characteristics of the Camera Model and some fundamental applications on images, shape or point clouds. The *Chapter 3* contains an exhaustive introduction about the well-known Matching Problem, which is divided in two distinct scenarios: the classical pairwise setting, where the inference task relates totally just on the pair of two data objects; the generalization of the matching problem, where the inference task involves a collection of data objects beyond the base case. The matching problem in the first part of this chapter is treated in the pairwise setting presenting two particular applications: first, it is related to consider combinatorial problems with

data graphs (*i.e.*, the Graph Matching problem), presenting well-known approximations which relax the associated isomorphism notion; second, it is related to solve matching problems on general point clouds (*i.e.*, the Point Set Matching problem), introducing efficient methods to estimate rigid or affine transformations with particular applications addressed to Computer Vision. The matching problem in the second part of this chapter is again treated both for graph and images, but considering the multiple data scenario. In this part we present state-of-the-art techniques that solve the fundamental problem to guarantee global consistency for these two special kinds of transformations.

II. **Transformation Synchronization.** In the second part of this thesis we introduce the overall contribution of our work, which consists in four novel approaches that treat matching problems in multiple data setting, both applied for graph structures and for visual instances with applications in Computer Vision. The first work presented in *Chapter 4* consists in a framework which realizes a transitive assignment kernel [47] applied on graphs with the purpose to solve typical classification tasks. This method performs a synchronization strategy [123] according the initial estimated solution given by some external and independent pairwise Graph Matching solver, therefore represents an off-line process. This approach introduces alignment step according the paradigm of assignment kernels and assumes the assumption of orthogonal permutations. The problem space of the permutations is relaxed in the space of real-valued matrices and the solution estimated iteratively. The formulation of the kernel is based on transitive permutation alignments and due to the global consistency it results to be positive semidefinite as well. The second work presented in *Chapter 5* consists in an optimization methodology to estimate consistent homographic transformations [110, 155, 182] derived from a dense set of 2D views for plane classification. The main assumption consists that all these images contain a common planar subject and an initial inconsistent solution of the homographies is given. This iterative process for each step simultaneously rectifies the affine transformations performing a synchronization [13] and in meanwhile learns a predictor over the pixel to distinguish the regions which contain the plane in the view. The third work presented in *Chapter 6* consists in synchronization process for multi-graph matching based on Birkhoff's Polytope [16]. This work is particularity inspired to the method presented in the Chapter 4, but differs for two fundamental aspects: first, the synchronization is led on-line, which means that can be completely integrated in the operative context of an extended matching algorithm in multiple data setting (similarity as in the previous Chapter 5); second, it does not consider orthogonal permutations, but works in the problem space of doubly-stochastic matrices. The resulting iterative optimization is rooted by the Birkhoff-Von Neumann theorem, whose permutations are replaced by alignments to solve the transitive double stochastic transformations. The fourth and last work in *Chapter 7* introduces a further generalization of multi-graph matching problem allowing to operate with graphs of different size, *i.e.*, solving graph matching as

the multiple data extension of the common pairwise subgraph problem [38, 179]. This work continues to exploit of permutation synchronization, but its formulation introduces a common node dimension to set the multi-simplex space, which can embrace the matching expectations for each possible pair of uneven graphs. This approach is inspired to the previous methods in the Chapters 4 and 6 for different aspects: in the former, there is no dependence with respect to the initial solution which can be imported from any graph matching solver; in the latter, the definition of the alignments is at least one-way stochastic.

III. **Conclusion and Future Work**. In the third part of this thesis we present our final conclusions about the overall work with transformation synchronization problems, describing the salient aspects of the proposed solutions and obtained results. Moreover, we introduce our future intentions in this special field, reasoning about the principal weak points in our current experience and presenting some further new directions.

I

Related Work

2

Preliminaries: Graphs and Computer Vision

In this chapter we present the preliminary and fundamental concepts which spread the various topics treated in this thesis. The rich existent literature from *Machine Learning*, *Graph Theory* and *Computer Vision* communities is supported by a well-established manner to describe certain theoretical notions; despite this fact, we prefer to underline also in this text how we conceive some rooted aspects in such specific areas of the science. The reader could have already a good familiarity with these contents, but we aim to get used him to our language in order to avoid possible misunderstandings of the next and more complex subjects.

In section 2.1 we introduce formally some structural data representations and well-known methods to derive affinities measures, with particular interest to Graph-based structures. Moreover, in section 2.2 we give a brief introduction of Computer Vision, treating the fundamental aspects related to geometrical transformations, camera model and special tasks with images and point clouds.

2.1 Similarity Methods in Structural Data

In *Machine Learning* community the concept of *similarity* or its counterpart *distance* plays a fundamental role in several pattern recognition applications. Surprisingly, the problem to model a reliable measure of affinity between two formal objects can be very complicated in some scenarios. The fundamental implications in this matter are strictly related to the real world nature of the objects and in particular the translation of themselves in a specific formal data structure, which is a required condition for the embedding in a mathematical context. After that, the complexity of any recognition task depends heavily by the special peculiarities of the involved data representations. In particular, the main problem consists in devising a discrimination approach which can be considered *reliable* according the following two aspects: first, the comparison has to take in account the fundamental differences both of the structures and features; second, there most not be ambiguous cases in the manipulation of the data.

In this section we introduce the general data representations in this thesis, which are based on vectorial, matricial and graph paradigms. Moreover, we present in detail several techniques for the definition of reliable measures of affinity in order to discriminate such structures.

2.1.1 Structural Data Representations

In Machine Learning field, the *vectors* and *matrices* are surely milestone data structures extensively employed in the formulation of mathematical problems. They are fundamental tools which can be used as fully or partial descriptors of objects in real world domain. Obviously the crucial point consists to clarify how to analyse the information in these structures with respect to how to import them, but there can be contexts where is not so trivial to look for a one-to-one mapping from the real world objects to vectorial/matricial paradigm. In this thesis the main objects of interest are described as sequence of features, indexed tables, data graphs or combinations of such latter structures. Although, even if these data structures could seem very difficult to generalize, in principle they are all derived just from sets of vectors and matrices defined in a specific domain. The general notation for structural data we apply in our work is summed up as follows.

Vectorial Paradigm

Any real-world object has measurable features, *e.g.*, for an image could be the pixel colour signals or for an animal some physical information (*e.g.*, height, weight, age, sex, and so on). In general, a single feature k is well defined in a proper domain $\mathcal{X}_k$ which may be numerical/quantitative (*e.g.*, height) or nominal/categorical (*e.g.*, sex). Therefore considering the representation of any object according a set of m known features, it is possible to define a common descriptor domain for all the objects of interest, which can be modelled as the feature space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_m$. Hence, the classical data structure which represents an objects is the *observation* or *feature vector* as follows:

$$\mathbf{x} = (x_1, x_2, \ldots, x_m)^T \in \mathcal{X}.$$

An important required condition in sequential data is the lack of dependence among its features, namely, a feature should not be obtainable from other features (*e.g.*, age *versus* birth date). In the practice, typical algorithms are designed to operate with points defined in the mathematical notion of *vector space*, in particular with real and complex numbers, that is $\mathcal{X} = \mathbb{R}^m$ and $\mathcal{X} = \mathbb{C}^m$ respectively. Anyway, nominal feature can be easily mappable to a quantitative finite domain, therefore we consider implied such preprocessing step.

Matricial Paradigm

The vectorial paradigm could not be sufficient to model further data representations in case the indexing of the features would require to be extended. Therefore, adding a new dimension n in the one-dimensional sequence model, the result is just a table of features which introduces two way indexing for each feature x_{ij}, which is named as *cell*. This enlarged structure inherits all the aspects of the vectorial paradigm and it is well-recognized as the following *feature matrix*:

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{pmatrix} \in \mathcal{X}^{m \times n}.$$

There exist many applications of this data structure in Machine Learning, for example to describe a whole dataset of different m object observations with a fixed number of n features, which are arranged per row in the matrix. In Computer Vision, if the feature domain is the colour signal of a 2D image with resolution $m \times n$, each cell can represent a pixel and in particular the indexes are the pixel coordinates. Finally, the matricial paradigm can be generalized even with further dimensions: the multi-dimensional feature matrix. An example is the 3D space to represent volumetric images in $\mathbb{R}^{m \times n \times o}$, the *voxel* (the extension of the cell in 2D tables) represents a 3D pixel, which is indexed by the three common coordinates (x, y, z) in the box of resolution $m \times n \times o$ pixels. In *Advanced Mathematics* exits several applications with exploit of multi-dimensional structures that go beyond the third dimension, even if there is not often a precise correspondence with real world objects anymore.

Graph-based Paradigm

The data structure most important in this context for its rich expressiveness is surely the well-known Graph-based representation of an object (*e.g.*, shape, social network, road map, *etc.*), which can exploit of both vectorial and matricial structures. The graph is described typically as a pair $G_i = (V_i, E_i)$ where $V_i = \{v_k^i\}_{k=1}^n$ is the set of nodes[1] and $E_i \subseteq V_i^2$ the set of edges. The cardinality of these sets are respectively denoted as $n = |V_i|$ and $m = |E_i|$. The topology of G_i can be described by an $n \times n$ matrix derived from the set E_i as well, since the edges between nodes require just two indexes: the pair of labelled nodes. This structure is well-known as *adjacency matrix* $\mathbf{A} = (a_{pq})$, whose entry $a_{pq} > 0$ when $(v_p^i, v_q^i) \in E_i$ or 0 otherwise. Nodes and edges of a graph can have assigned further information, which is typically composed by sets of vectorial data. For instance, if a graph represents a subject of an image, each node can contain the coordinates of the pixel point that represents; moreover, the value of the distance between pair of nodes can

[1] Note that the verbose notation $v_k^i \in V_i$ is just preferential to avoid ambiguities in multiple data references. Although, if the context is sufficiently clear, the related light version $k \in V_i$ is equally accepted.

be assigned as an edge attribute. If the latter is simply a scalar value, the adjacency matrix is a sufficient tool even to incorporate this further information as well, which is treated just as a weight. In general, a graph G_i can be *weighted* or *unweighed* just according the domain of the related adjacency matrix, respectively when $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{A} \in \{0,1\}^{n \times n}$. In addition, if edges have not an orientation, *i.e.*, the graph is *undirected*, the matrix $\mathbf{A}$ is *symmetric* as well, otherwise the graph is *directed*.

In a wide class of applications is necessary to introduce an additional important feature associated to the data, whose concrete meaning depends both from the problem of interest and what learning approach is employed. More specifically, it is sufficient to introduce in deep the idea of *label* or *class*, which limits the domain of such feature to the categorical case. We can express formally the space of k possible labels as $\mathcal{Y} = \{1, 2, \ldots, k\}$, therefore for each i-th structured object the data model is extended as the pair

$$(\mathbf{X}_i, y_i) \in \mathcal{X} \times \mathcal{Y},$$

where y_i is the label associated to a structured object $\mathbf{X}_i$. Such new information y_i is considered as the depended feature with respect to all the others contained in $\mathbf{X}_i$, which is hence an independent component; in other terms the label is assumed to be determinable on the basis of own associated descriptor. This model describes the well-known *classification problem* in machine learning.[2]

2.1.2 Kernel Function and Hilbert Space

An important task to treat in Machine Learning consists to definite a way in order to measure the *affinity* between couples of data objects, which are described for example as vectors or matrices. There are two possible forms or nomenclature to express the same concept from opposite point of views:

- $d(\cdot, \cdot)$ **dissimilarity** or **distance**: measures how much two elements are dissimilar, typically ranges in $[0, +\infty)$;
- $s(\cdot, \cdot)$ **similarity**: measures how much two elements are similar, typically ranges in $[0, 1]$.

In more general terms, the main goal consists in the definition of a function between two data entities which holds some special mathematical properties. This research is largely supported by the theoretical notion of *kernel function*. Formally, let $\mathcal{X}$ be a *vector space*,[3]

[2] The case when holds $\mathcal{Y} \subseteq \mathbb{R}$ determines another class of learning which is the field of *regression problem*; anyway it is skipped since not pertinent with respect to the work in this thesis.

[3] Actually, the space $\mathcal{X}$ could be treated as a more general *input space* of objects, *i.e.*, it is not a constraint of the kernel function definition. Although, we introduce the notion of vector space just to reflect the structural nature of data.

the two-argument real-valued function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *kernel* if for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ holds

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2) \rangle_{\mathcal{V}}$$

for some *inner-product space* $\mathcal{V}$ such that $\forall \mathbf{x} \in \mathcal{X}$ holds $\varphi(\mathbf{x}) \in \mathcal{V}$.

This definition is well-known in Kernel Methods field as the Kernel Trick [145], which allows to get linear learning algorithms to learn non-linear function or decision boundary. Kernel functions support fundamental properties, in particular they are *symmetric* and (semi-)*positive define*. Clearly, the former is due to the symmetry of inner products, while we can explain the latter property from by the definition of Positive (Semi-) Definiteness [160] of a matrix as follows:

- A symmetric real matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive semi-definite (PSD), if for all $\mathbf{x} \in \mathbb{R}^n$ holds $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$. We denote that $\mathbf{A}$ is PSD with $\mathbf{A} \succeq 0$.
- Moreover, $\mathbf{A}$ is (strictly) positive definite, if $\mathbf{A}$ is PSD and $\mathbf{x}^T \mathbf{A} \mathbf{x} = 0$ iff $\mathbf{x} = \mathbf{0}_n$. We denote this variant with $\mathbf{A} \succ 0$.

This condition in matricial calculus can be interpreted by the analogy with positive scalar numbers, in fact, there does not exist a consistent notion of "positive" operating with such structures, which are a special case of symmetric matrices as well. The definition of Negative (Semi-)Definiteness is easily derivable by inverting the inequalities.

Let $X = \{\mathbf{x}_i \in \mathcal{X}\}_{i=1}^N$ be a collection of N objects in the vector space $\mathcal{X}$, the *kernel matrix* (or *Gram matrix*) can be described as the symmetric matrix $\mathbf{K}$ that results from applying the kernel function $\kappa(\cdot, \cdot)$ to all the pairs of data points in set X as follows:

$$\mathbf{K} = \begin{pmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \dots & \kappa(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_N, \mathbf{x}_1) & \dots & \kappa(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}.$$

Therefore, the kernel function $\kappa(\cdot, \cdot)$ is semi-definite just because any its kernel matrix $\mathbf{K}$ is also a semi-definite matrix, *i.e.*, $\mathbf{K} \succeq 0$.

Finally, the crucial aspect in the definition of data kernels onto any structural domain $\mathcal{X}$ is the inner-product vectorial space $\mathcal{V}$, which is in particular vastly required to be a Hilbert space [119] too. This special class of spaces in real and complex domains is a generalization of the common Euclidean space beyond the third dimension.[4] In fact, the fundamental hard problem in this field consists just in the definition of a one-to-one mapping $\phi : \mathcal{X} \to \mathcal{H}$ from a general vectorial space $\mathcal{X}$ of the data to a Hilbert space $\mathcal{H}$, that is the main ingredient to formulate a semi positive kernel.

[4]Typical examples of Hilbert spaces are the interval $[0, 1]$, the real set $\mathbb{R}$, the complex set $\mathbb{C}$ and the set $\mathbb{R}^n$ with $n \in \mathbb{N}$.

Main Structural Kernels

In this section we introduce some well-known kernels in Machine Learning, for vectors, matrices and other more complexed data structures, presenting their fundamental aspects and several solutions in literature.

- **Linear Kernel:** $\kappa_{lin}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^T \mathbf{x}_2 = \sum_{k=1}^{m} (\mathbf{x}_1)_k (\mathbf{x}_2)_k$

  The Linear kernel is just the inner product of two vectors, hence it must be symmetric and positive semidefinite for definition trivially. This function produces results which range arbitrary in $\mathbb{R}$, therefore there is not a direct interpretation of affinity between two entities actually. In fact, the main information given by this kernel is related to orthogonally since $\kappa_{lin}(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1\| \, \|\mathbf{x}_2\| \cos(\theta)$, where θ is the angle between the vectors. From this result is possible to define also the Cosine kernel as follows

  $$\kappa_{cos}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\kappa_{lin}(\mathbf{x}_1, \mathbf{x}_2)}{\|\mathbf{x}_1\| \, \|\mathbf{x}_2\|} = \cos(\theta),$$

  which considers just the cosine of the angle between the vectors. This further function is still a kernel and in particular can give a notion of affinity since ranges in $[-1, 1]$ as well: negative values of the kernel are interpreted as a distance measure, while positive values as a similarity; finally, if $\kappa_{cos}(\mathbf{x}_1, \mathbf{x}_2) = 0$ means that $\mathbf{x}_1$ and $\mathbf{x}_2$ are orthogonal, therefore they are not correlated.

- **Polynomial Kernel:** $\kappa_{pol}(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^T \mathbf{x}_2 + c)^d$

  The Polynomial kernel computes a measure of similarity between two vectors over polynomials of the original variables with order $d > 0$, therefore allowing the learning of non-linear models. The free parameter $c \geq 0$ exploits of the influence of higher-order *versus* lower-order terms in the polynomial produced by the kernel, which is called *homogeneous* in case $c = 0$. This kernel allows to discriminate two vectors not only through a linear dimension of their features (as the linear kernel κ_{lin}), but also by combinations of the ones too. Therefore, this kernel can be an useful tool to work in training data with strong functional dependency among the features, but without requiring to learn all the parameters.

- **RBF Kernel:** $\kappa_{RBF}(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2\sigma^2}\right)$

  The Radial Basis Function (RBF) kernel, or Gaussian kernel, represents a well-known measure of similarity which is formulated from the classical Euclidean distance metric between vectors $d(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_2$ and the Gaussian function with a free scaling parameter $\sigma \in \mathbb{R}$. Indeed, since RBF kernel is just the exponential of always negative number, the outcome decreases with the distance of the two vectors and in particular ranges in $]0, 1]$, *i.e.*, the lower bound is 0 in the limit $\big(d(\mathbf{x}_1, \mathbf{x}_2) = +\infty\big)$ and upper bound is 1 when $\mathbf{x}_1 = \mathbf{x}_2$ $\big(d(\mathbf{x}_1, \mathbf{x}_2) = 0\big)$, which are the typical conditions for a well-formed similarity measure.

- **Graph Kernels**

  In this special class of kernels we have to introduce an ideal input space $\mathcal{G}$ whose elements are data graphs. Therefore, the definition of a Graph kernel consists just in a positive semidefinite function $\kappa_g : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$, which measure distance/similarity between a pair of graphs $G_1, G_2 \in \mathcal{G}$. Nevertheless, in this new context there cannot be a direct formulation of $\kappa_g(\cdot, \cdot)$ as other kernels whose entities reside in vector space $\mathcal{X}$, just because graphs are structures defined in a space $\mathcal{G}$ with totally different mathematical nature. The famous Kernel Trick [145] represents the main guideline to solve this problem, but looking for a proper injection $\varphi : \mathcal{G} \to \mathcal{H}$ (*i.e.*, a complete graph kernel) to embed a graph in a Hilbert space $\mathcal{H}$ is in general a NP-hard problem. This statement is supported by the fact that computing any complete graph kernel is at least as hard as deciding whether two graphs are *isomorphic* and the *graph isomorphism* can be yielded to a NP-hard problem [49]. Indeed, if φ is injective, then

$$\begin{aligned} & \sqrt{\kappa_g(G_1, G_1) - 2\kappa_g(G_1, G_2) + \kappa_g(G_2, G_2)} \\ = \; & \sqrt{\langle \varphi(G_1) - \varphi(G_2), \varphi(G_1) - \varphi(G_2) \rangle_{\mathcal{H}}} \\ = \; & \|\varphi(G_1) - \varphi(G_2)\|_2 \\ = \; & 0 \end{aligned}$$

  if and only if G_1 is isomorphic to G_2.

  For the special case of Graph kernels in this section, we decide to give a picture of their essential story in literature, evaluating pros and cons of the main approaches and solutions.

  Main approaches. The easily strategy to devise a graph kernel is just to define a process which projects graphs to feature vector space. In this way the computation of similarity is clearly addressed by classical distance metrics for vectorial data and thus the related well-known learning techniques can be imported. Unfortunately, in general this task could lead loss of the topological information of the original graphs or it yields to subgraph isomorphism problems.

  A well-known strategy that skips the direct structural representation is based on Graph Edit Distance [50], whose scheme consists to assign costs for edit operations (edge/node insertion/deletion and renaming of the nodes) which transform a graph G_1 to G_2 (see Figure 2.1). Therefore, the affinity measure is obtained as the cumulative cost from the set of operations necessary to complete the graph transformation. Even if this new strategy can describe partial similarity and cope with noise data, it is affected by the hard problem to assign proper cost for the operations and still requires to deal with subgraph isomorphism as verification step.

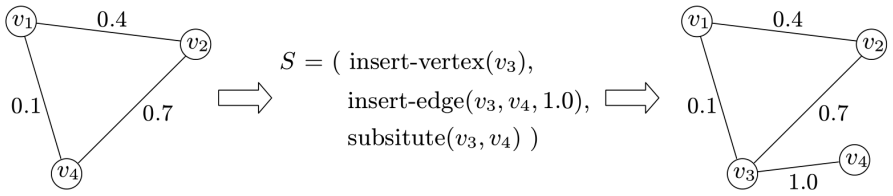


Figure 2.1: *Example of transformation of two graphs as a set S of Edit Distance operations with costs assigned to edges.*

The R-convolution kernels by Haussier [66] represents a new fundamental mathematical abstraction of structural data which can be used in the definition of graph kernels. The general principle is based to define a decomposition relation $\mathcal{R}$ which divides structural objects in ideal parts. For example, let O_1, O_2 be two ideal structural objects and o_i^1, o_i^2 the i-th related parts of such objects in $\mathcal{R}$, then the R-convolution kernel is defined as follows

$$\kappa_{R\text{-}conv}(O_1, O_2) = \sum_{(o_i^1, O_1) \in \mathcal{R}} \sum_{(o_i^2, O_2) \in \mathcal{R}} \kappa_{parts}(o_i^1, o_i^2),$$

where $\kappa_{parts}(\cdot, \cdot)$ is a further kernel function over the parts. In this way, graph kernels can be treated as convolution kernels $\kappa_g(G_1, G_2) = \kappa_{R\text{-}conv}(G_1, G_2)$ over pairs of graphs and each new decomposition relation $\mathcal{R}$ yields in a new graph kernel as well.

Another famous class of graph kernels is based on the Random Walks paradigm [80]. The main principle consists to discriminate two graphs G_1, G_2 counting their number of common random walks. This problem can be solved in a very elegant way constructing a new supporting structure $G_w = (V_w, E_w)$, which is a direct *product graph* between the graphs, therefore the random walks in G_w are just the common walks between G_1 and G_2. The walks on length k are obtained by looking the k-th power of the related adjacency matrix A_w of $n = |V_w|$ nodes, then the resulting Random Walks-based graph kernel is formulated as follows:

$$\kappa_{RW}(G_1, G_2) = \sum_{i,j=1}^{n} \left(\sum_{k=0}^{\infty} \lambda^k \mathbf{A}_w^k \right)_{ij},$$

where $\lambda > 0$ is a free scale parameter of the function. This model is affected by two fundamental drawbacks due to the nature of product graphs and random walks. The former is related to runtime performance operating with G_w, both to perform the direct computation in $O(n^6)$ and the space complexity to store the whole structure. Vishwanathan *et al.* [183] propose a fast computation version just by the factorization $\kappa_{RW}(G_1, G_2) = \mathbf{1}^T(1 - \lambda\mathbf{A})^{-1}\mathbf{1} = \mathbf{1}^T\mathbf{M}\mathbf{1}$, where the $n \times n$ matrix $\mathbf{M}$ is solved by Sylvester equations reducing the polynomial time to $O(n^3)$. Another marginal performance problem consists in the construction of G_w, which is affected by the

Label Enrichment process. Mahé *et al.* [105] reduce the space required to manage product graph by introducing new artificial node labels as topological descriptors based on Morgan index, which counts the k-th order of neighbour nodes. Finally, in this work is solved the well-known phenomenon called *tottering* or *halting* in random walks, in which possible cyclic walks for relative small substructures of the graphs can be unjustifiably measured with very high kernel values. The solution consists to modify the probabilistic model of random walk cutting the path with a repeated node in the form $v_1, \ldots, v_k$ with $v_i = v_{i+2}$.

Following a similar direction to discriminate two graphs according common walks, another well-known strategy is based to replace random walks with shortest paths. Borgwardt and Kriegel [17] propose a solution based by comparing all the pairs of shortest path lengths related the pair of graphs. Let $d(\cdot,\cdot)$ be the ideal distance metric of the shortest path from the node v_i^p to v_j^p in a generic graph G_p and $\kappa_{len}(\cdot,\cdot)$ a further kernel to compare two lengths, the shortest-path kernel is formulated as follows:

$$\kappa_{S\text{-}Path}(G_1, G_2) = \sum_{v_i^1, v_j^1 \in V_1} \sum_{v_k^2, v_l^2 \in V_2} \kappa_{len}\Big(d(v_i^1, v_j^1), d(v_k^2, v_l^2)\Big).$$

The choice of $\kappa_{len}(\cdot,\cdot)$ is in general treated as a Delta kernel $\kappa_\delta(l_1, l_2)$ which is 1 for $l_1 = l_2$ and 0 otherwise or a Linear kernel $\kappa_{lin}(l_1, l_2) = l_1 l_2$, where the latter yields to the product of the Wiener Indexes of the graphs [185]. Clearly, this kernel requires the computation and comparison of all the pairs of shortest paths, which can be solved in polynomial time respectively by Floyd-Warshall $O(n^3)$ and by Wiener Indexes $O(n^4)$. The Shortest-Path-based kernels are in general faster than Random Walks-based kernels and they give a more accurate measure of similarity. Finally, Ramon and Gaerther [131] introduce a kernel based on subtree-based pattern, which is a supporting structure that allows repetition of nodes and edges similarly as the walk/path paradigm. The general idea consists to construct subtree-based patterns G_1^{st}, G_2^{st} related the two graphs G_1, G_2, the measure is obtained comparing the nodes $v_i^1 \in G_1^{st}, v_j^2 \in G_2^{st}$ via a proper kernel function and recursively for all the sets of neighbours of such nodes. There are two common setbacks of all the graph kernels introduced so far: first, the weak ability to catch the topological information of the substructures of the graphs; second, the low scalability operating with big data.

Optimal Assignment kernel [48] is a new class of graph kernels, which can discriminate two graphs topologically by applying a kernel on aligned substructures. Let $\{x_1^i, \ldots, x_{|V_i|}^i\}$ be a set of substructures extracted from a general graph G_i and $\kappa_{sub}(\cdot,\cdot)$ a kernel over them. Moreover, let π be a permutation of the natural number in the set $[1, \min(|V_1|, |V_2|)] \subset \mathbb{N}$ according the dimensions of two graphs G_1, G_2, then the Optimal Assignment kernel is formulated as follows:

$$\kappa_{opt}(G_1, G_2) = \begin{cases} \max_\pi \sum_{i=1}^{|V_1|} \kappa_{sub}\left(x_i^1, x_{\pi(i)}^2\right) & |V_2| \geq |V_1| \\ \max_\pi \sum_{j=1}^{|V_2|} \kappa_{sub}\left(x_{\pi(j)}^1, x_j^2\right) & \text{otherwise.} \end{cases}$$

Neuhaus and Bunke [118] overcome the limit of edit distance and random walk proposing a graph kernel based on a further version which combines both the paradigms. The resulting Edit-Distance kernel can be described as a random walk kernel whose product graph is generated by only pairs of nodes which match with the related edit distance graph built over the couple of graphs. Although, Edit-Distance kernel as well as any other Optimal Assignment based kernel, does not guarantee to be also a positive semidefinite kernel [181], which is a general problem due to the alignment step of the substructures. Menchetti *et al.* [111] propose an interesting new paradigm of graph kernel, which imports structural discrimination following a similar principle of R-convolution kernels. The root idea consists in the definition of a new form of decomposition of a graph G_i that is expressed as the following pair of two special substructures $(s^i, \mathbf{z}^i)$: the *selector*, which is a subgraph s^i of G_i with associated a related kernel $\kappa_\delta(\cdot, \cdot)$; the *contexts* of occurrence of s^i in G_i, which are arranged in a vector $\mathbf{z}^i = (z_1^i, \ldots, z_D^i)$ of D subgraphs of G_i and for each substructure z_d is defined a proper kernel $\kappa_d(\cdot, \cdot)$. An example of decomposition could be when S^i is just a node and $\mathbf{z}^i$ the neighbourhood of S^i in G_i. Let $\mathcal{R}^{-1}(G_i)$ be the space of all possible decompositions of a graph G_i, we define the Weighted Decomposition kernel as follows:

$$\kappa_{WD}(G_1, G_2) = \sum_{\substack{(s^1, \mathbf{z}^1) \in \mathcal{R}^{-1}(G_1) \\ (s^2, \mathbf{z}^2) \in \mathcal{R}^{-1}(G_2)}} \kappa_\delta(s^1, s^2) \sum_{d=1}^{D} \kappa_d\big((\mathbf{z}^1)_d, (\mathbf{z}^2)_d\big).$$

This assignment kernel has the advantage to be positive semidefinite and to take in account of the substructures of two graphs as well, but there is not a reliable strategy to establish non-trivial selectors.

In conclusion, we can observe that defining a graph kernel which discriminates over the topologies consists in a task which yields mainly to a matching problem of substructures, in other terms the Graph Matching (GM) [35, 45]. Therefore, the vertex correspondences derived from any GM algorithms not only can be used to align two graphs, but they represent precious information to define a reliable graph kernel as well.

2.2 Foundations of Computer Vision

Computer Vision [165] is a well-known interdisciplinary field of research in which the main problems are focused on acquiring, processing, analysing, and understanding the specific information derived from images and videos. The underlying motivation in this

area is based on the imitation of the human vision and how to bring this ability in a machine with the purpose to gain the maximum as possible performances. Consequently, the structured new information which is extracted by these methods could be used to perform some decision.

The typical and fundamental tasks which are treated in Computer Vision could be listed in *recognition* of objects in a scene, *motion analysis* of points by changing their positions in sequential frames, *scene recognition* through the reconstruction of objects by combining several shots, and *image restoration* in which an image can be cleaned and enhanced from noise or general damages due to low quality in acquisition. Images, shapes or more general points clouds can be analysed in several ways according the two main classes of 2D and 3D point spaces. In this primary step, the camera model which is used to import data from real world to a structured domain represents a fundamental aspect which involves the implementation of a computer vision process. Typically, there are methodologies which either assume an initial guesses to lead the solution in a reduced problem space or rely just of the essential information extracted from images. Although, we are interested to introduce techniques according the latter case, discarding special conditions of the problem and dealing with just coarse data.

In this section we focus only some general foundations of Computer Vision to give a theoretical background which is necessary to cover the main topics in this thesis. In particular, we are interested to describe in deep the set of notions related to vision schemes, well-known data extraction techniques and matching problems operating in this special field.

2.2.1 Camera Model and Epipolar Geometry

Digital camera represents clearly that hardware device which can translate the visible light from real world as digital information towards a storage support. Actually, this process is divided in a very articulated work-flow of sub problems, but independently from aspects of physics or other pre/post processing stages, the underlying notion that leads such acquisition is the geometrical abstraction considered in the machine. After that, a camera can be interpreted as an ideal mapping from real world space to a mathematical system of coordinates.

First of all, there is required to define formally how the fundamental geometrical transformations are described in Computer Vision (see Figure 2.2). Considering the d-dimensional real-valued space $\mathbb{R}^d$, any related geometrical transformations in such domain can be modelled just as an ideal function $T_\Theta : \mathbb{R}^d \to \mathbb{R}^d$ subject to a set of parameters Θ, which transforms any point $\mathbf{p} \in \mathbb{R}^d$ in another point $\mathbf{p}' = T_\Theta(\mathbf{p})$. Typically, we are interested to study *linear* maps, that are in particular divided in two fundamental groups of transformations with the following salient aspects.

- **Rigid**: a *rigid* or similarity transformation is an application between Euclidean spaces and it has the main property on preserving the distances between every pair of transformed points, therefore visual objects will preserve shape and size. The typical model is $T_\Theta(\mathbf{p}) = s\mathbf{R}\mathbf{p} + \mathbf{t}$, where $\mathbf{R} \in \mathbb{R}^{d\times d}$ is the rotation matrix (an orthogonal matrix, *i.e.*, $\mathbf{R}^T = \mathbf{R}^{-1}$, constrained to $\det(\mathbf{R}) = 1$), $s \in \mathbb{R}^+$ is a scaling factor and $\mathbf{t} \in \mathbb{R}^d$ is the translation vector of the point w.r.t. the origin. Furthermore, the parameters of a rigid transformation are represented by the set $\Theta = \{s, \mathbf{R}, \mathbf{t}\}$. Generally, if there are known scaling, rotation and translation T_Θ is called *similarity transformation*, otherwise *euclidean transformation* without scaling; moreover, the degrees of freedom (dof) related to these two transformation schemes are $\frac{1}{2}d(d+1)+1$ and $\frac{1}{2}d(d+1)$ respectively.
- **Non-Rigid**: a non-rigid transformation is referred typically to *affine transformation*, whose projection resides in affine spaces, *i.e.*, generalizations of the typical Euclidean spaces. Therefore, the mapping will preserve visual patterns as points, straight lines, planes and so on, but not necessarily angles between lines or distances between points (*e.g.*, scaling, rotation, translation, reflection, *etc.*). The typical model is $T_\Theta(\mathbf{p}) = \mathbf{V}\mathbf{p} + \mathbf{t}$, where $\mathbf{V} \in \mathbb{R}^{d\times d}$ is a linear transformation matrix and $\mathbf{t} \in \mathbb{R}^d$ the translation vector of the point w.r.t. the origin. Hence, the affine transformation consists just in an application of the parameter set $\Theta = \{\mathbf{V}, \mathbf{t}\}$.[5]

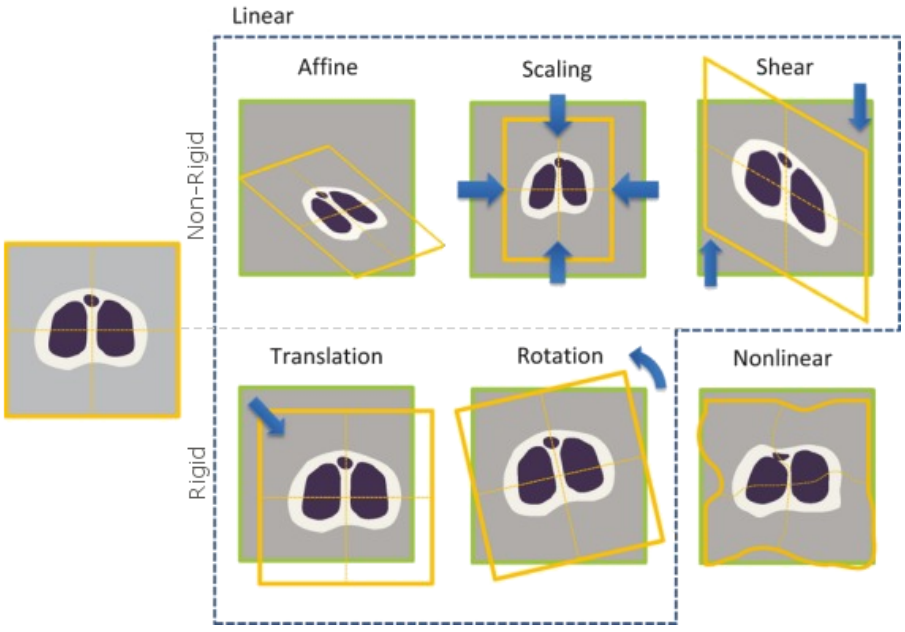


Figure 2.2: *Classification of several geometrical transformations in Computer Vision with some example on a 2D image.*

[5] In the context of non-rigid transformations are commonly used further geometrical transformations, as for example the non-linear transformations which can be parametrized by the eigenvalue or thin plate spline too. We skip to focus in deep these formulations since they are not relevant topics for this thesis.

In Computer Vision, there exist different formal camera models paired with digital camera devices, but the most employed and reference scheme which is considered in majority of applications is the well-known *pinhole camera model*. This camera is a closed black box with a single pinhole in the centre of a surface, whose light rays can enter from the external environment. The image is vertically flipped and projected over the parallel planar surface inside the camera box with respect to the pinhole. The latter determinates the position of a *virtual plane* outside the camera box, which contains the visible scene, *e.g.*, the specular copy of the acquired image. Therefore, this is a symmetric system which is aligned by the *optical axis* with origin in the pinhole (*i.e.*, perspective projection). In other terms, a pinhole camera is just a classical *Camera Obscura* without a physic lens (see Figure 2.3). The latter represents the main weak point in real cameras, since produces deformations of the incident light rays; although, there exist several methods that rectify this physical defect in polynomial time by the emulation of real world distortion patterns. Formally, a pinhole camera can be described by a set of fundamental parameters: the *pose* or *extrinsic parameters* of the camera with respect to real world $\Theta^c = \{\mathbf{R}, \mathbf{t}\}$, which are expressed by a *rotation matrix* $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a *translation vector* $\mathbf{t} \in \mathbb{R}^3$ from the origin; the camera *focal length*, which is the distance f of the projected image with respect to the pinhole; the *coordinate system* of the virtual plane which intersects the optical axis in the *principal point* $\mathbf{c} = (c_x, c_y)^T$, *i.e.*, the image centre.

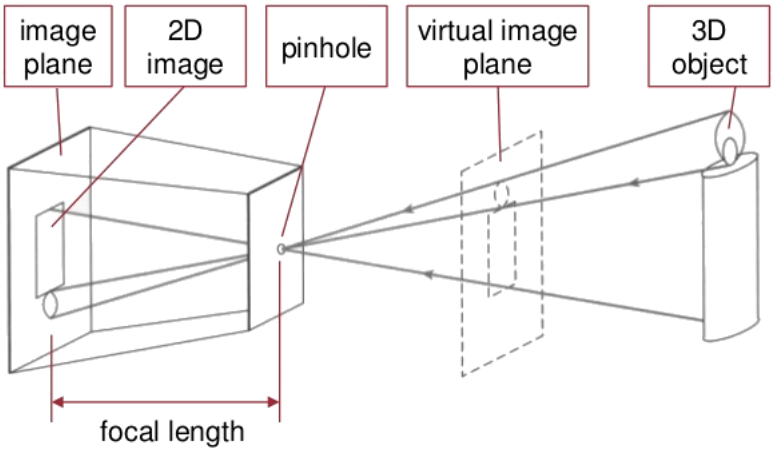


Figure 2.3: *Schema of the fundamental elements which describe the Pinhole Camera Model (Camera Obscura).*

The projection of a point $\mathbf{p} = (p_x, p_y, p_z)^T$ from the 3D real world coordinate system to the equivalent image point $\mathbf{p}' = (p'_x, p'_y)^T$ in 2D pixel coordinates is determined by two separated steps (see Figure 2.4).

1. **From World to Camera**. The first step consists to solve the point from real world $\mathbf{p}$ to camera (or film) coordinate systems $\mathbf{p}^c = (p^c_x, p^c_y, p^c_z)^T$. The mapping is obtained by a rigid body transformation according the *extrinsic parameters* of the camera

Θ^c, which is formulated in *homogeneous coordinates*[6] as follows

$$\begin{pmatrix} p_x^c \\ p_y^c \\ p_z^c \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}.$$

2. **From Camera to Pixel**. The second step consists to project the point in the camera coordinate system $\mathbf{p}^c$ onto the virtual planes of the system. This further transformation has to take in account the natural problem to discretise the real point in a pixel point, which is slightly referred to physical characteristics imported in the camera model. The latter are described by a set of *intrinsic parameters* of the camera system, which are the vertical f_x and horizontal f_y focal length to generalise non-squared pixel, an *aspect ratio* α, a skewness factor γ between the x and the y axis for non-rectangular pixels (rhombi/parallelograms) and the principal point $\mathbf{c}$. These parameters are organized in a further new structure called *camera calibration matrix* as follows:

$$\mathbf{K} = \begin{pmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -f & \gamma & c_x \\ 0 & -\alpha f & c_y \\ 0 & 0 & 1 \end{pmatrix}.$$

The final mapping is obtained by a linear transformation combining the calibration and projection $\mathbf{\Pi}_0 = (\mathbf{I}_3\ \mathbf{0}_3)$ matrices as follows:

$$\lambda_z \begin{pmatrix} p'_x \\ p'_y \\ 1 \end{pmatrix} = \mathbf{K}\mathbf{\Pi}_0 \begin{pmatrix} p_x^c \\ p_y^c \\ p_z^c \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} p_x^c \\ p_y^c \\ p_z^c \\ 1 \end{pmatrix}.$$

In practical applications, the majority of camera models are set with the assumption of perfected squared pixel $\alpha = 1$, resulting to fix intrinsic parameters as $f_x = f_y = -f$ and $\gamma = 0$ with the image centre in $\mathbf{c} = \mathbf{0}_2$. Typically, the whole conversion from 3D World to 2D image pixel coordinate systems is realized directly by the definition of an unique 3×4 projection matrix $\mathbf{P} = \mathbf{K}\mathbf{\Pi}_0 \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{pmatrix} = \mathbf{K}\,(\mathbf{R}\ \mathbf{t})$, which is well-known as *camera matrix*. Therefore, the general compact rule in homogeneous coordinates can be formulated as follows

$$\lambda_z \begin{pmatrix} p'_x \\ p'_y \\ 1 \end{pmatrix} = \mathbf{P} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}.$$

[6]The homogeneous coordinates or *projective coordinates* represents a system of coordinate which resides in *projective geometry*. By contrast to Cartesian coordinates in Euclidean geometry, a homogeneous point is actually a set of points *at infinity* in an ideal line projected in a certain direction from the origin. Given a point $(x, y)^T$ on the Euclidean plane, for any $\lambda_z \neq 0$ the triple $(x\lambda_z, y\lambda_z, \lambda_z)^T = \lambda_z(x, y, 1)^T$ determinates the set of homogeneous coordinates for the point.

The essential benefit in describing a camera system formally consists to give a methodology to perform an accurate interpretation of the real world scene. Moreover, once the parameters of model are established, the prospective geometry can be applied as verification tool to detect possible point which does not hold the expected rules, therefore potential outliers in the data.

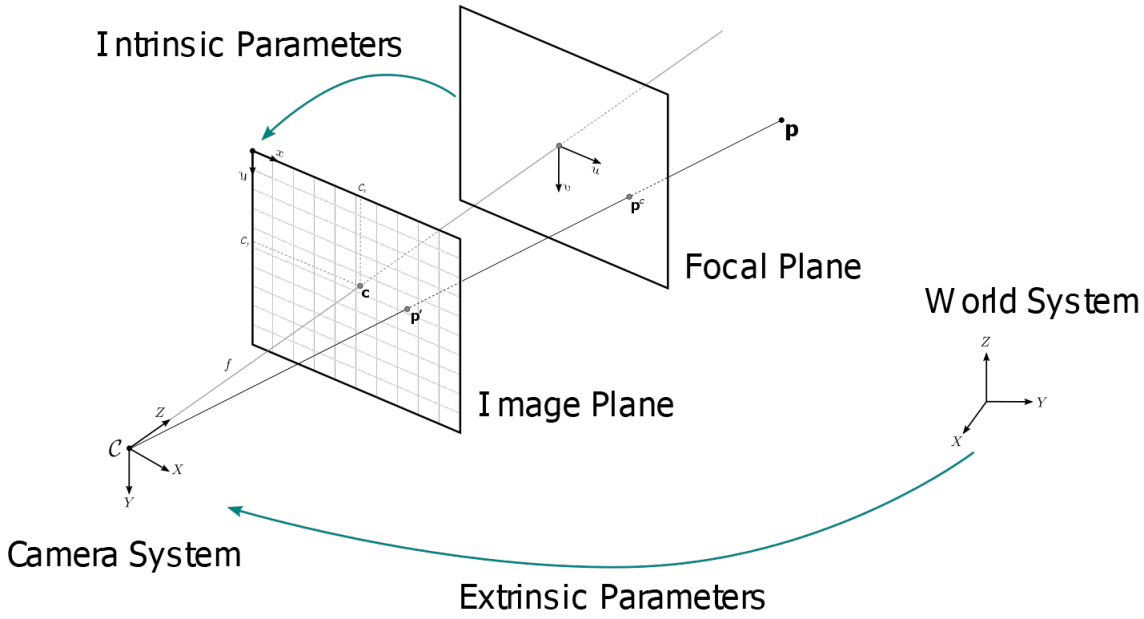


Figure 2.4: *Schema of the projection between Camera to Pixel coordinate systems.*

So far we have considered the case operating with a single camera, but such model can be extended considering an enlarged system composed by two distinct camera models. The result is a Stereo Vision scheme, which is governed by a fundamental relation that binds the coordinates of the common target point $\mathbf{p}$ in real world projected against two different cameras, which are not just independent systems actually.

The representation of this new vision model can be described formally by the well-known Epipolar geometry (see Figure 2.5). According the pinhole camera model, let $\mathbf{o}_1, \mathbf{o}_2$ be the axis origins of the camera coordinate systems and $\mathbf{x}_1, \mathbf{x}_2$ the projected points of $\mathbf{p}$ onto the image planes, the *Epipolar plane* is the space circumscribed among the *baseline* between the two origins and the optical lines of the two systems to the common target object in real world (*i.e.*, the triangle by connecting the vertexes $\mathbf{o}_1, \mathbf{p}, \mathbf{o}_2$). The *epipols* $\mathbf{e}_1, \mathbf{e}_2$ are the points on the two image planes pierced by the baseline. By the intersection of the epipolar plane with respect to the image planes are derived the *Epipolar lines* l_1, l_2, which are paired segments between the epipols and the respectively projections of $\mathbf{p}$ laying in the image planes. The root *Epipolar constraint* consists just that points laying on l_1 can match only in the epipolar line l_2.

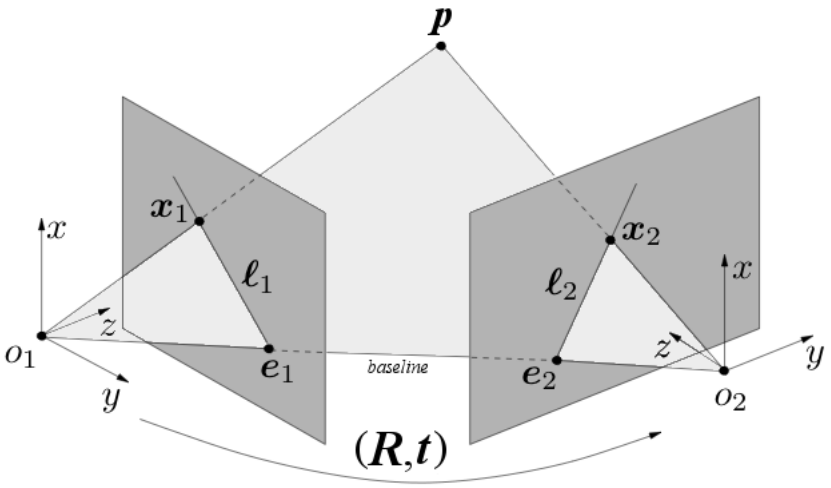


Figure 2.5: *Schema of the Epipolar Geometry applied in a Stereo Vision System.*

The formal description of the coordinates between these two systems is possible under the assumption that the rotation $\mathbf{R}$ and translation $\mathbf{t} = (t_x, t_y, t_z)^T$ from the first camera to the second one is known, which are extrinsic parameters $\Theta^s = \{\mathbf{R}, \mathbf{t}\}$ of the stereo camera system. There are two main epipolar constrains which can be formulated according the following cases.

1. **Uncalibrated case**. In this situation, the two cameras are not calibrated, *i.e.*, there is not possible to estimate the intrinsics parameters in $\mathbf{K}_1$ and $\mathbf{K}_2$. Nevertheless, a slight epipolar constraint can be formulated according the existence of the transformation $\mathbf{E} = \mathbf{t} \times \mathbf{R}$ well-known as *essential matrix* [100], which is a transformation derived by the cross product between extrinsic parameters of rotation and translation. Therefore, the mapping between the coordinates of both cameras has to hold the following condition:

$$\mathbf{x}_1^T \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_y \\ -t_y & t_x & 0 \end{pmatrix} \mathbf{R}\mathbf{x}_2 = \mathbf{x}_1^T \mathbf{E}\mathbf{x}_2 = 0.$$

   Moreover, there are some special properties deriving by possible applications of the essential matrix:

   - the epipolar lines of the two cameras, which are respectively determined by $l_1 = \mathbf{E}\mathbf{x}_2$ and $l_2 = \mathbf{E}^T\mathbf{x}_1$;
   - the epipols gets null the respective transformations $\mathbf{E}\mathbf{e}_2 = \mathbf{E}^T\mathbf{e}_1 = 0$;
   - $\mathbf{E}$ is a *singular matrix* with rank 2;
   - $\mathbf{E}$ has in overall 5 degrees of freedom (dof), because 3 are derived by $\mathbf{R}$ and 2 by $\mathbf{t}$ (since that is up a scale due to homogeneous coordinate system).

2. **Calibrated case**. In the scenario where the intrinsics parameters $\mathbf{K}_1$ and $\mathbf{K}_2$ of the two cameras are known respectively, the epipolar constraint can be finally generalized to enclose completely the overall information in the stereo vision system. From the underlying essential transformation is derived the natural extension of this scheme by the transformations of the points applying the inverses of the calibration matrices against $\mathbf{E}$. Therefore, the resulting mapping of the coordinates over the two image planes has to hold the following condition:

$$(\mathbf{K}_1^{-1}\mathbf{x}_1)\mathbf{E}(\mathbf{K}_2^{-1}\mathbf{x}_2^T) = \mathbf{x}_1(\mathbf{K}_1^{-1^T}\mathbf{E}\mathbf{K}_2^{-1})\mathbf{x}_2^T = \mathbf{x}_1\mathbf{F}\mathbf{x}_2^T = 0,$$

where the structure $\mathbf{F} = \mathbf{K}_1^{-1^T}\mathbf{E}\mathbf{K}_2^{-1}$ is well-known as *fundamental matrix* [104]. In the latter, we recognize some important properties similar to the essential matrix as follows:

- determination of the respectively epipolar lines, *i.e.*, $l_1 = \mathbf{F}\mathbf{x}_2$ and $l_2 = \mathbf{F}^T\mathbf{x}_1$ on the image planes;
- the epipols gets null the respective transformations $\mathbf{F}\mathbf{e}_2 = \mathbf{F}^T\mathbf{e}_1 = 0$;
- $\mathbf{F}$ is a *singular matrix* with rank 2 and $\det(\mathbf{F}) = 0$;
- $\mathbf{F}$ has in overall 7 degrees of freedom (dof), because 3 are derived by $\mathbf{R}$ and 2 by $\mathbf{t}$ (considering 9 variables due to a 3×3 matrix, there exists just 8 independent ratios because the one represents the common scaling is not significant, but since $\mathbf{F}$ has also null determinant, another dof is removed finally).

The Epipolar geometry, applied in the perspective projection vision by pinhole cameras, introduces the critical task to estimate two possible structures to solve the mapping between world and image domains, *i.e.*, the essential matrix $\mathbf{E}$ or the fundamental matrix $\mathbf{F}$. Actually, there is not always the requirement to solve both problems, since there are scenarios in which some parameters can be already known. For instance, if the intrinsic parameters of the camera are given, the latter are sufficient just to derive the essential matrix solving whole scheme, which is a task quite simple due to its low degrees of freedom; otherwise, this research will turn to the fundamental matrix, but requiring a greater effort. Moreover, there are further special scenarios in which knowing the kind of environments can simply the estimation. For example, if the points in world coordinates laying in a planar surface, the fundamental matrix is an Homography [110, 155, 182], which is a special class of affine transformation which can be estimated efficiently just establishing 4 point matchings. In literature there are many algorithms and approaches to solve this special task [63, 170, 197], but we skip to review them in detail for the low relevance than other topics of this thesis.

2.2.2 Feature Point Detection

In several applications where is required the alignments of different views from a common scene emerges the problem to detect special points which can be matched to perform

correctly the 3D reconstruction. Moreover, there exist further problems where the interest is not related to establish relation between multiple images, but just to extract salient reference points to induce new information or perform some image processing task. The preferable property which should be held by these keypoints consists to be both possibly repeatable in similar images and well characterized inside the single image. After that, the research of reliable reference points should be generally a task which cannot be dependent by further frames, but applicable just from the information contained in a single shot. Indeed, if we consider this problem in relation with the matching techniques, the latter cannot manage independently feature extraction, but they can be applied at most as filters to prune redundant keypoints or outliers, which is totally another task with respect to detect them for the first time. Typically, these special points (or small areas) can be recognized by some invariant properties of the light (*e.g.*, the corners of an object) as well as analysing several spatial aspects with respect to own neighbourhood (see Figure 2.6 for an example). Despite the rich literature in this field [87, 138], we review just the most well-known and vastly employed techniques in the panorama of Computer Vision.

Figure 2.6: *Example of feature keypoints (red circles) detected with SURF [10] descriptor on a 2D image.*

The early and basic methodologies to extract keypoints from an image can exploit of general detectors, as for example the Harris Operator [62] and the Difference of Gaussian [106], which give a solution with sub-pixel precision as well. Other strategies are based to recognize affine invariant regions, such as Maximally Stable Extremal Regions (MSER) [109] and Hessian-Affine [113]. The latter approach is more preferable, since affine transformations are also employed in further applications, *e.g.*, registration problems. In general terms, devising keypoint descriptors based on common methodologies which may be shared by other different techniques could increase considerably (at least in principle) the accurateness of the results in more complex pipeline processes.

Typically, the output of any keypoint detection approach consists just in a set of vectorial descriptors assigned to the retrieved keypoints, which contains their pixel coordinates and further affine invariant features of the regions. The Scale Invariant Feature Transform

(SIFT) [101] detects keypoints which are invariant by the emulation of transformations such as rotation, scaling, illumination and different 3D camera viewpoints. Speeded Up Robust Features (SURF) [10] is considered a faster alternative than SIFT, which is based to an integer approximation of the determinant of Hessian related to a Blob detector [94] just by three integer operations; the derived descriptor is based on a sum of the Haar Wavelet signals around the point of interest. The Gradient Location and Orientation Histogram [114] is based on SIFT, but considers more spatial regions for the histograms. The Gradient Location and Orientation Histogram [114] is built on the local energy model of feature perception, combining the signal of several filter orientation and local histograms from patches of the image. GIST descriptor [121] produces a low depiction of a picture as an holistic representation of its spatial envelope, without requiring segmentation methods. A common problem of the descriptors introduced so far consists to generate orderless bag-of-features depiction of the images, that cannot register properly the spatial layout of the features, hence getting difficult to detect shape or segmenting objects in a background. The Spatial Pyramid Matching (SPM) [88] refines an initial object descriptor partitioning iteratively the image into increasingly finer spatial subregions whence are computed statistics (or histograms) of local features.

Nevertheless the vast selection of feature descriptors available in the context of Computer Vision, the most employed techniques remain always SIFT-like, since offer a good trade-off between precision and performance. Recently, there are proposed further approaches which are based on variants inspired to SIFT descriptor: PCA-SIFT [81] increases the distinctivenesses of the features by applying PCA to reduce the redundant information in the gradient patch; PHOW [19] enlarges the density of the descriptor as well as exploits of the neighbours colours of the keypoint as further information; ASIFT [116] solves some general problems of the classical SIFT descriptor to deal with the *tilt phenomenon* of the cameras.

2.2.3 Matching Problems in Computer Vision

The amplest macro-group of problems in Computer Vision is essentially related by an inference task which involves pairs of visual instances (in two or three dimensions). After that, there is not other way in principle to infer very reliable structured information regarding the real world environment just analysing one and only one observation, at least that by a machine. In fact, a static image could be considered as a sort of "frozen entity", even if we may perform some operation on that, *e.g.*, feature extraction, filtering, segmentation, *etc.*, when we need to recognize some property that involves depth it yields to perform a comparison process, against other observations or some well-known patterns. Therefore, any Computer Vision task which involves multiples images contains implicitly and always a Matching problem (as amply discussed in sections 3.1 and 3.2).

In general, once a set of reliable reference point correspondences is obtained, the geometry of a given model is applied according the estimated parameters to verify the quality of the projection. This is commonly requested in different various tasks in Computer Vision and consists in an iterative rectification process well-known as *bundle ad-*

justment [175]. The goal consists to refine 3D coordinates describing the scene geometry, the parameters of related motion and cameras. Let N be the number of cameras and M be the number of 3D points, the general guideline is an optimization problem aimed to minimize the *reprojection error* as follows:

$$\sum_{j=1}^{N}\sum_{i=1}^{M} v_{ij} \left\| \mathbf{x}_{ij} - \mathbf{P}_i \mathbf{x}_j \right\|_2^2 ,$$

where against the i-th reference image, the binary variable v_{ij} denotes the visibility of the point $\mathbf{x}_j$ and $\mathbf{x}_{ij}$ is the associated projection.

In this section we sum up some fundamental Computer Vision tasks which involve set or sequence of visual observations, giving a general explanation of the main methods and well-known solutions in literature.

Camera Calibration

The preliminary step for any computer vision process consists on an accurate calibration of the parameters that govern the camera model, which is based mainly on the estimation of the intrinsics $\mathbf{K}$ and, in case the pose is unknown, the extrinsic parameters Θ^c as well. In Computer Vision this problem is well-known as *Camera Resectioning*, which does not have to be confused with the estimation of the motion Θ^s in a stereo vision system composed by two cameras, since camera calibration refers in general to one device. Operatively, the classical pipeline is based in two fundamental stages: an initial acquisition of 2D images of a specific *calibration object* from different point views (since a single view is not sufficiently informative to perform a complete calibration [132]); retrieving the matches between repeatable reference points in the acquired images and applying the geometrical rules which describe a camera model to estimate the parameters.

The initial step is very critical, because the choice of the geometry and additional information related to the target object involve totally the accurateness of the calibration. In this task, 3D calibration objects (*i.e.*, specific shapes which does not support coplanar geometry) represent the best solutions, because both there is not dependence between extrinsic and intrinsic parameters with respect to the reprojection error, and depth irregularities on the calibration object can compensate lens distortions. Although, there are two main setbacks, since establishing the best geometry for such objects is not always a clear task and their manufacture is in general quite expensive. To overcome these problems, cheaper solutions are based on one-dimensional objects [196], *i.e.*, planar surfaces, that can trust of efficient homographic transformations in the next steps. The final problem consists in the decision of the feature pattern which covers the 3D or planar surfaces related to the calibration objects, which is fundamental to detect the correct set of reference points to match. Zhang [195] proposes the typical chessboard-like pattern, which works very well in general but due to the asymmetry of the marker could arise localization errors in some real world circumstances (*e.g.*, light bleeding). Alternatives to solve such problem could be the symmetric marker such as circular point [68] and checker-board [102].

High quality target points on hand, registration approaches are used to derive proper correspondences (see section 3.1.2). Assuming such demand is solved in accurate way, the final task consists to estimate the camera parameters. In this new step is worth to be noted that a *calibration method* is bound to the reference *camera model*, which are two different aspects even though strictly subject to each other. After that, there are camera models which consider more complex real world abstractions, *e.g.*, the visual distortion of the lens, therefore the associated calibration method has to take in account of further elements. In a simplistic scheme of a pinhole camera which does not take in account of distortion parameters, this problem yields in a general *linear transformation* as in works by Hall *et al.* [61], Faugeras *et al.* [43] and the well-known Direct Linear Transformation (DLT) algorithm proposed by Sutherland [164]. By contrast, in more sophisticated camera models is necessary to introduce new parameters to describe distortions. Brown [23] proposes a complex approach based on three radial and two decentering distortion coefficients and more recently, Heikkilä [68] introduces even two new coefficients of tangential distortion. For a comprehensive review of the most important calibration methods we refer to [139].

Structured Light

In the typical scenario where a computer vision system returns a set of coarse 3D point clouds, applying a triangulation process there is possible to obtain a more structured description of the surfaces. The reliable reconstruction of the overall 3D environment is possible by merging each surface in suitable way (as Figure 2.7), which is possible just detecting some reference features to dovetail all the patches. This task requires to extract feature points between two surfaces that can be matched to understand those joint points are in common to perform such reconstruction. In this place, the critical problem consists to have the guarantee that these invariant features are always available and sufficiently dense to solve the problem without ambiguities. There exist several and complementary approaches to deal with this problem in literature [143], but the common strategy trusts to *photometric correlation* only, *i.e.*, exploiting of the direction of the incidence natural light available in the environment on the materials. The weakness in this approach can be dealt with the introduction of artificial non-contact guidelines on the scene directly by the system, for example projecting bright patterns to enhance the morphological characteristics of the target object: this further class of computer vision techniques is well-known as Structured Light [9].

Typically, the most employed solutions are based on multiplexing methodologies based on n-ary or Gray codes as well as further hybrid schemes [141]. The simplest case consists to assign to each pixel a binary codeword which matches to blank and white stripes projected in the scene over the time. Alternatively, this scheme can be enhanced introducing patterns in grayscale. Although, these approaches suffer of low scanning resolution due to the small coding space. Nevertheless, even if the number of measurements may be enlarged, the latter does not a guarantee to improve preciseness, which is a problem related to the nature of the light over special materials, *e.g.*, low reflective or smooth

metal areas. Therefore, these approaches can be really reliable just under the well-known *uniform albedo* assumption of the surfaces.

Another methodology for structured light applications is based on Phase Shifting [163] measurements. The scanning resolution can be increased by a periodic projection of different light patterns over the time: the new information associated to each pixel is just determined by the *phase shift*. These techniques yields to sub-pixel precision and spatial measures in high resolution, due to the uninterrupted phase distribution within the period. Moreover, the shifting pattern enhances soft discrepancies on the materials, resulting in a stronger tolerance in non-uniform albedo surfaces. Although, the *M-step relationships* applied in these approaches can arise ambiguities, since they are *relative* phase values actually. This weak point of the method may be reduced by integrating Gray coding schemes, even if that yields considerably in huge number of patterns which are clearly prone to generate errors [93].

Optical profilometer techniques represents a further class of approaches which is based on quantifying the roughness of the objects by well-known surface profiles. The general methodology extracts profilometer measures by the variation in height of a planar reference surface scanning the scene orthogonally [159]. The classical strategy is based to project grating or sinusoidal patterns which are shifted first on the reference plane and then on the object of interest. Finally, the phase shift information can be estimated on the well-known *fringe pattern*, in general by means of various techniques or other more precise approaches based on Fourier Transformation [167]. Although, optical profilometer methods share generally the common drawback to support limited variation in height of the surfaces causing a high rate of ambiguous measurements.

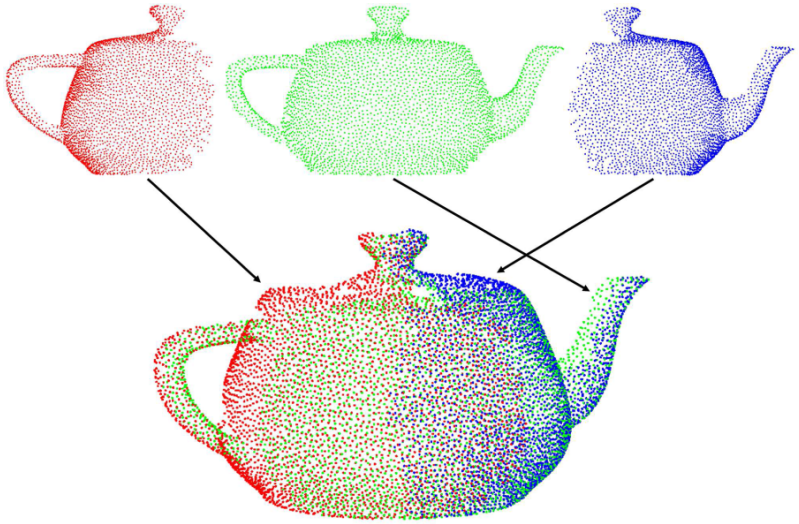


Figure 2.7: *Example of object reconstruction by merging three 3D point clouds which describe different shots of a teapot.*

Structure from Motion

One of the most influential and studied application in Computer Vision is addressed to extract 3D structures just taking in account of frame sequences in 2D dimensions, which can be coupled with motion information (*i.e.*, poses of the cameras). This method trusts on well-known *range imaging* techniques, in which for each pixel is assigned an estimation of the distance with respect to an established point, that is in general related to the camera pose. In fact, such application is inspired from the natural behaviour of human being to get reliable information of the environment just by *moving* inside it and simultaneity analysing the movements of fixed or even animated objects in their vision field over the time. In general, this problem is treated as an articulated pipeline of sub tasks which is described by the well-known paradigm of Structure from Motion (SfM) [188]. Typical applications that benefit of SfM are for example Advanced Human Interfaces [4], 3D Reconstruction [95], 3D object recognition [112] and SLAM [137].

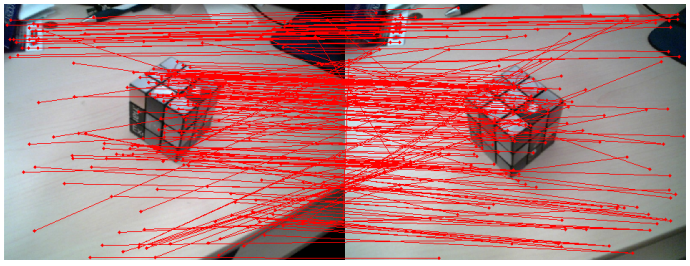
Structure from Motion pipeline is generally divided in two stage macro-groups: *Image based local steps* and *Structure based global steps*. In the former, the main activities consists to detect salient keypoints from rough 2D images by classical feature descriptors and then performing some general registration technique to align the features between consecutive instances. In the latter, correspondences on hand, the guess of depth and motion in 3D real world environment is performed in order to reconstruct globally the 3D scene. Typically, initial guess can be rectified by an iterative bundle adjustment step. Once the optimal solution is estimated the result consists in a set of 3D surfaces which can be reassembled together to obtain the final reconstruction of the 3D environment or the object of interest (as in Figure 2.7). This post-processing step is based in a further matching problem, in particular a registration task between range images, with the purpose to establish the best alignment of the surfaces and possibly using the motion information estimated for each point to lead aptly the process. Regardless the specific technique employed to establish reliable correspondences, the final step to complete the surface registration is always related to the estimation of a rigid transformation that aligns the couple of points reducing the squared distance between them (see section 3.1.2). Considering the overall SfM work-flow, in the transition from a stage to the other one there is always applied some filtering process to improve the quality of the intermediate solution, which can trust either on the nature of rough image data or more articulated properties such as camera projection rules, distance measurements and so on.

In general, the registration problems are categorized in two possible scenarios: *fine* or *coarse* registration methods. The main difference between these two classes consists that fine methods can rely on initial guess to mitigate wrong matches, which can be particularly related to the specific context of application, by contrast than rougher approximations based just on similarity measures over the features.

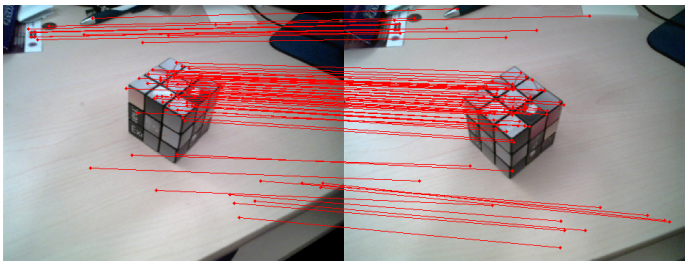
The majority of techniques based on fine alignments are in general just special variants of the well-known Iterative Closest Point (ICP) method [14, 135]. The main differences which distinguish such approaches relate to the particular way that points laying on the surfaces can be sampled, the rejection of incompatible matches and the error measures

to establish these decisions. In general, the root setback of all these techniques is the iterative nature of the ICP-based method, which does not guarantee to estimate optimal solutions. Recently, novel strategies introduce new probabilistic schemes to evaluate candidate correspondences based for example by Evolutionary model [96] or Expectation Maximization [60].

Conversely, coarse registration approaches may be categorized in three main classes, such as global methods, feature-based methods and inlier selection methods based on RANSAC [44] or PROSAC [32] (see Figure 2.8 for an example of this application). In general, global methods for instance PCA [34] and Algebraic Surface Model [168] rely on some global constrain related to the surface. Although, the main problem is due to well-known occlusion phenomenon, *i.e.*, some region may be hidden by other foreground elements in the scene. Feature-based methods are not sensitive in general to this problem since they operate locally in the scene and thus work very well even with surfaces which do not present complete overlapping areas. By contrast, the RANSAC-based methods solve the registration problem in very different strategy. Chen *et al.* [28] propose the RANSAC-Based DARCES algorithm, which consists to select randomly the set of correspondences between the surfaces and using the accuracy of the estimated transformation to weight the matches. More recently, Aiger *et al.* [5] introduce an alternative consensus strategy based on Four Points Congruent Sets, in which noisy data are removed by a filtering process and the registration is speeded-up with an early verification step. Further interesting works based on accuracy evaluation are exhaustively reviewed in [140].



(a)



(b)

Figure 2.8: *Two matching experiments with SIFT [101] descriptor, in the simple case (a) establishing the correspondences between the nearest key features or (b) applying RANSAC [44] method to filter out the wrong matches.*

3

Matching Problem and Generalizations

3.1 Matching Problem

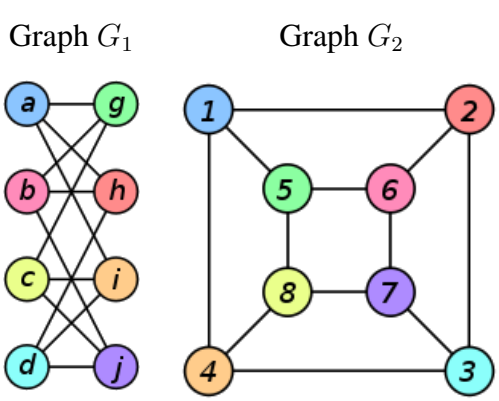
In several areas of Machine Learning is frequently required to retrieve a *correspondence* relation between two or multiple objects. This specific information can assume different meanings according the type of application where a given problem has to be treated. In Pattern Recognition or more generally in Data Mining, the relations among entities could reveal aspects of similarities, dependences, interactions and any other kind of bonds that once are discovered allow to derive new information from apparently uncorrelated objects. The technical term *matching* could equally refer to an *alignment* as well, since data structures which describe the individuals of a problem can be *permuted* or *transformed* according the estimated correspondences between them. The complexity of this problem depends mainly by the specific type of data representation that describes the objects, as for example vectorial points, graphs, images and so on. Moreover, even if there exist theoretical exact solutions for these problems, in practical applications are infeasible both for their complexity order and for the presence of noise in the data. Although, these fundamental drawbacks are overcome finding approximations of the real solutions by the formulation of relaxed schemes.

In this section we introduce two classical matching problems in Machine Learning which find extensively applications in the sub-branch of Computer Vision: the Graph Matching problem (section 3.1.1) and the Point Set Matching problem (section 3.1.2). We present such topics first giving a formal definition of the matching problems and secondly reviewing the main related approaches in literature.

3.1.1 Graph Matching Problem

Graph is a fundamental data structure which describes different kind of real-world objects as a set of interconnected nodes, in which can be assigned an orientation and weight to links as well as additional attributes to nodes (see section 2.1.1). Typical areas of the science which exploit of graphs are for example Bio-informatics, Computer Vision,

Chemistry and so on. The extreme power that is recognized by representing data as a graph consists mainly in the high capability to catch just the essential characteristics that describe an object. Although, exploiting of such expressiveness has a cost which emerges heavily when different graphs require to be analysed in some pattern recognition task. The Graph matching (GA) [35,45] represents the challenging problem to match the nodes of a couple of graphs according their intrinsic topologies, in order to discriminate or compare their substructures in reliable way.



An isomorphism between G_1 and G_2

$$\begin{aligned}\phi(a) &= 1\\ \phi(b) &= 6\\ \phi(c) &= 8\\ \phi(d) &= 3\\ \phi(g) &= 5\\ \phi(h) &= 2\\ \phi(i) &= 4\\ \phi(j) &= 7\end{aligned}$$

Figure 3.1: *Example of graph isomorphism ϕ between two graphs.*

Unfortunately, it is not just a trivial task to solve in practice this problem, because there does not exist an unique structural representation for the same graph since nodes are ordering-free entities: in Graph Theory this property is formalized by the well-known notion of *graph isomorphism* (see Figure 3.1). More specifically, the presence of an isomorphism between two graphs G_i and G_j is encoded with the existence of a bijection ϕ that maps the node labels of a graph with respect to each other preserving the same topologies. In other terms, if two graphs are isomorphic, it means they contain the same relations, hence they are equivalent. For example, an exact and naive solution for this problem consists just to list all possible permutations of nodes to look for the one that realigns the graph G_j to G_i. Formally, Graph Matching can be expressed as a set of combinatorial constrains which is mathematically formulated as a quadratic assignment problem (QAP) [82, 99]. Nevertheless the extensive research in this topic, the main limit is related on the complexity to solve ϕ in polynomial time, since such QAP remains infeasible in practicals tasks belonging to the NP-Hard class [49]. For this reason, real applications are based on heuristic approaches, where an approximated optimal solution is obtained from a relaxation of the crisp assignments as the trade-off to reduce computational time and complexity for the problem.

Problem Definition

We give here a general and formal definition of Graph Matching, which is sufficiently expressive to cover the majority of solutions we find in pattern recognition field.

Let $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$ be two graphs which are for simplicity undirected and unweighed. The Graph Matching problem consists to find a map $\phi : V_i \to V_j$ such that maximizes the objective function $S(G_i, G_j, \phi)$, which scores the correctness of the alignment, in other terms it can be expressed as the following maximization problem:

$$\overline{\phi}_{ij} = \arg\max_{\phi} S(G_i, G_j, \phi). \tag{3.1}$$

This formulation governed by an energy function is very flexible, since it does not force the process to return an *exact* solution for the matching, but it considers just the *best* estimated map $\overline{\phi}_{ij}$ that a graph matching method could require. Since the problem (3.1) is of maximization, the alignment score $s_{ij} = S(G_i, G_j, \overline{\phi}_{ij})$ could be interpreted even as an indicator of similarity between two graphs, *i.e.*, greater is s_{ij} much more G_i and G_j are similar. We have already said that the optimal map is preferred to be injective and total, which is the necessary condition for exact graph matching, but this is not requested in our general formulation actually, hence it allows to find an inexact (but *best*) solution even with graphs of different size $|V_i| \neq |V_j|$. As a consequence, we can generalize the formulation of subgraph matching by solving (at least) an injective mapping yielding in a form of induced subgraph isomorphism. Here, it is worth to be mentioned that such condition may bear to describe even the well-known subgraph monomorphism [54], which is a further exemplification that does not preserve the topological properties also in the opposite map direction, *i.e.*, including extra edges as well. Although, we specify that our reference definition of (sub)graph matching (which leads the works presented in this thesis) considers always symmetric transformations and thus the inverse mapping is ever determined as $\overline{\phi}_{ji} = \overline{\phi}_{ij}^{-1}$.

The same optimization (3.1) can be redefined in terms of a quadratic assignment problem (QAP) just reformulating its fundamental components in matricial form. We denote the cardinalities $n_i = |V_i|, n_j = |V_j|$ for the node sets and $m_i = |E_i|, m_j = |E_j|$ for the edge sets related to the two graphs. Let the affinity matrices be $\mathbf{W}_v \in \mathbb{R}^{n_i \times n_j}$ over the nodes and $\mathbf{W}_e \in \mathbb{R}^{m_i \times m_j}$ over the edges, we define the global affinity matrix in this way $\mathbf{W} \in \mathbb{R}^{n_i n_j \times n_i n_j}$, which can be simply formulated as a specific combination ω of node and edge affinities matrices as follows: $\mathbf{W} = \omega(\mathbf{W}_v, \mathbf{W}_e)$. Formally, the entry $w_{ar,bs}$ in $\mathbf{W}$ contains the affinity for edge pair $(v_a^i, v_b^i) \in E_i$ and $(v_r^j, v_s^j) \in E_j$, while the diagonal term $w_{ar,ar}$ the affinity between the node $v_a^i \in V_i$ and $v_r^j \in V_j$. The data structure $\mathbf{W}$ plays a fundamental role in Graph Matching, since encapsulates as a sort of black-box all the information available (node attributes and topologies) of the matching problem. It is not necessary to introduce in deep how to solve $\mathbf{W}$, but we can suggest just an elegant factorization in [198]. Let Σ^b be the space of the binary permutation matrices (one-to-one mapping) as the following set:

$$\Sigma^b = \{\mathbf{P} | \mathbf{P} \in \{0,1\}^{n_i \times n_j}, \mathbf{P}\mathbf{1}_{n_j} \leq \mathbf{1}_{n_i}, \mathbf{P}^T\mathbf{1}_{n_i} \leq \mathbf{1}_{n_j}\}, \tag{3.2}$$

where the inequalities in the definition allow to support graphs of different sizes and $\mathbf{1}_n$ is a vector composed by n ones.[1] Encoding ϕ just as the binary permutation matrix $\mathbf{P} = (p_{ar}) \in \Sigma^b$, which maps the vertexes in G_i to vertices in G_j (*i.e.*, given two nodes $v_a^i \in V_i$ and $v_r^j \in V_j$, if $\phi(v_a^i) = v_r^j$ holds that $p_{ar} = 1$), then Graph Matching problem is formulated as the following QAP (Lawler's formula [99]):

$$\overline{\mathbf{P}}_{ij} = \underset{\mathbf{P} \in \Sigma^b}{\arg\max} \quad \text{vec}(\mathbf{P})^T \mathbf{W} \text{vec}(\mathbf{P}), \tag{3.3}$$

where $\text{vec}(\cdot)$ is a function that returns the vectorized version of a matrix. Permutation matrices defines the basis to describe formally the structural similarity between two graphs, since the existence of exact (sub)graph isomorphism yields just to the equality as follows

$$\mathbf{A}_i = \overline{\mathbf{P}}_{ij} (\overline{\mathbf{P}}_{ij} \mathbf{A}_j)^T.$$

The computation and management of the affinity matrix $\mathbf{W}$ could be very critical in practical applications due to its dimensionality. There exists another version of the score function in (3.3) which does not take in account of the similarity between edges in matricial form. Let $\mathbf{A}_i$ and $\mathbf{A}_j$ be respectively the adjacency matrices representing the two graphs, then Graph Matching problem is formulated as the following QAP (Koopmans-Beckmann's formula [82]):

$$\overline{\mathbf{P}}_{ij} = \underset{\mathbf{P} \in \Sigma^b}{\arg\max} \quad \text{tr}(\mathbf{P}^T \mathbf{A}_i \mathbf{P} \mathbf{A}_j) + \text{tr}(\mathbf{W}_v^T \mathbf{P}). \tag{3.4}$$

Main Approaches

The Graph Matching problem has been studied for decades in pattern recognition community [35]. There two main families of solutions in literature: the *exact* graph matching, whose goal consists to find a bijection between the nodes of two graphs and the *inexact* graph matching, in which the combinatorial constrains are relaxed in order to reduce the complexity of the problem by finding a heuristic solution.

The early works from the first family due to their computational demanding and the rigid method to deal with the matching problem have found a weak success in practical tasks [38, 127, 179].

Conversely it is the manner for the methods in the second approach, which have found many applications in machine learning especially in Computer Vision area. In this thesis we are interested just to review the literature which involves this heuristic methodology for graph matching problems, which could be divided further in two sub groups: *Spectral Relaxation* and *Doubly-stochastic Relaxation*. There exists another third sub group well-known as *SDP Relaxation* actually, but we decided to skip it since does not concern

[1] In real implementations this condition is very often skipped to simplify matricial calculus making the graphs G_i and G_j uniform with a common number of n nodes (two-way constrains). This is realized just replacing the graph G_j with an extended version G'_j adding the necessary n_j^o number of dummy/outlier nodes in order to obtain the condition $n_i = n_j + n_j^o = n'_j = n$.

directly graph matching techniques treated in this work.

In Spectral Relaxation works the permutation matrix holds a fundamental condition of orthogonally, which means $\mathbf{P}\mathbf{P}^T = \mathbf{I}$. Leordeanu and Hebert [89] propose the Spectral Matching (SM) algorithm, wherein the quadric assignment problem (3.3) is relaxed in order to learn the permutation matrix moving onto a real space subject to have an unit length, *i.e.*, $\|\text{vec}(\mathbf{P})\|_2 = 1$, since only the relative values in $\mathbf{P}$ are considered really relevant. The optimal solution is obtained finding the principal eigenvector of $\mathbf{W}$, which maximizes the score S by Raleigh's ratio theorem. Cour *et al.* [39] propose a generalization (SMAC) of the spectral matching incorporating affine constraints on the relaxed solution as well. Regarding restrictedly the graph matching methods formulated as the quadratic problem (3.4), under the orthogonal constrain Umeyama [180] exploits in closed-form of the eigendecomposition on the adjacency matrices, generalizing the case both weighted and directed/undirected graphs. Intuitively, the trace $\text{Tr}(\mathbf{P}^T\mathbf{A}_i\mathbf{P}\mathbf{A}_j)$ could be expressed as $\text{Tr}(|\mathbf{U}_i||\mathbf{U}_j|^T\mathbf{P})$, where $|\mathbf{U}_i|$ and $|\mathbf{U}_j|$ are respectively the eigenvectors in absolute values derived from the adjacency matrices of the graphs $\mathbf{A}_i$ and $\mathbf{A}_j$. Finally, the well-known polynomial Hungarian algorithm [21] is applied to find $\mathbf{P}$ that maximizes the trace. In literature we find other formulations [25, 147, 149] of the same QAP (3.4), which are still based by an underlying closed-form eigendecomposition. Although, in these methods the main common drawback consists that the spectral embedding of the graph nodes is not uniquely defined, in particular for two main reasons: first, the eigenvectors are defined at most up to a sign flip, therefore it is necessary to define some convention for a sign synchronization in order to maintain the biggest component always positive; second, the adjacency matrices could have multiple eigenvalues resulting just in arbitrary rotations of the eigenvectors.

In Doubly-stochastic Relaxation works the main aspect that identifies such Graph Matching approaches consists in redefinition of the permutation matrix space as follows

$$\Sigma^d = \{\mathbf{P}|\mathbf{P} \in [0,1]^{n_i \times n_j}, \mathbf{P}\mathbf{1}_{n_j} \leq \mathbf{1}_{n_i}, \mathbf{P}^T\mathbf{1}_{n_i} \leq \mathbf{1}_{n_j}\} \tag{3.5}$$

in which the one-to-one correspondences between nodes are expressed as likelihoods. This new problem space, also called Birkhoff's polytope [16] in the balanced case $n_i = n_j$, is much less rigid with respect to the binary set Σ^b, indeed the latter represents just the subset containing its extreme points $\Sigma^b \subset \Sigma^d$. Therefore, solving the graph matching problem in this further space consists in the estimation of a double-stochastic matrix since $\mathbf{X} \in \Sigma^d$, which represents the local optimum for a non-convex quadratic programming problem. Another fundamental consequence to highlight is about the loss of orthogonality property than the estimated solutions with Spectral Relaxation methods, since Σ^d is a convex hull as well. However, it is easily possible to project the final solution in the permutation space $\mathbf{P} \in \Sigma^b$ performing a discretization process in polynomial time to the matrix $\mathbf{X}$, for example by the well-known Hungarian algorithm [21] as follows

$$\mathbf{P} = \text{HUNGARIAN}(\mathbf{X}).$$

In literature one of the most influential work which exploits of the doubly-stochastic formulation is the Path Following (PATH) algorithm proposed by Zaslavskiy *et al.* [193]. The approach is defined as a convex-concave quadratic program which supports labelled and weighted graphs. The root idea consists in a linear interpolation on $\mathbf{X}$ between a convex $f_0(\mathbf{X})$ and concave $f_1(\mathbf{X})$ relaxations, which is controlled by a parameter $\lambda \in [0, 1]$ as follows

$$f_\lambda(\mathbf{X}) = (1 - \lambda) f_0(\mathbf{X}) + \lambda f_1(\mathbf{X}). \tag{3.6}$$

The algorithm could be described as a power iteration process which tracks a path of local minima $f_{\lambda+d\lambda}$ over the double-stochastic permutation space Σ^d moving between the lower bound $f_0(\mathbf{X})$ with $\lambda = 0$ and upper bound $f_1(\mathbf{X})$ with $\lambda = 1$, where $d\lambda$ encodes the updating step, until a convergence criterion is reached. The computation of the energy $f_\lambda(\mathbf{X})$ is solved effectively by a revised version of the Frank-Wolfe algorithm [46]. Even if PATH algorithm supports labelled and weighted graphs, it requires that the weighted adjacency matrices are symmetrical, *i.e.*, it works just with undirected graphs. Recently, Liu *et al.* [97] propose just an extension of Path Following, which is able to operate with directed graphs. Almohand and Duffaa [6] propose another approximation solving the non-convex quadratic problem by linear programming, even if that is intrinsically an approach closest to Path Following since also Frank-Wolfe algorithm is treated as a linear program. All these methods are devised to support only the optimization (3.4) on doubly-stochastic domain, but in literature the majority of the works operating on the convex polytope Σ^d are formulated as the more general maximization problem (3.3). The first work in this fashion is the well-known Graduated Assignment proposed by Gold and Rangarajan [56]. The method is based on a combination of graduated non-convexity, two-way constraints and sparsity resulting in an efficient process which copes with noisy data to estimate a (sub)graph isomorphism both on weighted and attributed relational graphs. The authors start to reason from the definition of a quadratic assignment problem, which is based to find a double-stochastic matrix $\mathbf{X} = (x_{ar}) \in \Sigma^d$ such that minimizes the objective function

$$-\frac{1}{2} \sum_{a=1}^{n_i} \sum_{r=1}^{n_j} \sum_{b=1}^{n_i} \sum_{s=1}^{n_j} x_{ar} x_{bs} w_{ar;bs} \tag{3.7}$$

according the usual global affinity matrix $\mathbf{W} = (w_{ar,bs}) \in \mathbb{R}^{n_i n_j \times n_i n_j}$. The solution is rooted on the idea to expand (3.7) starting from an initial condition $\mathbf{X}^0$ via Taylor series approximation. This process is controlled by the gradient matrix $\mathbf{Z}^{(t)} = (q_{ar}^{(t)})$ which describes the current assignment at time t and whose entries are determined as $q_{ar}^{(t)} = \sum_{b=1}^{n_i} \sum_{s=1}^{n_j} x_{bs}^{(t)} w_{ar;bs}$. Nevertheless, it is not guaranteed that $\mathbf{Z}^{(t)} \in \Sigma^d$, hence the matrix is finally rectified to the double-stochastic space by Sinkhorn [156] method achieving a problem of *softassign*. In order to relax the complexity is used a continuation method which solves a succession of several assignment problems by the parameter $\beta^{(t)} \in \mathbb{R}^+$ which governs the whole process as the final formulation:

$$\mathbf{X}^{(t)} = \textsc{Sinkhorn}\left(\exp(\beta^{(t)} \mathbf{Z}^{(t)})\right), \tag{3.8}$$

wherein greater is $\beta^{(t)}$ much more the *softassign* step pushes the algorithm towards integer solutions. Finally, the framework introduces slack variables in the double-stochastic matrix adding a new and provisional dimension in order to model intermediately wrong assignments due to noisy data. From another prospective, Graduated Assignment can relax the crisp constrains as a sort of two-way relaxation labelling scheme [133]. Whilst this smoothed formulation such process remains quite rigid since treats own set of variables independently and thus without considering eventual relations among different assignments. Cho *et al.* [30] solve the problem proposing a graph matching method based on Reweighted Random Walks, which can be seen as a generalization of the well-known PageRank algorithm [124]. The root idea consists to construct an association graph which can be ideally modelled like a function which depends by two graphs G_i, G_j and the global affinity matrix $\mathbf{W}$ as follows $G^{rw} = f(G_i, G_j, \mathbf{W})$. The nodes of G^{rw} define just a candidate correspondence of the nodes between the graphs. Therefore, using Markov Random Walks to rank the nodes of G^{rw} a possible optimal matching is determined. In the classical formulation of PageRank the transition matrix which involves the node visited from a walker consists in a row normalization of $\mathbf{W}$. This formulation achieves a democracy policy in which each nodes has the same total out-going weights, but in the graph matching problem this aspect could lead to strengthen false candidate correspondences (*e.g.*, outliers). To overpass such problem the association graph is reformulated as an augmented version G^{rwa}, which preserves all the relative affinities relations in G^{rw} and includes an absorbing node v_{abs} reachable from all other ones: a special state that once reached cannot be transitioned out. The authors call this approach *affinity-preserving random walk*. Let $d_{max} = \max_i \sum_j w_{ij} = \max_i d_i$ be the maximum out-going degree d_i in the nodes of G^{rwa}, the sub-stochastic transition matrix is defined as

$$\mathbf{T} = \frac{\mathbf{W}}{d_{max}}.$$

In this further setting the vectorized double stochastic permutation matrix $\mathbf{x} = \text{vec}(\mathbf{X})$ to estimate can be interpreted as a probability distribution of the unabsorbed random walker, which is defined at time $t + 1$ as follows:

$$\mathbf{x}^{(t+1)^T} = \mathbf{x}^{(t)^T}\mathbf{T}. \tag{3.9}$$

This definition lacks of the typical behaviour in ranking methods which model the concept of *jump*, in which a random walker can traverse an edge with probability α or jump to some constrained nodes with probability $1 - \alpha$. The introduction of a *reweighing jump* in the (3.9) is based in a inflation step

$$\exp\left(\beta \frac{\mathbf{x}^{(t)}}{\max \mathbf{x}^{(t)}}\right), \tag{3.10}$$

which both smooths small values and amplifies the large values in $\mathbf{x}^{(t)}$ according a scaling factor β, hence reducing the weights for uncertainly matching. Similarly to Graduated

Assignment in (3.8), there is required the Sinkhorn method to guarantee that the matricial form of (3.10) is defined in the doubly-stochastic space Σ^d. For the sake of simplicity, the two-ways normalized inflation step structure can be encapsulated as the outcome of a more general function as $\mathbf{r}^{(t)} = f(\mathbf{x}^{(t)})$. Adding this jump model in the affinity-preserving schema (3.9) is finally formulated the Reweighted Random Walks for graph matching method as:

$$\mathbf{x}^{(t+1)^T} = \alpha \mathbf{x}^{(t)^T} \mathbf{T} + (1 - \alpha)\mathbf{r}^{(t)^T}. \tag{3.11}$$

All the methods introduced so far have a common drawback which is the memory space complexity due to the maintenance of the huge global affinity matrix $\mathbf{W}$, whose magnitude is generally $O(n_i^2 n_j^2)$. Recently, Zhou and De la Torre [198] propose a method which is based in a factorization of such matrix requiring only to store affinities over the edges $O(m_i m_j)$ and nodes $O(n_i n_j)$ separately. The resulting graph matching method is still based in a path following strategy, but revised according their factorization. Moreover, by this new formulation the authors incorporate geometrical transformation in the graph matching problem and define a clean connection with the well-known Iterative Closes Point (ICP) [14] matching (see section 3.1.2).

3.1.2 Point Set Matching Problem

In Computer Vision, the inference process which trusts on couple of views yields typically the crucial and final stage to combine 3D surfaces or generally n-dimensional point clouds. This is a fundamental task, since through the alignment of such sparse collection of points is possible to derive precious structural information of the scene, as for instance motion, depth and so on. There exist special contexts where this task can be simplified by additional *a priori* knowledge of the problem, but in general it is necessary to consider the worst case in which only a couple of disconnected surfaces constitutes the whole information available.

This class of problems is well-known as *Point Set Registration* or *Point Set Matching*, which is especially treated in Computer Vision, but it could be generalized to further scenarios actually. Furthermore, the estimation of reliable matches between points not only can be considered as a *ex post* processing step in complex systems. Indeed, the point set matching could be just a halfway stage inside some more complex pipelines (*e.g.*, Structure from Motion, Structured Light or further applications as in section 2.2.3). In this further context, the goal consists to improve the accuracy of a model by discarding noisy correspondences or filtering outlier points in the training data.

In this thesis we are interested to present formally the (general purpose) registration problem, introducing and reviewing the most famous approaches residing in literature.

Problem Definition

We give in this section the very essential definition of Point Set Matching, which may be easily applicable in more specific Computer Vision contexts. We define a geometric point as the d-dimensional vector in real space $\mathbf{p}_i \in \mathbb{R}^d$. Let $\mathcal{M} = \{\mathbf{p}_i\}_{i=1}^{n_i}$ and $\mathcal{S} = \{\mathbf{p}_j\}_{j=1}^{n_j}$ two sets of d-dimensional points with respectively n_i and n_j elements. The goal of point matching consists to define a geometric transformation T which moves all the points in the *model* set $\mathcal{M}$ such that the difference with respect to the *scene* set $\mathcal{S}$ is minimized. Typically, the desired transformation is governed by an optimization parameter set Θ, therefore it could be formulated as the function $T_\Theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$, whose resulting mapping may consist of a *rigid* or *non-rigid* transformation. The set of all the derived points is said "registered model" and it is formulated as follows:

$$\mathcal{T}_\Theta(\mathcal{M}) = \{\overline{\mathbf{p}}_k \mid \overline{\mathbf{p}}_k = T_\Theta(\mathbf{p}_i)\ \forall \mathbf{p}_i \in \mathcal{M}\}. \tag{3.12}$$

The Point Set Matching problem consists in the estimation of the set of parameters $\overline{\Theta}$ for the transformation T which aligns optimally $\mathcal{M}$ against $\mathcal{S}$. This principle is similar to a binary matching problem among labelled objects (*i.e.*, combinatorial constrains), but there is a fundamental difference: in this task the result of a transformation could return a set of unknown points, since it does not consist in a permutation of a given structure, in other terms, it is possible that $\mathcal{S} - \mathcal{T}_\Theta(\mathcal{M}) \neq \emptyset$. Therefore, this process does not guarantee to describe directly a one-to-one mapping, by contrast for example with Graph Matching (see section 3.1.1). The Point Set Registration is a complex problem and the majority of the methods are formulated according distance metrics between the points $d : \mathbb{R}^d \rightarrow \mathbb{R}^+$, the resulting process is controlled by an objective function which is based to solve the parameters in $\overline{\Theta}$ minimizing the distance of the mapped points as follows:

$$\overline{\Theta} = \underset{\Theta}{\operatorname{argmin}} \sum_{\mathbf{p}_k \in \mathcal{T}_\Theta(\mathcal{M})} \sum_{\mathbf{p}_j \in \mathcal{S}} d\big(\mathbf{p}_k, \mathbf{p}_j\big). \tag{3.13}$$

Although, this formulation is very minimal and particularly weak in presence of noisy data or outliers. We can propose another more robust version where the distance measures are weighted properly by an ideal function Ψ such that the local configuration of the point set $\mathcal{M}$ is insensitive with respect to very distant points as follows:

$$\overline{\Theta} = \underset{\Theta}{\operatorname{argmin}} \sum_{\mathbf{p}_k \in \mathcal{T}_\Theta(\mathcal{M})} \sum_{\mathbf{p}_j \in \mathcal{S}} \Psi\Big(d\big(\mathbf{p}_k, \mathbf{p}_j\big)\Big). \tag{3.14}$$

This formulation is well-known in statistics since the score objective in (3.14) is actually a M-estimator as well.

Generally, the various registration techniques are addressed to a problem space according the special nature of the data, *i.e.*, which geometrical transformation T_Θ is assumed to be suitable in a given scenario. In section 2.2.1, we introduce in detail two fundamental linear transformations in Computer Vision, which are the well-known *rigid*

and *non-rigid* applications. Therefore, for the problems of Point Set Matching, we categorize the possible methodologies as follows:

- **Rigid registration**: when $T_{\overline{\Theta}}$ describes a *rigid* or *similarity transformation*, therefore the goal consists to estimate the set of optimization parameters $\overline{\Theta} = \{s, \mathbf{R}, \mathbf{t}\}$ subject to $\mathbf{R} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix (*i.e.*, $\mathbf{R}^T = \mathbf{R}^{-1}$, constrained to $\det(\mathbf{R}) = 1$), $s \in \mathbb{R}^+$ is a scaling factor and $\mathbf{t} \in \mathbb{R}^d$ is the translation vector of the point w.r.t. the origin, such that $T_{\overline{\Theta}}(\mathbf{p}) = s\mathbf{R}\mathbf{p} + \mathbf{t}$.
- **Non-Rigid registration**: when $T_{\overline{\Theta}}$ describes a *affine transformation*, therefore the goal consists to estimate the set of optimization parameters $\overline{\Theta} = \{\mathbf{V}, \mathbf{t}\}$ subject to $\mathbf{V} \in \mathbb{R}^{d \times d}$ is a linear transformation matrix and $\mathbf{t} \in \mathbb{R}^d$ the translation vector of the point w.r.t. the origin, such that $T_{\overline{\Theta}}(\mathbf{p}) = \mathbf{V}\mathbf{p} + \mathbf{t}$.

However, in the majority of applications is employed a rigid transformation and the typical distance metric consists in the Euclidean measure $d(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2$ between two points; therefore, the process (3.13) is equivalent in solving a *least squares* problem.

Main Approaches

A very influential method in Computer Vision community to solve Point Set Matching is the well-known Iterative Closest Point method (ICP), which is firstly introduced by Besl and McKay [29] (see Figure 3.2). The main principle of this approach consists to establish the matching between the points from two sets considering each point $\overline{\mathbf{p}}_k \in \mathcal{T}_{\overline{\Theta}}(\mathcal{M})$ has to correspond to the nearest point $\mathbf{p}_j \in \mathcal{S}$. More specifically, the method consists in a rigid registration on Euclidean-based distance metric, which solves a least squares regression according the set of all the matched points $\{(\overline{\mathbf{p}}_k, \mathbf{p}_j)_i\}_{i=1}^N$, and estimating the parameter set $\overline{\Theta}$ including the rotation and translation of the related rigid transformation. This process is iterated until the differences between $\mathcal{T}_{\overline{\Theta}}(\mathcal{M})$ and $\mathcal{S}$ is sufficiently lower that a certain threshold, *i.e.*, all the set is registered. This approach suffers by the problem that all the points in the model point set $\mathcal{M}$ are used, including therefore outliers. Moreover, since the cost function depends strictly by the registered points (which are dynamical entities), then it could change trend during the whole process without guarantees there is reached a local optimum [177]. Indeed, ICP is in general very sensitive at the initialization step, in other terms, by the initial optimization parameters $\overline{\Theta}^{(0)}$. In literature we find several variants to overcome these limits [135], but essentially the common guideline consists to add new structural information in the registration schemes [22]. Alternatively, there exist further advanced optimization models with an approach more heuristic. Steven Gold *et al.* [58] propose a robust point matching method introducing a binary correspondence between the points in the sets $\mathcal{T}_{\overline{\Theta}}(\mathcal{M})$ and $\mathcal{S}$. Their formulation is relaxed in a deterministic annealing and normalized by *softassign* step to obtain a one-to-one mapping, a property which is not guarantee with normal ICP. Moreover, this method models affinity transformation consisting in a non-rigid registration. Chui and Rangarajan [31]

propose a similar method, but modelling the transformation as a *thin plate spline*. The main drawback of this new approach is the geometrical nature of the TPS, which works in 3D space only, *i.e.*, there does not deal onto greater space order $d > 3$ with respect to the robust registration scheme proposed by Gold *et al.* [58].

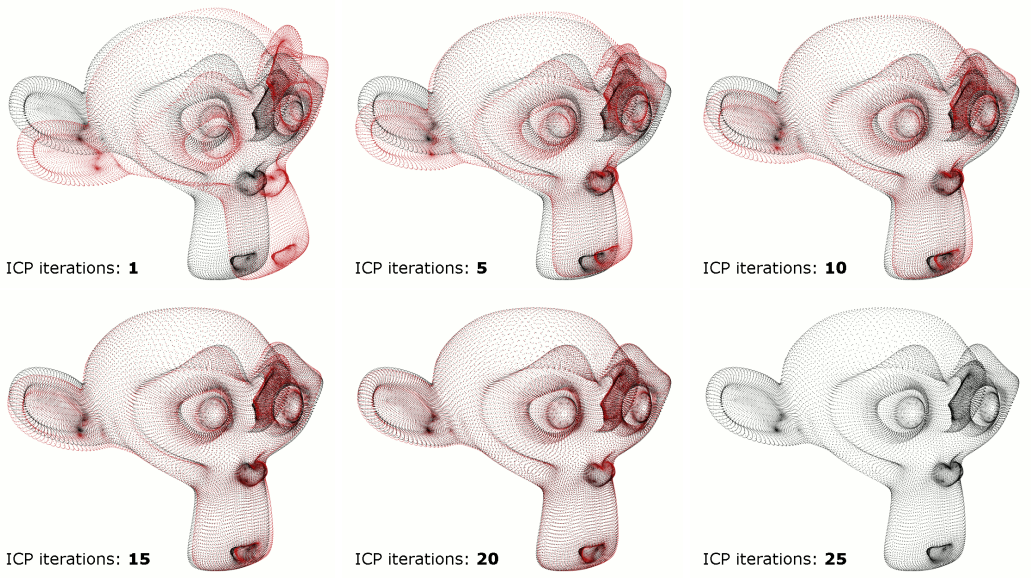


Figure 3.2: *Registration example of the 3D surface which describes a monkey face with ICP [29] method per iterations. The black dots denotes the original points of the point cloud, while the red dots are the aligned point cloud estimated by ICP. In the last image the model fits perfectly with the original point cloud.*

Another well-known method which is used typically for ICP problems is the RANSAC algorithm [44], even if it describes a general approach to fit a model discriminating data points as *inliers* and *outliers*, in other terms, it can be used as a filter to remove noisy data and obtain a more reliable registration. Assuming that a model requires a minimum of N points to solve its parameter set $\overline{\Theta}$, having a training set such as $|\mathcal{M}| \geq N$ the RANSAC method randomly selects a subset $\mathcal{M}^{(t)} \subset \mathcal{M}$ to estimate $\overline{\Theta}^{(t)}$ a time t. Successively, an evaluation strategy scores the capability of the learned model to classify all the points of $\mathcal{M}$ with a suitable way according the kind of application, for example the alignment of $\mathcal{T}_{\overline{\Theta}^{(t)}}(\mathcal{M})$ against $\mathcal{S}$. The process continues iteratively and according the scores recorded in each trial the set with most consensus emerges. The selection can terminate in case there will be found a model with a sufficient acceptable score or after a fixed number of trials is reached. Successively to have obtained a proper set of inliers, a least squares optimization could be run to enhance the solution. Despite RANSAC method is considered a valid approach to inlier selection is affected by two main drawbacks: first, the number of required iterations for the selection of the best consensus set depends strictly by the number of outlier points in the datasets; second, to evaluate the conformity of the model

with respect to the data a valid threshold has to be fixed. The latter is particularly difficult to solve, since if the threshold is too low could be impossible to establish a sufficient reliable set of inliers; conversely, a too large threshold could lead to include noisy points in the consensus set.

In literature there exist some alternative versions of RANSAC method to speed up the velocity of the process, which are meanly categorized according two principles: approaches that improve the verification stage of the model, and techniques that avoid a random selection of the points in the estimation of the model parameters.

In the first group Matas and Chum [107] propose a strategy to filter out in advance wrong hypothesis just selecting a small subset of data whence running the model validation: at the first test failure, the verification is terminated discarding the hypothesis. The main problem of this approach consists in the possibility to encounter false negatives, although experimental results have shown that this trade-off is acceptable with respect to the relevant reduction in execution time. Capel [26] proposes an early termination strategy modelling the assumption that the inliers in a random sample of points follow a hypergeometric distribution. Iteratively, fixing the best hypothesis with N inliers, each new hypothesis is evaluated onto a further subset of points, with the termination criterion in which the probability that the overall number of inliers is more of N is greater than a fixed threshold. Recently, Mates and Chum [108] exploit of the sequential decision-making Walds theory to devise an optimization problem which establishes when the model is good (or bad) and simultaneously decreases the number of verification steps executed.

In the second group, the sampling of the data points is not random, but the extraction is controlled exploiting of additional *a priori* information inferred from the data. The common principle of several methods is the fact that groups of similar instances tend to be inlier points as well. Tordoff and Murray [169] introduce a non-uniform sampling of correspondences in the Maximum Likelihood Estimation Sample Consensus (MLESAC) algorithm [171], a generalization of the original RANSAC method. MLESAC assumes an uniform prior to validate a match, but Tordoff and Murray replaces such priors deriving the probabilities by a quality function of the point matchings. PROSAC [32] method follows a similar approach, where the matches are ordered according their similarity scores as well as larger subsets of tentative correspondences are iteratively used to make hypothesis. Chum *et al.* [33] solve a weak assumption residing in the general termination step of RANSAC method, namely, a model is consistent just when the data sample contains all inliers, which is the main reason because the process is so sensitive when operates with noisy data. The solution is based to generate a constant number of hypothesis just from the inliers of the current best model. This inner RANSAC version have shown experimentally to increase the consensus score rapidly and requiring a lower number of iterations.

3.2 Multi-way Matching Problem

The problem to align the data structures of two objects remains nowadays the fundamental reference tool for recognition tasks. Although, in several contexts of Computer Vision, such as Image Registration [150], Shape Matching [12], Object Recognition [90], Structure from Motion (SfM) [188], Stereo [55] and so on, there may be employed supplementary set of noisy objects, *e.g.*, graphs, images, data points, *etc.*, as several observations of a common subject of interest. Here, we can give just two typical examples, such as the dataset composed by several graph representations related to shots of a same object from different viewpoints, or the sequence of frames registered from a camera installed on mobile robot in indoor/outdoor environments.

In these extended scenarios the matching problem needs to be reformulated in a generalized form of alignment which is spread on the multiple data entities. This is a very challenging task, because with respect to the classical pairwise analysis, in this new multiple setting the several alignments are not independent structures anymore, since they refer to a common cluster of objects. This bond emerges in a further constrain supported by the transformations, which is a consistency property typically refereed to *transitivity*. The fundamental advantage in exploiting of multiple data yields in the the recognition performances, since the impact of the error to align couples of objects can be smoothed by additional information derived from the remaining samples.

In this part of the thesis we introduce the generalizations of the classical pairwise problems of Graph Matching and Point Set Matching presented in detail in the sections 3.1.1 and 3.1.2 respectively. Without loss of generality, we give here a formal definition of these problems, then we review in deep the several related approaches in literature.

3.2.1 Multi-Graph Matching Problem

For the last three decades [36] the investigation concerning the well-known Graph Matching problem (GA) has focused matching techniques confined in an inference process which has mainly considered two graphs only. Therefore, GA is essentially a pairwise analysis, but this aspect may be revealed in a critical weak point when the estimation of the optimal correspondences is compromised by poor quality of the input data, *i.e.*, operating with graphs with high noisy information (both widespread on the links and, if any, the nodes attributes). In the scenario where the available data can trust on several observations of the same object, the Graph Matching problem could be generalized to reduce the bias related to a noisy couple of graphs, by the incorporation of further information carried from the remained instances in whole set: this is the Multi-graph Matching problem.

Problem Definition

In this section we give a formal definition of the Multi-Graph Matching problem as the continuation of the pairwise schema presented in section 3.1.1. Therefore, for a fluent explanation we avoid to reintroduce here some already treated concepts or structures.

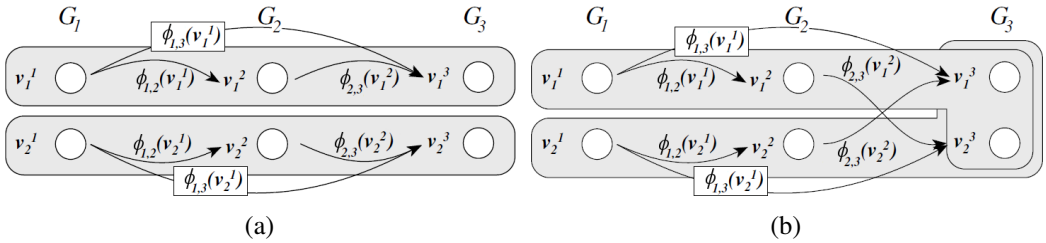


Figure 3.3: *Example of (a) consistent, and (b) inconsistent mapping between three graphs G_1, G_2 and G_3 from two possible starting nodes v_1^1 and v_2^1 of G_1.*

Let $\mathcal{G} = \{G_1, \dots, G_N\}$ be a set of N graphs, we can define a new ideal set which contains all the possible pairwise maps for each couple of graphs $G_i, G_j \in \mathcal{G}$ as follows: $\Phi_{\mathcal{G}} = \{\phi_{ij}\}_{i,j=1}^{N,N}$. With these premises, there would be natural to conclude that solving Multi-Graph Matching consists just to estimate an optimal *matching configuration* $\Phi_{\mathcal{G}}$ such that an energy function is maximized. Unfortunately, the problem is more complex in the multiple setting, since all the possible pairwise maps are not necessarily independent in each other. If we consider just the case of a triple of graphs $G_i, G_j, G_k \in \mathcal{G}$ and respectively the related maps as $\phi_{ij}, \phi_{ik}, \phi_{kj}$, if it holds $\phi_{ik}(v_a^i) = v_z^k$ and $\phi_{kj}(v_z^k) = v_r^j$, *i.e.*, the node $v_a^i \in V_i$ matches with the node $v_z^k \in V_k$ and the latter matches with the node $v_r^j \in V_j$, for transitivity is verified $\phi_{ij}(v_a^i) = v_r^j$ as well. In other terms, the three maps have to be *consistent*, since $\phi_{ij} = \phi_{kj} \circ \phi_{ik}$, *i.e.*, could be described as the functional composition of two boundary maps which include the k-th graph. This constrain, also well-known in literature as *cycle consistency* [190], is the new fundamental ingredient to take in account in the multi-graph matching problem. Let $\Phi_{\mathcal{G}}^* = \{\ \Phi_{\mathcal{G}} \mid \phi_{ij} \in \Phi_{\mathcal{G}}, \phi_{ij} = \phi_{kj} \circ \phi_{ik}\ \forall k = 1, \dots, N; k \neq i, j\ \}$ be the set of all possible matching configurations for $\mathcal{G}$ which are consistent too, the natural extension following the pairwise formulation in (3.1) consists to look for a set of maps in $\Phi_{\mathcal{G}}^*$ which maximize the cumulative score for each couple of graphs from $\mathcal{G}$, hence achieving the general Multi-Graph Matching problem as follows:

$$\overline{\Phi}_{\mathcal{G}} = \arg\max_{\Phi_{\mathcal{G}} \in \Phi_{\mathcal{G}}^*} \sum_{\phi_{ij} \in \Phi_{\mathcal{G}}} S(G_i, G_j, \phi_{ij}). \tag{3.15}$$

The same concepts could be expressed in matricial form with the purpose to describe the general problem as above in a structural domain. Considering the maps as permutation matrices the transitivity constraint can be modelled in a very simply way. Let $\mathbf{P}_{ij}, \mathbf{P}_{ik}, \mathbf{P}_{kj} \in \Sigma^b$ be the permutation matrices related to three graphs in $\mathcal{G}$, if $\mathbf{P}_{ij}$ is consistent with respect to k, then it holds $\mathbf{P}_{ij} = \mathbf{P}_{ik}\mathbf{P}_{kj}$. Therefore, redefining a matching

configuration as $\mathcal{P}_{\mathcal{G}} = \{\mathbf{P}_{ij} \in \Sigma^b\}_{i,j=1}^{N,N}$, namely, the set of all possible permutation matrices for the N graphs in $\mathcal{G}$, the space of *consistent* matching configurations is defined as follows:

$$\mathcal{P}^*_{\mathcal{G}} = \{\ \mathcal{P}_{\mathcal{G}} \mid \mathbf{P}_{ij} \in \mathcal{P}_{\mathcal{G}}, \mathbf{P}_{ij} = \mathbf{P}_{ik}\mathbf{P}_{kj}\ \forall k = 1, \ldots, N; k \neq i, j\ \}\,.$$

Taking in account the score formulation for the pairwise graph matching optimization based on Lawler's QAP (3.3), the resulting Multi-Graph matching problem from the general model (3.15) could be expressed as follows:

$$\overline{\mathcal{P}_{\mathcal{G}}} = \arg\max_{\mathcal{P}_{\mathcal{G}} \in \mathcal{P}^*_{\mathcal{G}}} \sum_{\mathbf{P}_{ij} \in \mathcal{P}_{\mathcal{G}}} \mathrm{vec}(\mathbf{P}_{ij})^T \mathbf{W}_{ij} \mathrm{vec}(\mathbf{P}_{ij}). \tag{3.16}$$

This model represents typically the starting point in several works we find in literature and in particular the methods treated in this thesis.[2] Since this new problem inheritances still a pairwise graph matching, the class of complexity clearly remains in NP-hard [49].

Main Approaches

The formal definitions of the problems (3.15) and (3.16) skip the fundamental aspect concerning how to solve respectively the constraints $\Phi_{\mathcal{G}} \in \Phi^*_{\mathcal{G}}$ and $\mathbf{P}_{ij} \in \mathcal{P}_{\mathcal{G}}$ into practice, *i.e.*, how to work in the permutation space of consistent machining configurations. The several methods presented in this section are just diversified for their way to reach such condition.

The simplistic strategy to deal with Multi-Graph Matching would consist to solve multiple pairwise graph matching problems treating the set of graphs as a pool or sequence [128], in other terms, by a matching chain $G_1 \rightarrow G_2 \rightarrow \ldots \rightarrow G_N$ obtaining a sequence of maps $\phi_{1,2}, \phi_{2,3}, \ldots, \phi_{N-1,N}$. Although, this approach suffers of two main drawbacks: first, it is necessary to impose a specific order between graphs, causing a partial form of transitivity and without taking in account of the overall information available in the set of graphs; second, real-world data contains noise, therefore it is sufficient to estimate a wrong match at the beginning of the sequence to twist critically the next maps with an exponentially propagations of errors along the chain.

Early works in Multi-Graph Matching face the problem by probabilistic methods. The work proposed by Williams *et al.* [186] represents the first approach in this direction, even if it could be considered just a proof of the benefit in exploiting of multiple data than the pairwise solution of this problem. In this framework the cycle consistency is obtained by inducing inference triangles (or graph triples) through the composition of pairwise matching functions in a Bayesian fashion. More specifically, the candidate output node in the matching between two graphs (G_i, G_j) is extracted according the one with maximum probability in a fuzzy-inference matrix, whose entries are determined as the sum of all possible compositions moving an intermediate graphs G_k spanned in the pair.

[2] We skip to present also the extension based on Koopmans-Beckmann's QAP (3.4), which is trivially derivable by replacing the score function in the schema (3.16).

The work of Solé-Ribalta and Serratosa [158] is aimed to solve a well-known problem in literature which is the *common labelling* in a set of attributed graphs. The intermediate step which is necessary to derive the final labelling is just essentially a multi-graph matching problem. The authors devise a probabilistic scheme building a N-dimensional hypercube, whose points are joint probabilities over the nodes of the N graphs and weights are likelihoods of the matches. Denoting with $v_a^i \in V_i$ the a-th node in the i-th graph of the set, the probability is define as follows:

$$p\left(v_a^1, v_b^2, \ldots, v_c^N\right) \;=\; p\left(\phi_{1,2}(v_a^1) = v_b^2 \wedge \phi_{1,3}(v_a^1) = v_p^3 \wedge \ldots \wedge \phi_{N-1,N}(v_c^{N-1}) = v_q^N\right). \tag{3.17}$$

In a certain sense this model is just a generalization of the problem for $N \geq 2$, but if we consider the pairwise case $N = 2$, the hypercube becomes a matrix indeed, which is just the same data structure treated in Double-stochastic relaxation methods as in section 3.1.1. Hypercube on hand, the final step consists to perform a discretization process to obtain the final maps, which are consistent since the probabilistic model assumes dependency among all possibility matches, therefore represents surely an isomorphism too. The authors propose two strategies to solve the computation of the whole structure: first, as an extension of the pairwise Graduated Assignment algorithm [57] in the N-dimensional fashion, which consider the joint probabilities derived from the assignments cost as (3.17); second, as the previous version but the computation of the hypercube is exemplified considering independence between the isomorphism, *i.e.*, by the simply product of probabilities as follows:

$$p\left(v_a^1, v_b^2, \ldots, v_c^N\right) = \prod_{i=1}^{N}\prod_{j=1}^{N}\prod_{a=1}^{n_i}\prod_{b=1}^{n_j} p\left(\phi_{i,j}(v_a^i) = v_b^j\right).$$

Although, the main drawback of this probabilistic scheme is the extreme space complexity due to the managements of the whole hypercube during the learning.

The multi-graph matching begins to be treated as an optimization problem with the work of Pachauri *et al.* [123], which introduce the concept of "permutation synchronization". This process works in an out-of-box fashion, *i.e.*, it does not treat explicitly graphs actually, but require as input a general multi-way matching configuration between ideal objects, independently of their nature and the origin algorithm used to derive it. The relaxation proposed is realized from a spectral perspective by an eigenvector decomposition. Considering the case of squared permutation matrices of dimensionality $O(n^2)$, if the consistency constrain between pairwise transformations is satisfy, *i.e.*, $\mathbf{P}_{ij} = \mathbf{P}_{ik}\mathbf{P}_{kj}\ \forall i, j, k$, then there exists an unknown reference ordering in which the N objects induce own permutations $\mathcal{Q} = \{\mathbf{Q}_k \in \Sigma^b\}_{k=1}^N$ such that

$$\mathbf{P}_{ij} = \mathbf{Q}_i\mathbf{Q}_j^{-1}. \tag{3.18}$$

Therefore the problem consists just to solve the set of transitive alignments $\mathcal{Q}$, which are sufficient to derive whole matching configuration $\mathcal{P}_{\mathcal{G}}(\mathcal{Q}) = \{\mathbf{Q}_i\mathbf{Q}_j^{-1}\}_{i,j=1}^{N,N}$ by (3.18).

In a general scenario we can denote an initial *non-consistent* matching configuration as $\overline{\mathcal{P}_{\mathcal{G}}} = \{\overline{\mathbf{P}}_{ij} \in \Sigma^b\}_{i,j=1}^{N,N}$, therefore the intuitive approach is based just to estimate the alignments which produce permutations that are the closest as possible according a distance measure $d : \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \to \mathbb{R}$ as follows

$$\overline{\mathcal{Q}} = \underset{\mathcal{Q}}{\operatorname{argmin}} \sum_{\substack{\overline{\mathbf{P}}_{ij}\in\mathcal{P}_{\mathcal{G}} \\ \mathbf{Q}_i,\mathbf{Q}_j\in\mathcal{Q}}} d(\overline{\mathbf{P}}_{ij}, \mathbf{Q}_i\mathbf{Q}_j^{-1}). \tag{3.19}$$

The authors propose the general distance metric $d(\mathbf{A},\mathbf{B}) = n - \langle \mathbf{A},\mathbf{B}\rangle$, where $\langle\cdot,\cdot\rangle$ is the inner product for the matrices $\mathbf{A},\mathbf{B} \in \mathbb{R}^{n\times n}$. According the indexing of the data structures in the sets $\mathcal{Q}$, $\mathcal{P}_{\mathcal{G}}(\mathcal{Q})$ and $\mathcal{P}_{\mathcal{G}}$, we can give without loss of generality the equivalent matricial representations as the block matrices $\mathbf{K} \in \{0,1\}^{nN\times n}$, $\mathbf{P}_{\mathcal{G}}(\mathbf{K}) \in \{0,1\}^{nN\times nN}$, $\mathbf{P}_{\mathcal{G}} \in \{0,1\}^{nN\times nN}$ respectively. In this way the formulation of the optimization (3.19) can turn in the maximization problem as follows:

$$\overline{\mathbf{K}} = \underset{\mathbf{K}}{\operatorname{argmax}} \langle \mathbf{P}_{\mathcal{G}}(\mathbf{K}), \mathbf{P}_{\mathcal{G}}\rangle. \tag{3.20}$$

The proposed relaxation for (3.20) is based in the particular form of the matrix $\mathbf{K}$. The space of the permutations Σ^d is orthogonal, *i.e.*, it holds that $\mathbf{Q}_j^{-1} = \mathbf{Q}_j^T$, therefore we can rewrite $\mathbf{P}_{\mathcal{G}}(\mathbf{K}) = \mathbf{K}\mathbf{K}^T$. Moreover, since $\mathbf{P}_{\mathcal{G}}(\mathbf{K})$ is a nN-dimensional rank n symmetric matrix whose non-zero eigenvalues are N, it is possible to approximate such structure by solving a generalized Rayleigh problem factorizing $\mathbf{P}_{\mathcal{G}}(\mathbf{K}) = \mathbf{U}\mathbf{U}^T$ with

$$\mathbf{U} = \sqrt{N}\Big[\mathbf{v}_1\ \mathbf{v}_2\ \ldots\ \mathbf{v}_n\Big],$$

in which the columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are the n leading normalized vectors of $\mathbf{P}_{\mathcal{G}}$. Denoting with $[\mathbf{P}_{\mathcal{G}}(\mathbf{K})]_{i,1}$ the $n \times n$ matrix placed in the i-th row in the first column of $\mathbf{P}_{\mathcal{G}}(\mathbf{K})$, then the final alignment is obtained in permutation space by Hungarian algorithm $\overline{\mathbf{Q}}_i = \textsc{Hungarian}([\mathbf{P}_{\mathcal{G}}(\mathbf{K})]_{i,1})$. This approach has the advance to work very well with noisy data, in particular when the number of available objects is suitable to tolerate weak instances. Although, this process is less scalable since requires a spectral decomposition over a matrix with order $O(n^2N^2)$, which could be a very critical task operating with huge data structures.

The framework proposed by Yan *et al.* [191] is based in a power iteration process alternating the updates of the several permutation matrices, which are solved by an IQP quite similar to (3.3). The principal cost matrix is computed from the composition of three graphs $G_i, G_k, G_j \in \mathcal{G}$ starting from the general maximization problem as follows:

$$\underset{\mathbf{P}_{ik},\mathbf{P}_{ij},\mathbf{P}_{kj}\in\mathcal{P}_{\mathcal{G}}}{\arg\max}\ \operatorname{vec}(\mathbf{P}_{ik})^T\mathbf{W}_{ij}\operatorname{vec}(\mathbf{P}_{ik}) + \operatorname{vec}(\mathbf{P}_{ij})^T\mathbf{W}_{ij}\operatorname{vec}(\mathbf{P}_{ij}) + \operatorname{vec}(\mathbf{P}_{kj})^T\mathbf{W}_{kj}\operatorname{vec}(\mathbf{P}_{kj}).$$

Yan *et al.* [190] proposed another method which can be described as a regularization of the multi-graph matching based on the pairwise problem (3.4) and exploiting of the Lagrange multiplier with the purpose to enforce the cycle consistency as follows:

$$\overline{\mathcal{P}_{\mathcal{G}}} = \underset{\mathcal{P}_{\mathcal{G}}}{\operatorname{argmax}} \sum_{\mathbf{P}_{ij} \in \mathcal{P}_{\mathcal{G}}} \operatorname{vec}(\mathbf{P}_{ij})^T \mathbf{W}_{ij} \operatorname{vec}(\mathbf{P}_{ij}) + \lambda \sum_{k=1}^{N} ||\mathbf{P}_{ik} - \mathbf{P}_{kj}\mathbf{P}_{ij}||_F, \tag{3.21}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier. Since the problem (3.21) is still NP-Hard, the author propose a relaxation strategy introducing a measure of graph consistency, which could be described as the accuracy that an intermediary graph G_k has between all possible pairs of graphs in the set $\mathcal{G}$ as follows:

$$C_{\mathcal{G}}(G_k, \mathcal{P}_{\mathcal{G}}) = 1 - \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|\mathbf{P}_{ik} - \mathbf{P}_{kj}\mathbf{P}_{ij}\|_F}{nN(N-1)/2} \in [0, 1],$$

where $N = |\mathcal{G}|$ is the number of graphs and $n = |V_1| = |V_2| = \ldots = |V_N|$ denotes the common number of nodes of all the graphs. The final iterative process consists to update the permutation matrix $\mathbf{P}_{ij}^{(t)} = \mathbf{P}_{ik}^{(t-1)}\mathbf{P}_{kj}^{(t-1)}$ at time t for the graph G_k which maximizes the regularized objective $C_{\mathcal{G}}(G_k) + (1-\lambda)\operatorname{vec}(\mathbf{P}_{ij}^{(t)})^T \mathbf{W}_{ij} \operatorname{vec}(\mathbf{P}_{ij}^{(t)})/J_{max}^{(0)}$, where $J_{max}^{(0)}$ is the constant energy scale reference (maximum score measured at the beginning of the learning process). However, a drawback of this method consists in the lack of guarantee to provide fully consistency after the convergence, *i.e.*, $\overline{\mathcal{P}_{\mathcal{G}}} \notin \mathcal{P}_{\mathcal{G}}^*$, requiring smoothing methods [191] or by maximum span tree [53] to rectify the final solution. Moreover, in the same way like other methods as above [158, 186, 190], the initial solution has to be inconsistent, otherwise the process cannot work properly.

Yan *et al.* [192] overcome these limits proposing a method which is a sort of fusion from the last works [190, 191]. In fact, it is not necessary to compute all N^2 permutations matrices in a matching configuration, since for the transitivity constrain any transformations $\mathbf{P}_{ij} = \mathbf{P}_{ir}\mathbf{P}_{rj} = \mathbf{P}_{ri}^T\mathbf{P}_{rj}$ with $r \neq i, j$, *i.e.*, all the pairwise permutations can be derived fixing a common reference order r. From this result, the solution consists in an alternating optimization process by establishing a *reference graph* to drive the global consistency during the learning. The latter is derived as the graph which reveals the maximum consistency in the initial solution $G_r = \underset{G_k \in \mathcal{G}}{\operatorname{argmax}} C_{\mathcal{G}}\left(G_k, \mathcal{P}_{\mathcal{G}}^{(0)}\right)$. Listing the pairwise consistencies of all graphs w.r.t. the reference, the iterative process is driven by selecting in ascending order that graph G_u and updating the related $\mathbf{P}_{ur}^{(t)}$ just fixing the remaining $N-2$ variables $\{\mathbf{P}_{fr}^{(t)}\}_{f=1, f\neq r,u}^{N}$. The new version at time t is discarded if the so-far best objective score $J(\mathbf{P}_{ur}^{(t)})$ is not increased. The authors proposed two versions based on this general strategy, which are distinguished according the way is formulated the maximization problem, such as through a global weight matrix or proposing a less space expensive factorization.

The methods in our review [158, 191, 192] can be defined as *affinity score*-driven, since firstly there is generated a compact set of underlying pairwise matching variables,

then an objective function to measure the overall affinity score is maximized. Instead the method [123] can be categorized as *pairwise matching consistency*-driven, because starting from an initial set of pairwise solutions given from an independent solver, it may introduce or increase the overall consistency as a rectification step. Although, both these two strategies have an own weak point: in the former, if the bases of variables contain noise or they are lightly corrupted, such errors would lead the learning process to enhance the distortion towards the final solution; conversely, in the latter the assumption that an unknown algorithm provides always an inconsistent initial solution is a requirement pretty awkward. Recently, Yan *et al.* [189] propose a method that in a certain way combines the advantage of these two strategies for a flexible process which exploits of the matching consistency as a regularizer for the overall affinity score. In other terms, since the affinity score could be used as an indicator for the semantic of the matching in the early iterations, when it begins to be less informative the consistency of the matching can be used as a regularizer to unweighed the contribute carried from possible noisy graphs in the overall score.

3.2.2 Multi-Point Set Matching Problem

The challenging problem to estimate a geometrical transformation between images, shapes or more generally point clouds is extensively treated in literature and with particular interest to solve such task through disjointed pairs of objects [77]. Here, by contrast with other similar problems, *e.g.*, Graph Matching, the rigid or affine transformation can be derived even exploiting of partial subsets extracted from whole available data. For example, to estimate planar linear transformations is sufficient a reduced set of 4 key points from each input images actually. Moreover, the geometrical alignments are not forced to describe an one-to-one mapping between the chosen points of two objects. Therefore, this problem itself gives the opportunity to devise easily strategies under such relaxed conditions in order to smooth and filter noisy data as well as removing outlier points. Although, the majority of research in this field is focused in the pairwise setting of the problem, which is clearly generalizable to a multiple point registration as well. This request is two-fold necessary: first, there are specific problems in this field which require explicitly the alignment of multiple set of objects, as for example the creation of statistical shape models [37]; second, by the introduction of a global consistency among all the transformations in a multiple set of objects, the impact that noisy data could affect onto the derived registrations can be more tolerable.

Problem Definition

In this section we introduce the Multi-Point Set Registration problem recalling notation and fundamental concepts which there have been already treated in the pairwise schema presented in section 3.1.2, therefore we avoid to be too verbose by reintroducing such

details again.

The initial step to extend the classical pairwise Point Set Matching consists to generalize the data in multiple setting. Let $\mathcal{C}_i = \{\mathbf{p}_s \in \mathbb{R}^d\}_{s=1}^{n_i}$ be a point cloud of n_i real-valued points with a common number of d dimensions, we can formalize the input of the problem as a set of N different point clouds as follows: $\mathcal{Z} = \{\mathcal{C}_i\}_{i=1}^{N}$. According a specific application of the problem, we can assume the existence of an unique and common geometrical transformation function T, which is applied to all possible N^2 optimization set Θ_{ij} related to each pair of point clouds $\mathcal{C}_i, \mathcal{C}_j \in \mathcal{Z}$ whence is derived the registered set of $\mathcal{C}_j$ by the usual transformation $\mathcal{T}_{\Theta_{ij}}(\mathcal{C}_i)$. Therefore, we can define an *optimization set configuration* for the overall input data as the set $\Theta_{\mathcal{Z}} = \{\Theta_{ij}\}_{i,j=1}^{N,N}$. Since we are defining a process which spreads information of a multiple set of point clouds to estimate pairwise transformations, there has to be supported some consistency property in the final solution. We can express this concept in a very general way just stating that an optimization set Θ_{ij} is *consistent* when for each $k = 1, \ldots, N$ holds the *transitive constraint* as follows: $\Theta_{ij} \sim \Theta_{ik} \circ \Theta_{kj}$.[3] Exploiting of this essential formalism, the set of all possible optimization set configurations on $\Theta_{\mathcal{Z}}$ which are globally consistent is defined as follows:

$$\Theta_{\mathcal{Z}}^{*} = \{\Theta_{\mathcal{Z}} |\ \Theta_{ij} \in \Theta_{\mathcal{Z}}, \Theta_{ij} \sim \Theta_{ik} \circ \Theta_{kj}\ \forall k = 1, \ldots, N; k \neq i, j\ \}.$$

At this point, we have introduced all the ingredients to formulate finally the Multi-point Set Matching in terms of the following optimization problem:

$$\overline{\Theta}_{\mathcal{Z}} = \underset{\Theta_{\mathcal{Z}} \in \Theta_{\mathcal{Z}}^{*}}{\arg\min} \sum_{\Theta_{ij} \in \Theta_{\mathcal{Z}}} \sum_{\mathbf{p}_s \in \mathcal{T}_{\Theta_{ij}}(\mathcal{C}_i)} \sum_{\mathbf{p}_r \in \mathcal{C}_j} d\big(\mathbf{p}_s, \mathbf{p}_r\big). \tag{3.22}$$

The goal reacted by the problem (3.22) consists to estimate a consistent optimization set configuration such that the distances between the points of all possible pairwise transformations are minimized. It is worth to be noted that if we consider transformations where location, scale and rotation are removed (*i.e.*, Absolute Orientation Problem [70] or Procrustes Analysis [59]), in the optimization set remains the transformation matrix $\Theta_{ij} = \{\mathbf{T}_{ij}\}$ only, in other terms, the points are derived by $\mathbf{p}_s = T_{\Theta_{ij}}(\mathbf{p}_r) = \mathbf{T}_{ij}\mathbf{p}_r$. This further matricial scheme becomes a problem where the goal consists just to estimate a set $\overline{\Theta}_{\mathcal{Z}}$ of transitive geometrical transformations, which is just a similar approach we have already treated with the permutation synchronization in Multiple Graph Matching (see section 3.1.1).

Main Approaches

In this section we are interested to review the literature just focusing on a specific subclass of transformation problems relevant for the topics introduced in this thesis. More specifically, we consider rigid registrations among shapes in terms of point correspondences.

[3] This expression has to be considered just a special notation for this thesis, with the purpose to denote in general terms the property of consistency/transitivity in geometrical transformations as well.

This is a specific task well-known in Computer Vision community as *Procrustes Analysis*, which is deeply studied in statistics and shape analysis [59, 134]. The related extension of such problem which considers set of point clouds greater than the pairwise case $N = 2$ is called *Generalized Procrustes Analysis*. Typically, this new problem is solved selecting initially a reference shape $\mathcal{C}_r \in \mathcal{Z}$ and registering in an "alternating process" each other shapes averaging the reference such that the global alignments are optimized by a score function [41].

Krishnan *et al.* [84] propose a simultaneous global registration approach for multi-view 3D point sets where the correspondences between overlapping scans is known. The main idea consists just in solving an unconstrained optimization problem on a constrained manifold formed by *N*-fold product of orthogonal groups. The solution is obtained by an Newton-type iterative process governed by the reduction of cost functions on such smooth manifold. The latter represents the principal power of this method, since such special manifold can incorporate transformation groups in Computer Vision as $\mathbb{SO}_3$ or $\mathbb{SE}_3$, which are provided of Lie algebra, notions used by the authors to model the geometrical nature of the problem. Despite this multi-registration method reveals good performances with real world data, it is limited to treat just instances of 3D point clouds.

The work of Wen *et al.* [184] generalizes the Absolute Orientation Problem to deal with points of any dimensions. Considering the case where all the N point clouds are composed by n points in $\mathbb{R}^d$, the general goal consists to estimate a fixed reference point set $\mathcal{P}_r = \{\mathbf{r}_i\}_{i=1}^n$ and optimization $\Theta_r = \{\Theta_j = (\mathbf{T}_j, \mathbf{t}_j, c_j)\}_{j=1}^N$ sets, such that the following score is minimized:

$$\underset{\mathcal{P}_r,\, \Theta_r}{\arg\min} \frac{1}{N} \sum_{j=1}^{N} \left[\frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{p}_i^j - c_j \mathbf{T}_j \mathbf{r}_i - \mathbf{t}_j \right\|_2^2 \right]. \tag{3.23}$$

The solution is formulated by a reduced gradient process devising a total least squares fitting algorithm. The approach is based in two steps: first, iteratively computing $\Theta_r^{(t)}$ at time t by SVD algorithm for Procrustes Analysis; second, computing the optimum minimum value $\mathbf{x}_0^{(t)}$ of the mean squared error in (3.23) which updates each point in $\mathcal{P}_r^{(t)}$. The convergence is reached when the global error is lower than a fixed threshold. Recently, Chaudhury *et al.* [27] solve another similar work addressed to realign globally overlapped patches of points against an unique scene set, but formulating the problem as a semidefinite program. Nevertheless, there remains a fundamental and common drawback of all these last approaches, which essentially resides being governed by an unique reference structure, which could induce noise in the alignments. Some different proposals are based to update alternating the reference frame meanwhile the global alignments are solved, *i.e.*, either by means or adaptive strategies. Although, such solutions remain iterative processes, which are in general schemes that could not give guarantee both in convergence and in achievement of a global minimum.

Pizarro and Bartoli [129] propose a statistical cost function which involves the transformation against an unknown reference shape to maximize the overall likelihood. The formulation is derived from the algebraic geometry applied on Sum Of Squares (SOS)

functions [126], which is a fundamental tool to find the global bounds for polynomials that describe equality and inequality constrains. The problem is proposed as a SOS program (SOSP) relaxation, which is easily converted in an equivalent Semidefinite program (SDF) in order to exploit of external convex optimization tools as final solvers. Their process can find always a global minimum and it supports both 2D and 3D shapes for Euclidean and similarity transformations.

Recently, Bernard *et al.* [13] propose a solution for solving the multi-alignment GPA as a very flexible generalization with respect to the classical techniques [27, 129], which enforces global self-consistency through a closed-form formulation in a semidefinite program for orthogonal pairwise transformations. Their method not only overcomes the disadvantage of iterative processes being an eigenvalue decomposition, but supports invertible linear transformations too, which includes similarity, Euclidean and rigid registrations; furthermore, such strategy finds even applications to synchronize permutation-based transformations [123]. Considering to have available from an external process the initial unsynchronized set of all pairwise linear transformations $\{\mathbf{T}_{ij}\}_{i,j=1}^{N,N}$ related to N points clouds, then there exists an unknown reference coordinate frame $*$ such that $\mathbf{T}_{ij} = \mathbf{T}_{i*}\mathbf{T}_{*j} = \mathbf{T}_{i*}\mathbf{T}_{j*}^{-1}$ for all i, j. In this way the global consistency of the linear and invertible transformations is guarantee computing the transitive block matrix $\mathbf{W} = [\mathbf{T}_{ij}]_{i,j=1}^{N,N} = \mathbf{U}_1\mathbf{U}_2$, solving $\mathbf{U}_1 = [\mathbf{T}_{1*} \cdots \mathbf{T}_{N*}]^T$ or $\mathbf{U}_2 = [\mathbf{T}_{1*}^{-1} \cdots \mathbf{T}_{N*}^{-1}]^T$. This model is very similar than [123] with the difference that does not require necessary the orthogonally property $\mathbf{T}_{i*}^{-1} = \mathbf{T}_{i*}^T$ (see section 3.2.1). Posing $\mathbf{Z} = \mathbf{W} - N\mathbf{I}$ is possible to state that the structure $\mathbf{U}_1$ can be computed just finding the d-dimensional null space of $\mathbf{Z}$. Therefore, performing a singular value decomposition as $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the final solution consists to extract the d columns of $\mathbf{V}$ reconstructing just $\mathbf{U}_1$. This formulation holds if we consider perfect information in the problem, *i.e.*, absence of noisy data $\mathbf{Z}$. Therefore, to cope with real world contexts, *i.e.*, assuming the presence of noise in the structure $\overline{\mathbf{Z}}$, the process can be rewritten as a least-squares transformation synchronization finding $\overline{\mathbf{U}}_1$ which minimizes the Frobenius norm $\left\|\overline{\mathbf{Z}}\overline{\mathbf{U}}_1\right\|_F^2$. In this case, the alignments $\overline{\mathbf{U}}_1 = [\overline{\mathbf{T}}_{1*} \cdots \overline{\mathbf{T}}_{N*}]^T$ are retrieved extracting the d smallest singular values from the spectral decomposition of $\overline{\mathbf{Z}}$. Moreover, the authors show by just reformulating the initial transformation matrices of the problem, that is possible to solve affine, similarity, Euclidean and rigid transformations. Whilst this method is extremely suitable in various applications, it suffers of some drawbacks: the process can be less scalable in the scenario of high values of $O(d)$ getting the spectral decomposition very time expensive; the retrieved transitive alignments could not be always invertible transformations in presence of high noisy data.

II

Transformation Synchronization

4

Transitive Assignment Kernels for Structural Classification

Kernel methods provide a convenient way to apply a wide range of learning techniques to complex and structured data by shifting the representational problem from one of finding an embedding of the data to that of defining a positive semi-definite kernel. One problem with the most widely used kernels is that they neglect the locational information within the structures, resulting in less discrimination. Correspondence-based kernels, on the other hand, are in general more discriminating, at the cost of sacrificing positive-definiteness due to their inability to guarantee transitivity of the correspondences between multiple graphs.

In this chapter we adopt a general framework for the projection of (relaxed) correspondences onto the space of transitive correspondences, thus transforming any given matching algorithm onto a transitive multi-graph matching approach. The resulting transitive correspondences can then be used to provide a kernel that both maintains locational information and is guaranteed to be positive-definite. Experimental evaluation validates the effectiveness of the kernel for several structural classification tasks.

4.1 Introduction

Graph-based representations have proven invaluable in several application domains due to their ability to characterize complex ensembles in terms of parts and binary relations. Concrete examples include the use of graphs to represent shapes [154], metabolic networks [76], protein structure [72], and road maps [78]. However, the expressive power of graphs comes at the cost of a reduced pattern analysis toolset available to the practitioner. In fact, our ability to analyse data abstracted in terms of graphs is severely limited by the restrictions posed by standard feature-based paradigm dominating pattern recognition techniques, which require data to be representable in a vectorial form.

There are two reasons why graphs are not easily reduced to a vectorial form. First, unlike the components of a vector, there is no canonical ordering for the nodes in a graph, requiring correspondences to be established as a prerequisite for analysis. Second, the variation in the graphs of a particular class may manifest itself as subtle changes in struc-

ture. Hence, even if the nodes or the edges of a graph could be encoded in a vectorial manner, the vectors would be of variable length, thus residing in different spaces.

The first 30 years of research in structural pattern recognition have been mostly concerned with the solution of the graph matching problem as the fundamental means of assessing structural similarity [36]. With the correspondences at hand, similarity-based recognition and classification techniques can be used. Alternatively, graphs can be embedded in a low-dimensional pattern space using either multidimensional scaling or nonlinear manifold learning techniques.

Another alternative is to extract feature vectors from the graphs providing a pattern-space representation by extracting structural or topological features. For example, spectral features extracted from the singular value decomposition of the graph Laplacian have been proven effective [52,103,173,187]. For an overall survey about the current state-of-the-art in the graph matching problem, refers to the work by Livi and Lizzi [98].

4.1.1 Graph Kernels

The famous kernel trick [145] has shifted the problem from the vectorial representation of data, which now becomes implicit, to a similarity representation. This has allowed standard learning techniques to be applied to data for which no easy vectorial representation exists. Once we define a positive semi-definite kernel $k : X \times X \to \mathbb{R}$ on a set X, there exists a map $\phi : X \to \mathcal{H}$ into a Hilbert space $\mathcal{H}$, such that $k(x, y) = \phi(x)^T\phi(y)$ for all $x, y \in X$. Also, given the kernel value between $\phi(x)$ and $\phi(y)$ one can easily compute the distance between them by noting that $\|\phi(x) - \phi(y)\|_2^2 = \phi(x)^T\phi(x) + \phi(y)^T\phi(y) - 2\phi(x)^T\phi(y)$. Thus, any algorithm that can be formulated in terms of dot products between the input vectors can be applied to the implicitly mapped data points through the direct substitution of the kernel for the dot product. For this reason, in recent years the structural pattern recognition field has witnessed an increasing interest in graph kernels. However, due to the rich expressiveness of graphs, this task has also proven to be difficult, with the problem of defining *complete* kernels, *i.e.*, ones where the implicit map ϕ is injective, sharing the same computational complexity of the graph isomorphism problem [49].

One of the most influential works on structural kernels is the definition of the class of R-convolution kernel proposed by Haussler [67]. Here, graph kernels are computed by comparing the similarity of the basic elements for a given decomposition of the two graphs. Depending on the decomposition chosen, we obtain different kernels. Most R-convolution kernels simply count the number of isomorphic substructures in the two graphs. For example, Kashima *et al.* [79] compute the kernel by decomposing the graph into random walks, while Borgwardt *et al.* [18] have proposed a kernel based on shortest paths. Here, the similarity is determined by counting the numbers of pairs of shortest paths of the same length in a pair of graphs. Shervashidze *et al.* [151] have developed a subtree kernel on subtrees of limited size, where the number of subtrees common between two graphs is computed efficiently using the Weisfeiler-Lehman graph invariant. Recently, Kriege *et al.* [83] proposed that a kernel based on the number of isomorphisms between pairs of subgraphs, while Neumann *et al.* [120] have introduced the concept of

propagation kernels to handle partially labelled graphs through the use of continuous-valued vertex attributes.

4.1.2 Assignment Kernels

One drawback of these kernels is that they neglect the locational information for the substructures in a graph. In other words, the similarity does not depend on the relationships between substructures. As a consequence, these kernels cannot establish reliable structural correspondences. This limits the precision of the resulting similarity measure. Ong *et al.* [122] introduce several kernel methods about indefinite kernel for general structures, while Geibel *et al.* [73, 74] give a solution to deal with not positive semidefinited kernel based on Schur-Hadamard Inner Product applied on graphs. Further, Schietgat *et al.* [144] propose a graph metric which is based on the maximum common subgraph, while in [115] the authors exploit indefinte maximum common subgraph kernels using the potential of support vector machine for indefinite matrices, extending the work proposed by Hochreiter and Obermayer [69]. Another interesting solution described by Fröhlich *et al.* [48] presents alternative optimal assignment kernels. Here, each pair of structures is aligned before comparison. Another example of alignment-based kernels are the edit-distance-based kernels introduced by Neuhaus and Bunke [118]. Here, the alignments obtained from graph-edit distance are used to guide random walks on the structures being compared.

Unfortunately, the introduction of the alignment step results in a kernel that is not positive definite in general [181]. The problem arises from the fact that alignments are not in general transitive. In other words, if σ is the vertex-alignment between graph G_a and graph G_b, and π is the alignment between graph G_b and graph G_c, in general we cannot guarantee that the optimal alignment between graph G_a and graph G_c is $\pi \circ \sigma$. Lacking positive definiteness the optimal assignment kernels cannot be guaranteed to represent an implicit embedding into a Hilbert space. However, they have proven to be very effective in classifying structures.

4.1.3 Multi-Graph Matching

The problem of estimating a transitive set of correspondences between structures, known as the multi-graph matching problem, has received much less attention by the research community than pairwise matching. One of the earliest work, due to Williams *et al.* [186], imposes the transitive vertex-matching constraint in a softened Bayesian manner, inducing inference triangles by forming fuzzy compositions of pairwise matching functions. Sole-Ribalta and Serratosa [158] extended the Graduated Assignment algorithm [56] to the multi-graph scenario by raising the assignment matrices associated to pair of graphs to assignment hypercube, or tensors, between all the graphs. For computational efficiency, the hypercube is constructed via sequential local pair matching. More recently, Yan *et al.* [191, 192] proposed a new framework explicitly extending the Integer Quadratic Programming (IQP) formulation of pairwise matching to the multi-graph matching scenario.

The resulting IQP is then solved through alternating optimization approach. Pachauri *et al.* [123], on the other hand, synchronize a given set of assignments through a spectral relaxation.

4.1.4 Contribution

In this chapter we want to investigate the use of multi-graph matching techniques in the context of graph kernels. By forcing the correspondences between the structures under study to satisfy transitivity, we obtain an alignment kernel that, not only is positive definite, but also makes use of more reliable locational information obtained through the enforcement of global consistency constraints. In fact, when the alignments are transitive, there is a common simultaneous alignment of all the graphs. Under this alignment, the kernel is simply the sum over all the vertex/edge kernels, which is positive definite since it is the sum of separate positive definite kernels.

Here, we adopt an approach similar to Pachauri *et al.* [123] in avoiding the definition of a specific multi-graph matching algorithm. Rather, we project a set of (possibly relaxed) assignments to the set of transitive correspondences. Transformation synchronization techniques such as this have been proven effective in several fields due to their effectiveness, their ability to leverage the state of the art in pairwise transformation estimation, and their computational efficiency [64, 174]. The proposed synchronization technique shares some similarities with [123], but we adopt a different relaxation scheme that does not result in a generalized low rank Rayleigh problem, but can however be solved with a projected power method, avoiding the requirement for an eigendecomposition of the matching tensor.

4.2 Projection on the Transitive Alignment Space

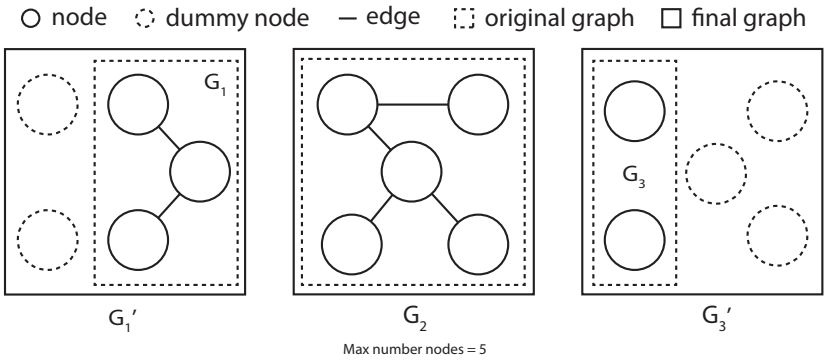
Let $G_1, G_2, \ldots, G_N$ be graphs and let $\mathbf{P}_{ij}$ for $i, j = 1, \ldots, N$ be a matrix matching vertices in G_i to vertices in G_j obtained with any pairwise matching algorithm. Here, we assume that $(\mathbf{P}_{ij})_{vw}$ expresses a likelihood that node v in G_i is matched to node w in G_j, but is not required to represent a permutation, and can be in a relaxed space such as the space of doubly stochastic matrices. Our goal is to find a set of permutation matrices $\overline{\mathbf{P}}_{ij} \in \Sigma_n$ (with Σ_n the permutation space and $i, j = 1, \ldots, N$) as similar as possible, in the least squares sense, to $\mathbf{P}_{ij}$, which satisfy the transitivity constraint. Namely,

$$\overline{\mathbf{P}}_{ij}\overline{\mathbf{P}}_{jk} = \overline{\mathbf{P}}_{ik} \qquad \forall i, j, k = 1, \ldots, N. \tag{4.1}$$

In order to do this first we force the graphs all to the same size n by padding them with dummy disconnected nodes to the maximum size of all the graphs of the set (see figure 4.1).

Once the graphs are all of the same size, we can enforce transitivity through the introduction of an unknown reference canonical ordering and the alignment matrices $\mathbf{Q}_i \in \Sigma_n$

Figure 4.1: *Graphical example about the refinement task of our datasets. In the figure, the set is composed of three graphs G_1, G_2 and G_3. The maximum number of nodes is 5 (the second graph), hence we add two disconnected nodes in G_1 and three in G_3 in order to obtain respectively the extended graphs G'_1 and G'_2. The final dataset with the same number of nodes $n = 5$ is composed by the graphs set G'_1, G_2 and G'_3.*



$i = 1, \ldots, N$ that map vertices in G_i to the reference order. With these matrices to hand we set $\overline{\mathbf{P}}_{ij} = \mathbf{Q}_i \mathbf{Q}_j^T$. Note that there is no lack in representation power, as the transitivity constraint guarantees the existence of such canonical ordering. In fact, let for example $\mathbf{Q}_i = \overline{\mathbf{P}}_{i1}$. For transitivity, we have

$$\overline{\mathbf{P}}_{ij} = \overline{\mathbf{P}}_{i1}\overline{\mathbf{P}}_{1j} = \overline{\mathbf{P}}_{i1}\overline{\mathbf{P}}_{j1}^T = \mathbf{Q}_i \mathbf{Q}_j^T. \tag{4.2}$$

Furthermore, such canonical ordering is not unique, since for any permutation matrix $\mathbf{P} \in \Sigma_n$, we have

$$\overline{\mathbf{P}}_{ij} = \mathbf{Q}_i \mathbf{Q}_j^T = (\mathbf{Q}_i \mathbf{P})(\mathbf{P}^T \mathbf{Q}_j^T)\,. \tag{4.3}$$

With the canonical ordering representation the projection on the transitive space of permutations cast as the following minimization process

$$\begin{aligned}
&\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\min} \sum_{i,j=1}^{N} \left\|\mathbf{P}_{ij} - \mathbf{Q}_i \mathbf{Q}_j^T\right\|_F^2 = \\
&\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\min} \sum_{i,j=1}^{N} \left(\|\mathbf{P}_{ij}\|_F^2 + \left\|\mathbf{Q}_i \mathbf{Q}_j^T\right\|_F^2 - 2\mathrm{Tr}(\mathbf{Q}_j \mathbf{Q}_i^T \mathbf{P}_{ij})\right) = \\
&\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\min}\, 2N^2 n - 2 \sum_{i,j=1}^{N} \mathrm{Tr}(\mathbf{Q}_i^T \mathbf{P}_{ij} \mathbf{Q}_j) = \\
&\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\min}\, 2N^2 n - 2 \sum_{i,j=1}^{N} \mathrm{vec}(\mathbf{Q}_i)^T \left(\mathbf{I} \otimes \mathbf{P}_{ij}\right) \mathrm{vec}(\mathbf{Q}_j)\,,
\end{aligned} \tag{4.4}$$

where $\|\cdot\|_F$ is the Frobenius matrix norm, $\mathbf{I}$ is the identity matrix and Tr is the linear trace operator.

This is equivalent to the following Integer Quadratic Problem

$$\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\max} \underbrace{\begin{pmatrix} \mathrm{vec}(\mathbf{Q}_1) \\ \mathrm{vec}(\mathbf{Q}_2) \\ \vdots \\ \mathrm{vec}(\mathbf{Q}_N) \end{pmatrix}^T}_{\mathrm{vec}(\mathbf{Q})^T} \underbrace{\begin{pmatrix} I\otimes\mathbf{P}_{11} & I\otimes\mathbf{P}_{12} & \dots & I\otimes\mathbf{P}_{1N} \\ I\otimes\mathbf{P}_{21} & I\otimes\mathbf{P}_{22} & \dots & I\otimes\mathbf{P}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ I\otimes\mathbf{P}_{N1} & I\otimes\mathbf{P}_{N2} & \dots & I\otimes\mathbf{P}_{NN} \end{pmatrix}}_{\mathbf{\Pi}} \underbrace{\begin{pmatrix} \mathrm{vec}(\mathbf{Q}_1) \\ \mathrm{vec}(\mathbf{Q}_2) \\ \vdots \\ \mathrm{vec}(\mathbf{Q}_N) \end{pmatrix}}_{\mathrm{vec}(\mathbf{Q})}, \tag{4.5}$$

where $\otimes$ represents the Kronecker product. Note that if the pairwise matches are symmetric, *i.e.*, $\mathbf{P}_{ij} = \mathbf{P}_{ji}^T$, then $\mathbf{\Pi}$ is symmetric as well. However, as in all quadratic problem, $\mathbf{\Pi}$ (and thus $\mathbf{P}_{ij}$) can be made symmetric without affecting the result.

Our proposal is to relax this to the problem

$$\begin{aligned} \text{maximize} \quad & \mathbf{x}^T\mathbf{\Pi}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in (S_n)^N, \end{aligned} \tag{4.6}$$

where S_n is the unit sphere in $\mathbb{R}^n$, and then project the solution to $(\Sigma_n)^N$ in order to obtain the alignment matrices $\mathbf{S}_i$ (which differs from the $\mathbf{Q}_i$ seen before since we are working on a relaxed space) and, consequently, the transitive permutation matrices $\overline{\mathbf{P}}_{ij} = \mathbf{S}_i\mathbf{S}_j^T$.

We solve (4.6) efficiently through a power iteration projected to $(S_n)^N$ by noting that the gradient of the quadratic form can be computed in terms of multiplications and additions of the matching and alignment matrices:

$$\begin{aligned} \mathbf{\Pi}\mathbf{x} &= \begin{pmatrix} \mathbf{I}\otimes\mathbf{P}_{11} & \mathbf{I}\otimes\mathbf{P}_{12} & \dots & \mathbf{I}\otimes\mathbf{P}_{1N} \\ \mathbf{I}\otimes\mathbf{P}_{21} & \mathbf{I}\otimes\mathbf{P}_{22} & \dots & \mathbf{I}\otimes\mathbf{P}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}\otimes\mathbf{P}_{N1} & \mathbf{I}\otimes\mathbf{P}_{N2} & \dots & \mathbf{I}\otimes\mathbf{P}_{NN} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^N \mathbf{P}_{1i}\mathbf{X}_i \\ \sum_{i=1}^N \mathbf{P}_{2i}\mathbf{X}_i \\ \vdots \\ \sum_{i=1}^N \mathbf{P}_{Ni}\mathbf{X}_i \end{pmatrix}, \end{aligned} \tag{4.7}$$

where $\mathbf{x}^T = (\mathbf{x}_1^T, \dots, \mathbf{x}_N^T)^T$ expresses the N spherical components of $\mathbf{x}$, and $\mathbf{X}_i$ is the matrix representing the current relaxation of $\mathbf{Q}_i$, for which we have $\mathbf{x}_i = \mathrm{vec}(\mathbf{X}_i)$.

Hence, we can maximize (4.6) by iterating the recurrence

$$\mathbf{X}_i^{(t+1)} = \frac{\sum_{j=1}^N \mathbf{P}_{ij}\mathbf{X}_j^{(t)}}{\left\|\sum_{j=1}^N \mathbf{P}_{ij}\mathbf{X}_j^{(t)}\right\|_F}. \tag{4.8}$$

Once the matrices $\mathbf{X}_i$ are at hand, we obtain the closest (in least squares sense) permutations $\mathbf{Q}_i$ by solving N maximum bipartite assignment problems.

4.3 Transitive Assignment Kernel

With transitive matches to hand, we follow Fröhlich *et al.* [48] in the definition of an assignment kernel between graphs $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$: we define two sets of kernels, one $K_v : V_i \times V_j \to \mathbb{R}$ for the vertices, and one $K_e : V_i^2 \times V_j^2 \to \mathbb{R}$ for the edges and fuse them with the transitive correspondence $\pi_{ij} : V_i \to V_j$ encoded in $\overline{\mathbf{P}}_{ij}$, to obtain the *Transitive Assignment Kernel*:

$$\mathrm{TAK}(G_i, G_j) = \sum_{v \in V_i} K_v\big(v, \pi_{ij}(v)\big) + \sum_{v \in V_i} \sum_{w \in V_i} K_e\big((v,w), (\pi_{ij}(v), \pi_{ij}(w))\big). \tag{4.9}$$

Here, both kernels are assumed to be positive semidefinite and symmetric. In our experiments we used the dot product between Heat Kernel Signatures [162] (HKS) for the vertex kernel K_v. More precisely, given an undirected graph G of n nodes, let $\mathbf{A} = (a_{ij})$ the $n \times n$ adjacency matrix (where a_{ij} is the weight of the edge between the nodes i and j in G) and $\mathbf{D}$ the degree matrix, we compute the related $n \times n$ Laplacian matrix $\mathbf{L}$ as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}.$$

Let $\boldsymbol{\phi}_i$ the i-th eigenvector of $\mathbf{L}$ (with $i = 1, \ldots, n$) and $\boldsymbol{\Lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)^T$ the eigenvalues of the Laplacian. Finally, let m be a set of time values $\{t_1, t_2, \ldots, t_m\}$, we define the HKS feature vector $\mathbf{f} = (f_1, f_2, \ldots, f_n)^T$ for each time t_j as

$$\mathbf{f}_j = \sum_{k=1}^{n} \exp(-t_j \lambda_k) \boldsymbol{\phi}_k^2,$$

where the square of k-th eigenvector is meant just as a punctual operation over the components $\mathbf{x}^2 = (x_1^2, x_2^2, \ldots, x_n^2)^T$. Once computed, the feature vectors are collected on a $n \times m$ matrix $\mathbf{F}$ as columns

$$\mathbf{F} = (\mathbf{f}_1\ \mathbf{f}_2\ \cdots\ \mathbf{f}_m).$$

Given two graphs G^i and G^j (with the same number of nodes n), our HKS kernel is defined as the sum of the dot product between the respective feature matrices $k = \langle \mathbf{F}_i, \mathbf{F}_j \rangle = (k_1, k_2, \ldots, k_n)^T$. Hence, the kernel matrix is defined as

$$K_v(G^i, G^j) = \sum_{w=1}^{n} k_w.$$

On the other hand, the edge kernel K_e was chosen to be a discrete enforcement of the topological structure:

$$K_e\big((u,v),(a,b)\big) = \begin{cases} 1 & \text{if } \big((u,w) \in E_i \wedge (a,b) \in E_j\big) \vee \big((u,w) \notin E_i \wedge (a,b) \notin E_j\big) \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

The positive semidefiniteness of the proposed kernel can be proved through the closure properties of positive definite functions. The closure under sum states that, given a non-empty set X and two positive semidefinite symmetric kernels $K_A, K_B : X \times X \to \mathbb{R}$, it holds

$$K = K_A + K_B : X \times X \to \mathbb{R}. \tag{4.11}$$

Then, K is a positive semidefinite symmetric kernel. In other words, in order to construct a new positive semidefinite kernel as the sum of existing ones (K_v and K_e in our instance), first the kernels need to be positive semidefinite. Second, they all must be defined in the same space. The kernels employed in (4.9) are positive semidefinite by hypothesis. Furthermore, since the projection on the transitive alignment space introduces a reference canonical order (and such canonical ordering is guaranteed by the transitivity constraints, see section 4.2), the space of the kernels is the same. In fact, the kernels defined as the sum of all K_vs (K_A) and the sum of all K_es (K_B) are clearly positive semidefinite since all K_vs and all K_es belong to the same respective spaces. Hence, the kernel defined in (4.9) is positive semidefinite. Note that without the transitive alignment and its induced canonical ordering, the assumption that all K_vs and K_es belong to the same respective spaces would be wrong.

4.4 Experimental Evaluation

We evaluate the performance of the proposed method in terms of classification accuracy and we compare it with a number of well-known kernels, namely the Weisfeiler-Lehman kernel [151] (where the number of iterations parameter was set to $h = 3$ and we used the degree of each node as the node attribute), the graphlet kernel [153], the shortest-path kernel [18], the random walk kernel [79] and an experimental kernel based on the Heat Kernel Signature [162] method. In particular, we employ the Heat Kernel Signature to compute the feature descriptors with respect to $k = 100$ time parameters t uniformly distributed within the range $[1, 10]$ and we build the kernel as described in section 4.3.

Furthermore, we compare the performance of the proposed method with respect to the state-of-the-art of graph matching methods, namely the Spectral Matching (SM) [89] and Reweighted Random Walks Matching (RRWM) [30]. In order to do so, we address the classification task using several popular datasets with and without the permutations computed by the graph matching methods.

Given a pair of graphs (G^p, G^q) with the same number of nodes n, we compute the $n^2 \times n^2$ affinity matrix $\mathbf{M}_{pq} = (m_{ia,jb})$ on the respective edge weights (a^p_{ij}, a^q_{ab}) as

$$m_{ia,jb} = \exp\left(-\frac{(a^p_{ij} - a^q_{ab})^2}{\sigma^2}\right),$$

where σ^2 is a scale factor which is experimentally set to 0.15. This affinity matrix is employed as the input of one of the graph matching technique (GM) introduced above

(SM and RRWM), obtaining the $n \times n$ weight matrix $\mathbf{W}_{pq} = \text{GM}(\mathbf{M}_{pq})$. Note that the number of nodes of the graphs G^p and G^q are not required to be same, since if they are different, we will just add some disconnected dummy nodes in order to make the number of the nodes equal, as explained in section 4.2. Finally, we use the real matrix $\mathbf{W}_{pq}$ as the input for the Hungarian algorithm, which is a well-known method that performs a combinatorial optimization finding a maximum score matching in a weighted bipartite graph. This results in a discretised version of the weight matrix, which is, in practice, a permutation matrix. Hence, we define the permutation matrix $\mathbf{P}_{pq}$ as

$$\mathbf{P}_{pq} = \text{Hungarian}(\mathbf{W}_{pq}).$$

We run our experiments on the following datasets:

MUTAG dataset [40] was constructed based on data from review of literatures about mutagenicities in *Salmonella Typhimurium* based on 200 aromatic and heteroaromatic nitro compounds. As a result, 188 congeners were extracted together with their structure-activity relationship (SAR) data.

PPI dataset, which consists of protein-protein interaction (PPIs) networks related to Histidine Kinase [75] (40 PPIs from *Acidovorax avenae* and 46 PPIs from *Acidobacteria*).

PTC (The Predictive Toxicology Challenge) dataset, which records the carcinogenicity of several hundred chemical compounds for male rats (MR), female rats (FR), male mice (MM) and female mice (FM) [91] (here we use the 344 graphs in the MR class).

COIL dataset, which consists of 5 objects from [117], each with 72 views obtained from equally spaced viewing directions, where for each view a graph was built by triangulating the extracted Harris corner points.

Reeb dataset, which consists of a set of adjacency matrices associated to the computation of reeb graphs of 3D shapes [15].

ENZYMES dataset [146] is based on graphs representing protein tertiary structures consisting of 600 enzymes from the BRENDA enzyme database, which are correctly assigned to one of the 6 EC top-level classes.

SHOCK dataset consists of graphs from the Shock 2D shape database. Each graph of the 150 graphs divided into 10 classes is a skeletal-based representation of the differential structure of the boundary of a 2D shape.

For efficiency purposes, the experiments do not involve the whole datasets (see section 4.5). In particular, we select a certain number of classes and a certain number of graphs for each class. The selection of these subsets is performed randomly on the original datasets. Table 4.1 shows the number of classes and the number of graphs of each dataset that has been used to compute the results. In order to get a homogeneous number of nodes within the graphs of the same dataset, we add to each graph $n_{MAX} - n_i$ dummy nodes (*i.e.* not connected nodes), where n_{MAX} is the maximum number of nodes among the graphs of a certain dataset, while n_i is the number of nodes of the i-th graph.

We used a binary C-SVM to test the efficacy of the kernels. We performed 10-fold cross validation, where for each sample we independently tune the value of C, the SVM

Table 4.1: *Details of the pruned datasets employed in classification experiments.*

Dataset Name	Classes	Graphs per class	Total Graphs	Graph Nodes
MUTAG	2	≈ 94	188	28
PPI	2	15	30	149
PTC	2	30	60	70
COIL	2	20	40	112
Reeb	3	20	60	86
ENZYMES	3	20	60	26
SHOCK	10	15	150	33

regularizer constant, by considering the training data from that sample. The process was averaged over 100 random partitions of the data, and the results are reported in terms of average accuracy $\pm$ standard error. In particular, at each 10-fold cross validation iteration, the dataset is randomly permuted and subdivided in 10 folds. Every fold is used as a cross-validation fold, while the remaining are use to train the SVM. The process is repeated 100 times. Finally, we define the standard error as

$$\hat{\sigma}_X = \sqrt{n} \cdot \sqrt{\frac{\sum_{i=1}^{n}(x-\bar{x})^2}{n}} = \sqrt{\sum_{i=1}^{n}(x-\bar{x})^2},$$

where $\bar{x}$ is the mean accuracy obtained in a cross-validation iteration with n samples $X = \{x_1, x_2, \ldots, x_n\}$. Our experimental setup and evaluation strategy are inspired to the protocol [52] from which we imported the proper setting of parameters for graph kernels and classifier.

Table 4.2 shows the average classification accuracy ($\pm$ standard error) of the different kernels on the selected datasets. The first part of the table shows the accuracy computed using the datasets after the pruning operation mentioned before. These measurements are achieved just by pairwise kernel with respect to whole set of graphs. We can observe that our HKS graph kernel is quite competitive in the majority of the datasets, even without using information from direct correspondences between each pairs of graphs. The second part of the table (after the double line) shows the classification accuracy achieved after the application of the permutations yielded by the compared graph matching methods. More precisely, given $\mathbf{P}_{ij}$ the permutation matrix which defines the correspondences of the graph i with respect to graph j, we compute the value of the kernel between the permuted graph i and the graph j. In particular, HKS-SM shows the classification accuracy obtained permuting the graphs using the Spectral Matching results, while HKS-TSM shows the results obtained using the proposed method which has been initialized using Spectral Matching. The results show that the proposed method is competitive and outperform the other graph matching algorithms in almost all the datasets. COIL and PTC datasets are an exception, since HKS-RRWM performs slightly better with respect to our proposal.

Table 4.2: *Classification accuracy (± standard error) on unattributed graph datasets. Respectively, HKS is the Heat Kernel Signature [162], WL is the Weisfeiler-Lehman kernel [151], GR denotes the graphlet kernel computed using all graphlets of size 3 [153], SP is the shortest-path kernel [18], and RW is the random walk kernel [79]. The second part of the table collects the accuracy of HKS kernel employing the permutations from Spectral Matching (SM) [89] and Reweighted Random Walks Matching (RRWM) [30] with respect to the transitive versions produced by our method (denoted by the prefix T). For each kernel and dataset, the best performing kernel is highlighted in italic, while the bold highlights the maximum just considering data i n the second part of the table for each pair of graph matchings (non transitive w.r.t. transitive).*

Kernel	MUTAG	PPI	PTC	COIL	Reeb	ENZYMES	SHOCK
HKS	80.5 ± 0.2	63.6 ± 0.7	50.2 ± 0.5	*87.8 ± 0.8*	46.6 ± 0.6	56.9 ± 0.6	*46.8 ± 0.3*
WL	78.3 ± 0.2	*70.4 ± 0.8*	*67.1 ± 0.6*	70.6 ± 0.7	*68.7 ± 0.4*	55.4 ± 0.6	35.0 ± 0.2
SP	*83.3 ± 0.2*	58.5 ± 0.7	50.5 ± 0.6	86.7 ± 0.6	68.1 ± 0.4	52.2 ± 0.5	39.0 ± 0.3
RW	80.1 ± 0.2	48.5 ± 0.8	41.6 ± 0.6	65.2 ± 0.7	49.8 ± 0.6	13.6 ± 0.3	1.7 ± 0.1
GR	81.5 ± 0.2	30.3 ± 0.5	51.6 ± 0.6	87.1 ± 0.5	22.7 ± 0.6	47.0 ± 0.6	26.1 ± 0.3
HKS-SM	69.0 ± 0.3	60.9 ± 0.8	49.4 ± 0.6	84.8 ± 1.0	45.7 ± 0.6	49.1 ± 0.6	39.4 ± 0.4
HKS-TSM	**80.7 ± 0.2**	**64.2 ± 0.8**	**50.1 ± 0.6**	**87.0 ± 0.9**	**46.2 ± 0.5**	***57.2 ± 0.7***	**46.7 ± 0.3**
HKS-RRWM	79.8 ± 0.2	60.4 ± 0.9	**52.1 ± 0.5**	**87.3 ± 0.9**	44.5 ± 0.6	44.9 ± 0.6	25.7 ± 0.2
HKS-TRRWM	**80.5 ± 0.2**	**64.3 ± 0.8**	50.9 ± 0.5	86.1 ± 0.9	**44.8 ± 0.6**	**45.5 ± 0.6**	**46.4 ± 0.3**

Note that the first part of the table should be treated by the reader just as a reference of the accuracies that the state-of-the-art kernel methods achieve. Indeed, these kernels work independently from the alignment of the graphs to be classified.

On the whole, we highlight the evident aspect that best registered performances are mainly related to kernels which do not exploit of structural information, *i.e.*, the measurements organized in the first part of Table 4.2. There would have been expected to observe an opposite behaviour actually, but in particular with classification tasks the discrimination by topologies cannot reflect properly the ground-truth as well, which is just a handmade-assigned attribute (not a vertex-permutation). Therefore, supporting this deeper observation such divergence has less relevance in the overall evaluation of our work. After that, the main goal of our experimental results was the comparison between the proposed aligned methods with respect to the compared ones, namely Spectral Matching and Reweighted Random Walks Matching. In particular, we wanted to proof that our transitive approaches outperformed the performances by the current state of the art in graph matching methods.

Nevertheless, reasoning just about the plain gain in classification accuracy derived by our method it may lead to erroneous conclusions since the observed measures are rough averages actually. With the purpose to support more faithfully the discussion of our results, we performed the well-known Student's t-test to determinate if the difference between two means is statistically significant, *i.e.*, if we can reject the null hypothesis that the means related to the unknown probability distributions which yield to the measured performances (*i.e.*, those in second part of the Table 4.2) are equal for the original graph matching technique μ_{GM} and related transitive version through our method μ_{TGM}. More-

Table 4.3: *Details of t-tests performed to compare the classification accuracy means from the second part of the Table 4.2 with respect to the transitive kernels on (a) Spectral Matching and (b) Reweighted Random Walks Matching for each datasets. The null hypothesis H_0 states the accuracy means come from independent random samples with normal distributions which have equal means, while the alternative hypothesis is the means from our transitive kernels are greater than normal performances (left-tailed test). We reject H_0 at the significance level $\alpha = 5\%$ and assuming that the unknown variances from the two distribution are equal. Since the results are means from 100 trials, the degrees of freedom are 198.*

Dataset	p-value	Rejected H_0
MUTAG	0.000	**Yes**
PPI	0.002	**Yes**
PTC	0.184	No
COIL	0.059	No
Reeb	0.284	No
ENZYMES	0.000	**Yes**
SHOCK	0.000	**Yes**

(a) Kernel means for HKS-SM *versus* HKS-TSM

Dataset	p-value	Rejected H_0
MUTAG	0.013	**Yes**
PPI	0.002	**Yes**
PTC	0.943	No
COIL	0.829	No
Reeb	0.367	No
ENZYMES	0.205	No
SHOCK	0.000	**Yes**

(b) Kernel means for HKS-RRWM *versus* HKS-TRRWM

over, once we have established the means are different we impose the desirable alternative hypothesis that the mean of our method is greater than the non transitive multi-graph matching $\mu_{TGM} > \mu_{GM}$ as well. In Table 4.3 we collected the final decisions of all the possibles t-tests for each datasets dividing the comparisons related to the kernels by (a) Spectral Matching and (b) Reweighted Random Walks Matching. The overall conclusions are quite in agreement with respect to our initial comments. Sure enough, the cases where we can reject the null hypothesises are just featured by a significant divergence in the average classification accuracy. Furthermore, we can observe that PTC, COIL, and Reeb are quite difficult datasets for both our two methods (a) and (b). In general, the failures of some tests do not reveal necessarily that our method is worse with respect to the original techniques, but just the accuracies are statistically similar. Indeed, we encountered such trend when the classification results are aligned (and especially not under) the baseline performances.

4.5 Dimensionality Analysis

In this section we present a detailed analysis to investigate further about the dynamics and dimensionality capabilities of our method. Considering a dataset of N graphs and n nodes, the computation of the kernel between two instances G_i and G_j can be divided in three different steps, whose time and space complexity are formalised as follows.

1. **Initial Permutations**. According some external graph matching solver with con-

stant cost $O(\kappa_{(n^2\times n^2)})$ to generate the pairwise solution (whose order is justified by the cost to operate on the $n^2 \times n^2$ affinity matrix), the whole computation would require $O(N^2\kappa_{(n^2\times n^2)})$ or $O\big(\binom{N}{2}\kappa_{(n^2\times n^2)}\big)$ without considering the redundant configurations.[1] The maximum magnitude in space required for a couple of graphs is $O(n^4)$, since we need to consider the huge affinity edge matrix, hence whole process requires at most $O(N^2n^4)$ or $O\big(\binom{N}{2}n^4\big)$.

2. **Transitive Alignments**. The computation of a relaxed alignment $\mathbf{X}_i$ at time t requires a cost of $O(N\kappa_{(n\times n)})$, hence the whole process has $O(tN^2\kappa_{1(n\times n)})$. Considering the discretization cost of an alignment as $O(\kappa_{2(n\times n)})$, the whole time complexity is $O(tN^2\kappa_{1(n\times n)} + N\kappa_{2(n\times n)}) = O(tN^2\kappa_{(n\times n)})$. This elaboration requires mainly all the initial permutation matrices and the alignments, therefore the space is $O(N^2n^2 + Nn^2) = O(N^2n^2)$.

3. **Graph Kernel**. For each graph G_i is required three steps, the constant cost to compute the Laplacian $O(\kappa_{1(n\times n)})$, the SVD decomposition $O(\kappa_{2(n\times n)})$, the feature matrix $\mathbf{F}_i$ of m HKS feature vectors $O(mn\kappa_{3(n)})$, hence we can denote such overall cost as $O(\kappa_{1(n\times n)} + \kappa_{2(n\times n)} + mn\kappa_{3(n)}) = O(\kappa_{123(n\times n)})$ with the general assumption that holds $m \leq n$. Finally, considering the time for the dot product of two feature matrices as $O(\kappa_{4(n\times m)})$ and the summation of the kernel values n, the final cost to compute the whole kernel value spends $O(2\kappa_{123(n\times n)} + \kappa_{4(n\times m)} + n)$, which can be expressed to $O(\kappa_{(n\times n)})$. In terms of space complexity this task for one graph requires the structures to store the SVD decompositions $O(n^2 + n)$ and the HKS features $O(nm)$, hence for the assumption on m the memory used is in the order of $O(n^2+n+nm) = O(n^2)$. Finally, there is required to store the dot product $O(n)$ of two feature matrices to produce the final kernel value $O(1)$, therefore the overall space is $O(2n^2 + n + 1) = O(2n^2)$.

We need to highlight the operative dependencies of these fundamental tasks to understand better the implications in our method. The computation of the initial solution at first step (1) has to be considered always an independent problem. Therefore, we are forced to import the cost of such external graph matching technique. The second step (2) represents the fundamental process in our algorithm, whose cost is related to dynamics of the power iteration. There is not a reliable way to predict *a priori* the time t, since this behaviour depends strictly by the input data, which are all the permutations derived from the external graph matching solver. The impact of our Graph Kernel (3) could be retained marginal from an internal point of view, since it is just a static procedure with a constant cost. Considering a complete execution to compute whole kernel matrix, the overall time complexity is $O\big(N^2\kappa_{1(n^2\times n^2)} + tN^2\kappa_{2(n\times n)} + N^2\kappa_{3(n\times n)}\big)$, but if we consider that the number of iterations is limited always to a maximum threshold, the final time could be roughly

[1] We introduce the shortcut notation $\kappa_{(n_1\times n_2)}$ just to express in very general terms some constant computational time that is spent in structures of magnitude $n_1 \times n_2$, as for example matricial product, norm, black-box algorithms and so on; denoting with k the scalar cost of an operation, another possibly interpretation could be $O(kn_1n_2)$.

approximated to $O\big(N^2\kappa_1(n^2\times n^2)\big)$. In other terms, the real weight of the whole process is mainly dominated by the complexity of the external graph matching solver, which involves intuitively the final space complexity $O(N^2n^4 + N^2n^2 + 2n^2N^2) = O(N^2n^4)$ as well. Moreover, we observe that the order of nodes n has more impact with respect to the number of graphs N in the dataset.

From the consideration as above we reveal two fundamental weak points in our method. First, we cannot control completely the cost of the computation, which is dominated by the external pairwise graph matching technique. After that, we designed our work just as a rectification process [123] to introduce cycle consistency to an initial independent solution. Second, as concern our graph kernel we can realize that even to compute one measure between two graphs only, there is involved always whole weight of the problem to solve the related pair of transitive alignments. This aspect is very different in comparison to other graph kernels that do not require additional work beyond the couple of graphs.

For these main reasons the employment of our method on large-scale data becomes very critical, both in terms of computational time and data storage. Nevertheless, since our approach discriminates by factorizing permutations solving a common set of transitive alignments (differently by means of structures as other kernel methods) we guess that, with the purpose to maintain the global consistency on big data, the accuracy should be always led to saturate (or downgrade) once reached a certain dimensionality.

4.6 Conclusion

In this chapter we investigated the use of multi-graph matching techniques in the context of graph kernels. By forcing the correspondences between the structures under study to satisfy transitivity, we obtain an alignment kernel that, not only is positive definite, but also makes use of more reliable locational information obtained through the enforcement of global consistency constraints. We proposed a general framework for the projection of (relaxed) correspondences onto the space of transitive correspondences, thus transforming any given matching algorithm to a transitive multi-graph matching approach. The resulting transitive correspondences were used to provide an alignment-based kernel that was able to both maintain locational information and guarantee positive-definiteness. Experimental evaluation shows that the projection onto the transitive space almost invariably increases the classification performance of the alignment kernel, often taking it to a performance level that is at least statistically equivalent to the best performing well-tuned graph kernels present in the literature.

5

Dense Multi-view Homography Estimation and Plane Segmentation

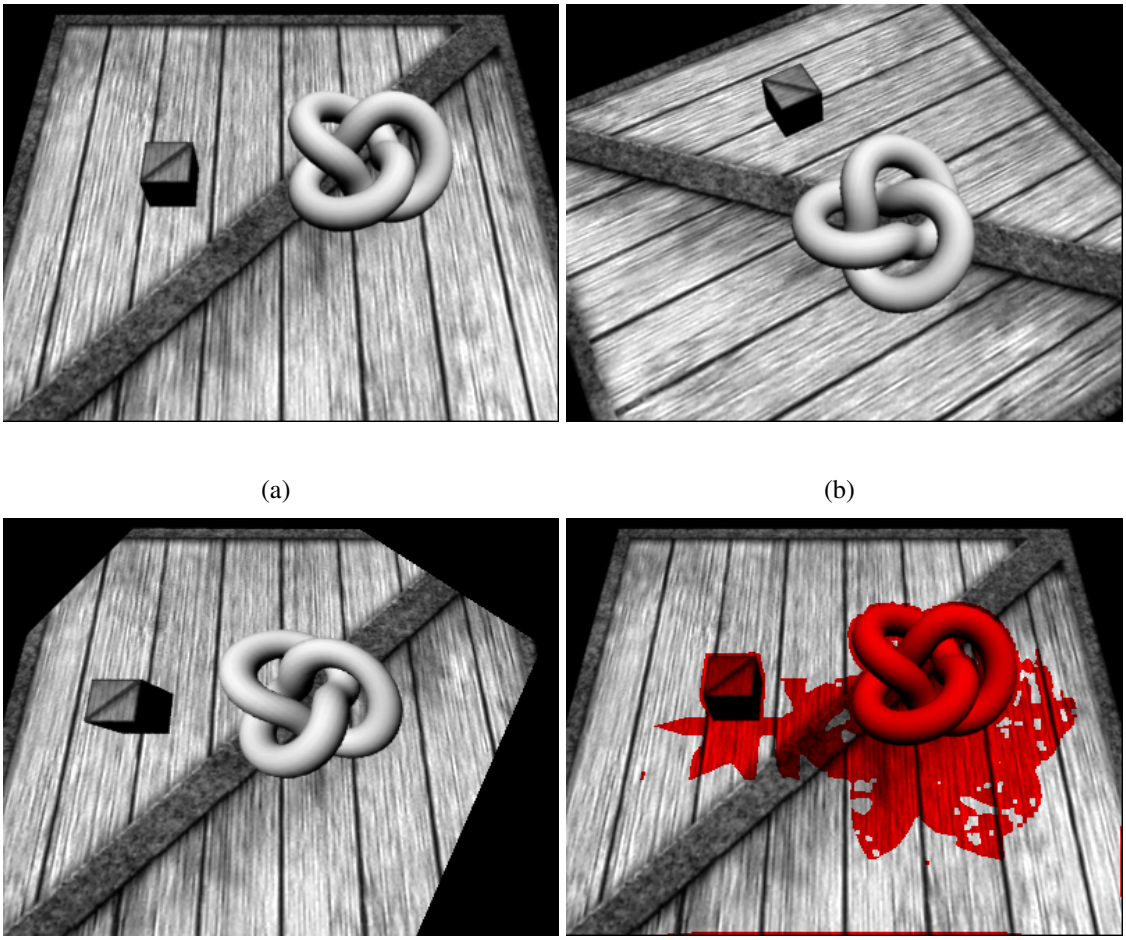
When a planar structure is observed from multiple views, the projections of its corresponding 3D points on each image are related by a homography. Its estimation is a key step in many computer vision tasks where either the rigid motion between views or a per-pixel image correspondence is sought. The vast majority of multi-view homography estimation techniques relies on matching a sparse set of point-to-point correspondences to establish a connected graph in the camera network. This track creation step is critical to ensure that the following bundle adjustment can estimate a globally optimal alignment in which the error is diffused coherently on each pairwise homography. On the other hand, erroneous or short tracks often cause misalignments among the views.

We propose an optimization technique to simultaneously recover a transitively consistent network of planar homographies between multiple views together with a segmentation of the pixels comprising the observed plane. Our method acts on a per-pixel basis to avoid a preliminary multi-view sparse feature matching step. Similarly to bundle adjustment, the error is diffused so that each homography in the view graph is transitively consistent with the others. The effectiveness of the proposed approach is evaluated in synthetically generated scenes and real-world scenarios.

5.1 Introduction

Given the ubiquitous nature of planar surfaces in urban and man-made environments, the estimation of homographies giving an image-to-image mapping of plane projections is at the core of many computer vision techniques. These are for instance: augmented reality [130], multi-camera [178] or camera-projector [125] calibration, metric rectification [92], and ground-plane recognition for object detection and tracking [7]. When the planar surface is at infinity (*i.e.*, views are related by a pure rotation around the optical centre), such mapping can be used in panoramic stitching [24] to create wide-angle images with normal lenses.

The majority of homography recovery approaches resolve the estimation by spatially matching a set of geometric primitives (points, lines or conics) between the images [110,



(a) (b)

(c) (d)

Figure 5.1: *(a,b) Synthetically generated views of a scene composed by a plane and two objects. (c) Image (b) warped with the plane homography computed by our algorithm. (d) The recovered plane segmentation (non-plane points are marked in red).*

155]. Approaches that operate globally on the image have also been proposed [86], but are effective only if the observed planar structure fills the entire image with no additional clutter. Our method falls in this category as it operates pixel-wise among the images, but we overcome such limitation by estimating a mask that filters the pixels outside the plane during the estimation (see Figure 5.1).

More recently, great effort has been spent in the study of the estimation of a set of interdependent homographic transformations. These can be generated by a multi-view observation of the same planar scene or when the scene is composed by multiple planar objects. In the former case, care has to be taken to enforce that each pairwise transformation is transitively consistent with the other. This ensures that the estimation error is evenly distributed across the views and reflects the fact that the scene is not changing over

time. This problem is usually solved with a bundle adjustment approach that globally minimizes the feature matching error with respect to a set of homographies transforming each image into a common (unknown) reference frame [176]. In the latter, when multiple planes are simultaneously estimated, additional constraints have to be imposed to ensure that all the homographies are mutually consistent with the rigid motions performed by the camera [194]. As recently claimed in [166], this is still an open problem that will require to find novel ways to enforce consistency between the recovered homographies and the epipolar geometry involved.

In this chapter we propose a pixel-based approach to recover a transitively consistent set of pairwise homographies between multiple views of a single plane. Together with the homographies, we estimate a plane segmentation mask for each image to mark the plane pixels. The advantages are threefold. First, we do not rely on the identification of a sparse sequence of feature matches over multiple images (*i.e.*, tracks). This may represent a non-trivial step in scenes for which the visibility of a 3D scene points is limited to a small subset of the images, or where changes in view-direction alter the appearance of the feature points significantly. Second, we always operate on a pair-wise bases (for both homography estimation and plane segmentation) while enforcing multi-view consistency among the iterations. This increases the performances w.r.t. bundle adjustment approaches since each pairwise step can be efficiently parallelized. Finally, we obtain more accurate results than sparse methods since every image pixel contributes to the estimation.

5.2 Preliminaries

We assume to have a set of $I_1, \dots I_n$ images of a scene in which at least one plane Π is visible. For each image pair (I_i, I_j), there exists a 3×3 homography matrix $\mathbf{H}_{ij} = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix}$ mapping the projection of the 3D points of Π from I_i to I_j through the function:

$$H_{ij}(x, y) = \frac{[\mathbf{h}_1 \ \mathbf{h}_2]^T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}{\mathbf{h}_3^T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}. \tag{5.1}$$

If the camera calibrations $\mathbf{K}_i, \mathbf{K}_j$ and Π are known, $\mathbf{H}_{ij}$ can be expressed as $\mathbf{H}_{ij} = \mathbf{K}_j(\mathbf{R}_{ij} - \frac{\mathbf{t}\mathbf{n}^T}{d})\mathbf{K}_i^{-1}$ where $(\mathbf{R}_{ij}, \mathbf{t})$ is the relative motion between the cameras, $\mathbf{n}$ is the plane normal and d is the plane distance w.r.t. the i-th view.

The homographic mapping (5.1) is subject to 8 degrees of freedom since $\mathbf{H}_{ij}$ describes the same transformations up to any non-zero scale factor. To resolve the unknown scaling,

in our method we restrict each homography to be part of the special linear group $\mathbb{SL}_3$ with respect to standard matrix multiplication by normalizing $\mathbf{H}_{ij}$ so that $\det(\mathbf{H}_{ij}) = 1$.

When we consider a camera network topology in which the view graph contains cliques with cardinality $n \geq 2$, the set of relative pairwise homographies are transitively consistent if

$$\mathbf{H}_{ij}\mathbf{H}_{jk} = \mathbf{H}_{ik} \qquad \forall i, j, k = 1, \ldots, n. \tag{5.2}$$

This implies that the transformation from i to j followed by the transformation from j to k leads to the same result that the direct transformation from i to k. This is a crucial condition that raises naturally by the fact that all the planar mappings are related to the common scene reference frame.

5.3 Combined Homography and Plane Segmentation Recovery

Our goal is to find the set of homographies $\bar{H} = \{\mathbf{H}_{ij}\}$ $\forall i, j = 1 \ldots n$ with respect to the observed plane Π. Since the scene may also contain other elements than the plane, for each image I_i we introduce a plane mask function $\chi_i(x, y) : \mathbb{R}^2 \rightarrow \{0, 1\}$ that classifies each pixel (x, y) as being part or not of a plane ($\chi_i(x, y) = 1$ in the former case, and $\chi_i(x, y) = 0$ in the latter).

We pose the problem as an energy minimization of the functional:

$$E = \sum_i \sum_j \int_x \int_y \hat{\chi}_i(x, y|\bar{H}) \hat{\chi}_j\big(H_{ij}(x, y)|\bar{H}\big) r_{ij}^2(x, y)\, dxdy \tag{5.3}$$

with respect to $\mathbf{H}_{1\ldots n,1\ldots n}$, where

$$r_{ij}^2(x, y) = \Big(I_i(x, y) - I_j\big(\mathbf{H}_{ij}(x, y)\big)\Big)^2$$

is the squared difference between I_i and I_j warped through $\mathbf{H}_{ij}$, and $\hat{\chi}_i(x, y|\bar{H})$ is an estimator of χ_i conditioned by $\bar{H}$. Each $\mathbf{H}_{ij}$ is also constrained in $\mathbb{SL}_3$ and to be transitively consistent with the others as stated in (5.2).

To optimize E we assume to get an initial estimate of $\bar{H}^{(0)}$ at time $t = 0$ and iteratively compute $\bar{H}^{(t+1)}$ approximating $\hat{\chi}_i(x, y|\bar{H})$ in (5.3) with $\hat{\chi}_i(x, y|\bar{H}^{(t)})$.

In practice, we iteratively minimize the functional by alternating the following steps:

1. Optimize each pairwise homography assuming the plane masks estimators to be known;
2. Synchronize the transformations so that (5.2) is satisfied;
3. Recompute the plane masks estimators conditioned to the currently estimated homographies.

We iterate until the absolute difference between $\bar{H}^{(t+1)}$ and $\bar{H}^{(t)}$ is less than a threshold ϵ. In the following sections we describe each step in detail.

5.3.1 Homography Optimization

In the homography optimization step we perform a projected gradient descent over the functional (5.3) so that the special linear group constraint is preserved. At each step, we compute the gradient of E with respect to each component $h^{ij}_{u=1\dots3,v=1\dots3}$ of $\mathbf{H}_{ij}$ before projecting it into the tangent plane of $\mathbb{SL}_3$.

In the specific, at each iteration we compute the gradient $\delta_{\mathbf{H}ij}E$ as:

$$\begin{aligned}\frac{dE}{dh^{ij}_{uv}} &= \int_x\int_y \hat{\chi}_i(x,y|\bar{H})r_{ij}(x,y) \\ &\quad \cdot\Big[r_{ij}(x,y)\nabla\hat{\chi}_j\big(H_{ij}(x,y)|\bar{H}\big) \\ &\quad -2\hat{\chi}_j\big(H_{ij}(x,y)|\bar{H}\big)\nabla I_j\big(H_{ij}(x,y)\big)\Big]^T\frac{dH(x,y)}{dh^{ij}_{uv}}\,dxdy,\end{aligned}$$

where

$$\frac{dH(x,y)}{dh^{ij}_{uv}} = \frac{1}{\lambda}\begin{bmatrix}1 & 0 & -\frac{1}{\lambda}\mathbf{h}_1^T\begin{bmatrix}x\\y\\1\end{bmatrix}\\ 0 & 1 & -\frac{1}{\lambda}\mathbf{h}_2^T\begin{bmatrix}x\\y\\1\end{bmatrix}\end{bmatrix}\otimes[x\ y\ 1]$$

with $\lambda = \mathbf{h}_3^T\begin{bmatrix}x\\y\\1\end{bmatrix}$ and the image gradients ∇I_j numerically approximated with central finite differences.

Then, we enforce the $\mathbb{SL}_3$ constraint $C = \det(\mathbf{H}_{ij}) - 1 = 0$ by projecting $\delta_{\mathbf{H}ij}E$ in its tangent space:

$$\delta^{\|}_{\mathbf{H}ij}E = \delta_{\mathbf{H}ij}E - \frac{\delta_{\mathbf{H}ij}C^T\delta_{\mathbf{H}ij}E}{\delta_{\mathbf{H}ij}C^T\delta_{\mathbf{H}ij}C}\delta_{\mathbf{H}ij}C.$$

After updating each $\mathbf{H}_{ij}$ with a step in the opposite direction of $\delta^{\|}_{\mathbf{H}ij}E$, the determinants of the homographies might drift away slightly from the unitary value. To fix that, after each step we renormalise each transformation dividing $\mathbf{H}_{ij}$ by $\sqrt[3]{det(\mathbf{H}_{ij})}$.

5.3.2 Transformation Synchronization

After the previous step there is no guarantee that the transformations are transitively consistent since every homography $\mathbf{H}_{ij}$ is independently updated. To enforce this consistency, we use the approach recently presented in [13] to find the optimal (in least squares sense) common reference frame $\mathbf{H}_{i*}$ so that, if we update all the transformations with respect to that frame

$$\mathbf{H}_{ij} = \mathbf{H}_{i*}\mathbf{H}_{j*}^{-1} \qquad \forall i, j = 1 \dots n,$$

we obtain a synchronized network of views. Note that to let all the $\mathbf{H}_{i*}$ exist it is crucial to fix the scaling issue of the homography matrices. So, it is mandatory to constrain all the transformations in $\mathbb{SL}_3$ before performing the synchronization step.

5.3.3 Plane Mask Recovery

We used a multi-classifier fusion approach [136] to implement the plane mask estimator $\hat{\chi}_i(x, y|\bar{H}^{(t)})$. The idea is that, when we compare an image I_i with the warped I_j, we expect a good match for the pixels that are actual projections of the 3D plane points and an inconsistent behaviour with the others. Driven by this assumption, for each image pair (i, j) we classify plane/non-plane points with a binary segmentation of the aligned image difference:

$$\mathrm{Er}_{ij}(x, y) = e^{-\lambda\left(I_i(x,y) - I_j\left(\mathbf{H}_{ij}^{(t)}(x,y)\right)\right)^2}, \tag{5.4}$$

where λ is a parameter to tune the plane classifier sensitivity. The segmentation of each Er_{ij} is performed via Graph-Cut [71] to obtain a set of binary images which are described as $\Omega_{ij} : \mathbb{R}^2 \to \{0, 1\}$ that express the plane visibility of each pixel in I_i considering the photometric consistency induced by I_j. All the pairwise Ω_{ij} are then convolved with a $2D$ gaussian kernel G and accumulated as votes to estimate the visibility of each image pixel with respect to all the other images:

$$\hat{\chi}_i(x, y|\bar{H}^{(t)}) = \begin{cases} 0 & \text{if } \sum_i \sum_j G * \Omega_{ij} < v \\ 1 & \text{otherwise} \end{cases} \tag{5.5}$$

with a parameter v that thresholds the number of votes that each pixel must have to be considered part of the plane.

5.3.4 Implementation Details

Since our method works on a per-pixel basis, care must be taken in the remapping operation between two images. Indeed, the mapping $I_j\big(\mathbf{H}_{ij}(x, y)\big)$ is essentially a resampling of the image I_j at points $\mathbf{H}_{ij}(x, y)$ which would possibly fall in between the samples of the regular image lattice of I_j. The projective distortion induced by the required homographic transformation can be so strong that, in practice, a simple bilinear interpolation may not produce a good quality remapping. For this reason, we implemented a mipmapping strategy in which the original image is repeatedly downsized by a factor of two to create a stack of images with a decreasing level of detail: 1 for full size image, 2 for half-size, 3 for $\frac{1}{4}$ and so on. We define the level of detail associated to a pixel (x, y)

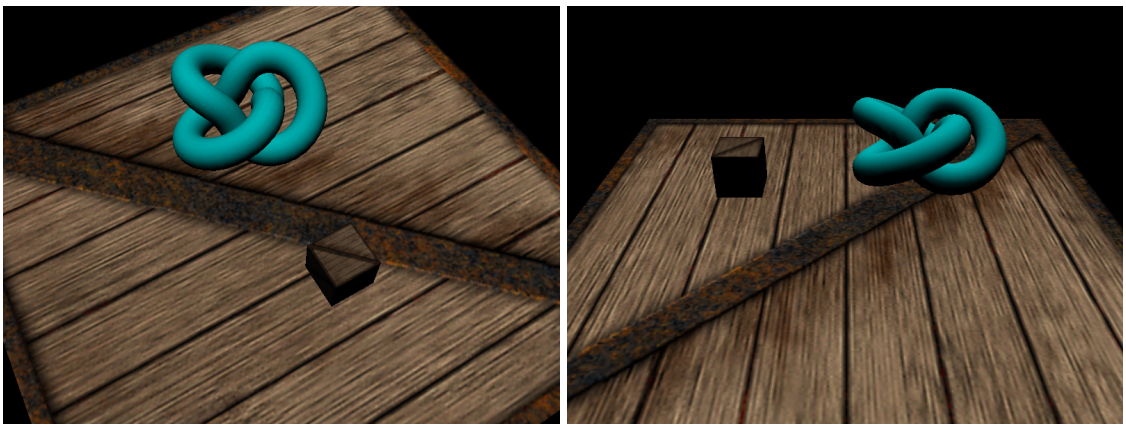


Figure 5.2: *Two example shots of our synthetically generated scene. The homographies are evaluated with respect to the big plane at the bottom of the two objects.*

warped through a homography H as:

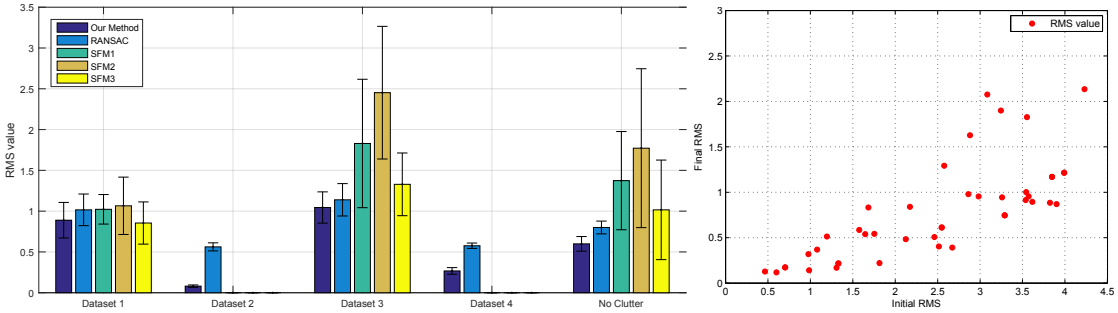
$$\mathrm{lod}_H(x, y) = \log\left(|\mathbf{J}_H(x, y)|\right),$$

where $|\cdot|$ is the determinant operator and $\mathbf{J}_H$ is the jacobian of $H(x, y)$. Finally, to resample an image, we use trilinear filtering to linearly interpolate the results of bilinear filtering performed at the two mip-map levels nearest to the level of detail of each pixel.

Finally, it is worth to mention that there is a trivial solution when $\chi_i(x, y)$ is 0 everywhere. We decided to avoid a regularization term for χ_i for two reasons. First, we assume that the initial configuration given by sparse matching is close enough to a local-optimum to give a reasonable result. Second, it is difficult to constrain the extent of the planar region for any generic scene without inevitably affecting each $\mathbf{H}_{ij}(x, y)$. With these premises, our method should be considered as an iterative refinement of an initial solution provided by sparse matching instead of a global energy minimization technique.

5.4 Experimental Validation

We tested the performance of our method quantitatively with a set of experiments on synthetically generated sample scenes. This allow us to get the ground-truth homographies between all the views together with the exact segmentation of all plane pixels. In our tests, we measure the accuracy of each pair-wise homography by generating a regular grid of 200×200 points and computing the squared error between each point transferred with the estimated homography and the ground truth. Then, all the squared errors are averaged among all the possible view pairs to compute the square root and obtain the final RMS error. Moreover, we presented qualitative tests with real world datasets to show the effective improvement in the optimized homography derived with the proposed method. Finally,



(a) (b)

Figure 5.3: *(a) Homography recovery comparison between the standard RANSAC approach, VisualSFM and our algorithm for all the synthetically generated datasets. (b) Homography accuracy recovered by our algorithm varying the initial pairwise homography errors.*

we combined quantitative and qualitative validations for both synthetic and real-world datasets by measuring the visual sharpness between original and transformed images.

We highlight that to guarantee the practicability of our process we need to operate with small motion datasets, due to the weak reference point set which arises working onto a single common plane spanned in multiple views [194]. Moreover, even if we can evaluate our method by applying different baseline data conditions, we are forced to skip the direct comparison with alternative schemes in literature for their incompatibility with respect to the special strategy treated here.

5.4.1 Synthetic Experiments

We generated a set of 5 different scenes comprising a single textured plane and two floating objects to simulate clutter and distracting elements that can be found on a real scenario. We rendered each dataset simulating a camera with resolution of 800×600 pixels, known intrinsics (800px focal length and principal point at the image centre) and a variable camera pose to compose a fully-connected network of 6 views. For each rendering, a per-pixel mask of the visible portion of the plane was generated as a plane segmentation mask ground truth. Dataset 1 was generated simulating a plane being at the base of a turntable so that the scene is imaged from 6 different points of view disposed evenly around the centre. This is the ideal condition for any Structure-from-Motion pipeline as it gives an optimal track extension and enough baseline for triangulation. Dataset 2 simulates a pure translational movement of a camera moving forward with a vector oriented slightly upward its optical axis. In Dataset 3 we tilted the plane around its x-axis (varying the plane pitch w.r.t. the camera) and, finally, in Dataset 4 we simulated a fixed camera with varying focal length ranging from 800 to 950 pixels. As a reference, we have also generated the "No Clutter" dataset with the same poses of Dataset 1, but with the plane as

the only visible object in the scene. In all the tests we set $\lambda = 0.003$, $v = \frac{n(n-1)}{2}$ and the smoothing kernel G with $\sigma = 0.5$.

We first compared the homography recovery accuracy of our method with a baseline standard approach (SIFT feature matching and RANSAC inlier selection) and the structure-from-motion pipeline VisualSFM [188]. In the latter, we tested both the uncalibrated and the calibrated camera case (named SFM1 and SFM2 respectively in the plots), to obtain a set of camera poses and sparse point cloud. To recover the homographies, we fitted a plane to the point cloud with care to not include any outlier deriving from the floating objects. Additionally, we tested the calibrated camera case with the same point matches used to initialize our method (SFM3).

In Figure 5.3(a) we observe that our method gives better results than the alternative sparse approaches in terms of final RMS error. SFM fails to converge to a valid solution for Dataset 2 and 3 since there is not enough baseline between the views to triangulate the plane points. As expected, the structure-from-motion approach gives its best results in Dataset 1 that are comparable with ours when a calibrated camera is given. However, our method do not use this assumption and hence we imagine a broader applicability. It is also worth to be noted that our method exhibits less variability (lower standard deviation in multiple repetitions of the experiment) with respect to the other methods.

To test the sensitivity of our method to the initial conditions we perturbed the ground truth homographies of Dataset 1 with zero-mean gaussian random noise with a standard deviation $\sigma = 10^{-3} \dots 10^{-2}$ and plotted the final homography RMS with respect to the initial $\big($Figure 5.3(b)$\big)$. In all the tests we observed a consistent decrease of the homography error which is particularly relevant when the initial RMS error is less than 2.5 px.

(a) (b)

Figure 5.4: *Example of the initial (a) and final (b) plane segmentation for a view of the Dataset 3. Note how the refined homographies allow a better classification of the plane points (in this case, the recall increases while exhibiting an almost constant precision).*

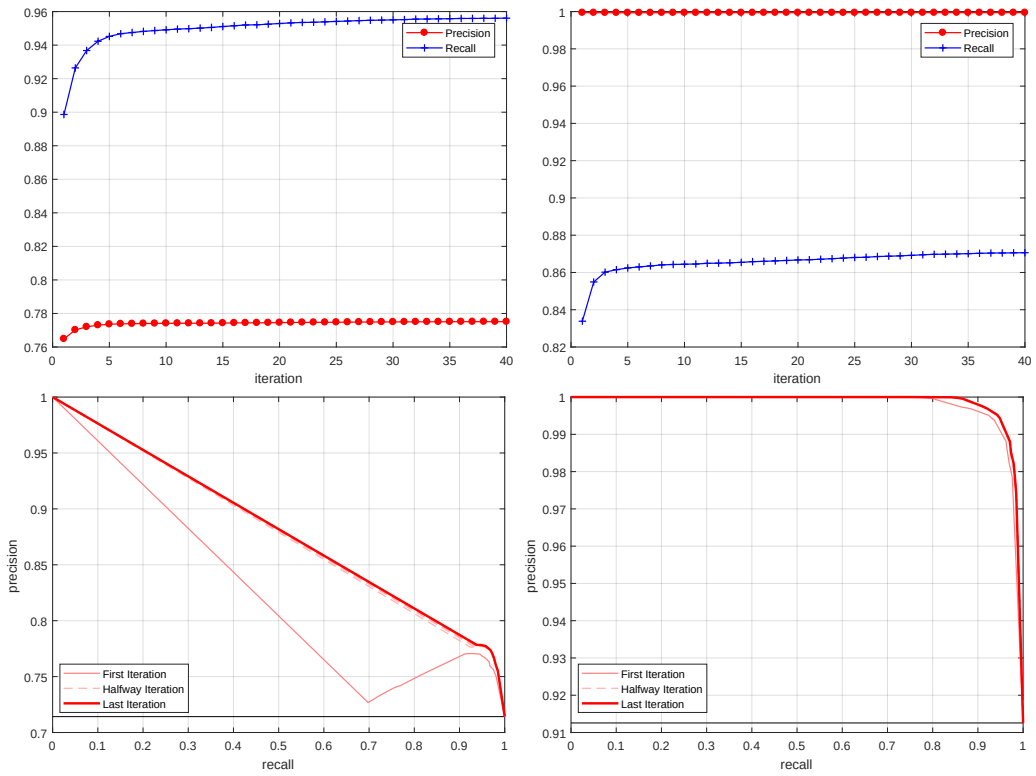


Figure 5.5: *Precision and recall performances of the plane point segmentation for Dataset 2 (left column) and Dataset 3 (right column) expressed as values (top row) and curves (bottom row) per iteration.*

This is expected since we operate on a pixel basis that requires the starting condition being not too distant from the optimal condition. Nevertheless, an initial error of about 3 pixels is easily achievable with any non sophisticated feature matching strategy.

To evaluate the quality plane segmentation we compared the obtained plane mask with the ground truth considering our method as a binary classifier (plane/non-plane) and computing the aggregated precision/recall among all the views during the optimizations. In Figure 5.5 we show an example of the precision and recall for Dataset 2 and 3 (respectively for the columns), which are plotted as values and curves (respectively for the rows) in 40 iterations. In both datasets the recall increases with the iterations since the refined homographies improve accuracy of the plane classifiers. Indeed, we can visually observe this behaviour on Figure 5.4(a) and Figure 5.4(b) looking how the classified non-plane points (in red) change between the first and last iteration. Finally, another aspect we realize from precision/recall experiments consists that the maximum accurateness of the plane classifiers are reached very fast actually, *i.e.*, more or less beyond the 5^{th} iteration. This is not a surprising behaviour, but it suggests that our optimization can require few time to be solved properly.

5.4.2 Qualitative Evaluation

We show some qualitative example of the results obtainable with our method on some real-world scenarios. In the first two rows of Figure 5.6 we show the results on a dataset composed by 5 different views of some object disposed on a flat surface whereas in the last two rows we examine a subset of the "castle-P19" scene of the dataset presented in [161]. In the first column we rendered one of the views of the network with the plane segmentation superimposed in red. In both the cases the object lying in front of the plane get correctly discarded as they are not coherent with the remapping induced by the homography. Additionally, the results after the optimization (2^{nd} and 4^{th} rows) exhibit less false negatives for the better alignment of the views. In the last 3 columns we show two views of the scenes warped by the estimated homographies and rendered in different colour channels to enhance the alignment errors. After the optimization we note a better alignment that is particularly visible in high contrast areas (in the last column we show a closeup of some regions).

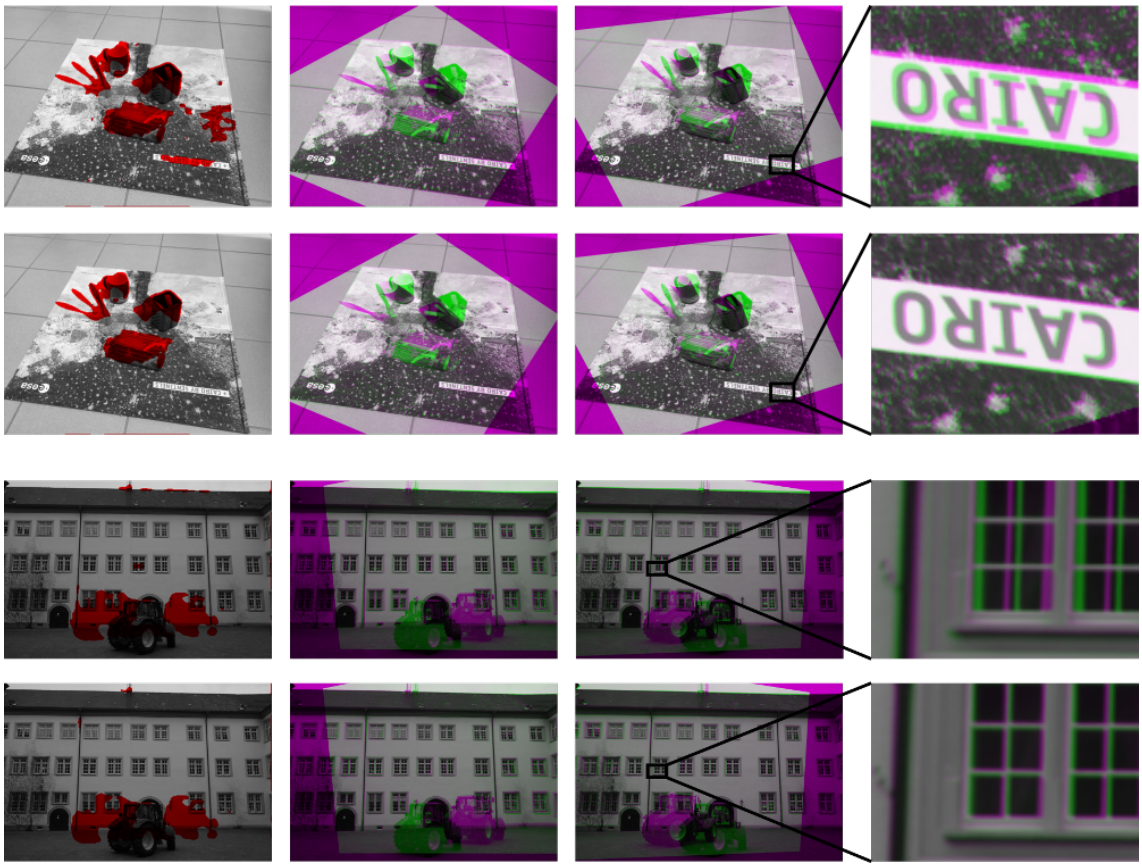


Figure 5.6: *Qualitative evaluation of the results obtained with the proposed method on two different real-world cases. First two rows: dataset composed by some objects lying on a flat surface. Last two rows: Image sequence from a subset of the "castle-P19" dataset.* 1^{st} *and* 3^{rd} *row show the starting point of the optimization obtained via sparse feature matching,* 2^{nd} *and* 4^{th} *row shows the results after the optimization.*

5.4.3 Visual Sharpness Evaluation

We complete the evaluation of the proposed method measuring concretely the visible accurateness of the optimized and synchronized homographies with respect to the related transformed images. In other terms, we aim to quantify the visual quality which a human eyes can infer by itself as we introduced in section 5.4.2. Our strategy consists to evaluate the overlapping of the image I_i and its reconstruction derived from another image I_j by applying the homography $I_j\big(\mathbf{H}_{ij}(x,y)\big) = I_{j\to i}(x,y)$. This alignment can be represented by the averaged image between the original and transformed instances as

$$\frac{1}{2}\Big(I_i(x,y) + I_{j\to i}(x,y)\Big)\hat{\chi}_i(x,y), \tag{5.6}$$

which is properly cleaned by the associated mask function $\hat{\chi}_i(x,y)$. We expect that as greater is the quality of homography $\mathbf{H}_{ij}$ as bigger will be the degree of focus for the mean image by the functional (5.6). In our experiment we used the well-known Brenner focus measure [65], which is based on summing the squares of the horizontal first-order derivative of the pixels. This measure is largely employed in photography to quantify the focus of objects which are contained in an image. We computed the Brenner focus measures for each datasets we used in our experiments and considering all possible configurations of homographies as well as the average focus values. The measurements related to the several synthetic datasets are reported in Tables 5.1, 5.2, 5.3, 5.4 and 5.5, while those of real-world scenarios in Tables 5.6 and 5.7. In general, we can observe that the sharpness values in the transformed images obtained with the optimized homographies increased considerably. Furthermore, also considering those cases where the general focus does not improve with respect to original images, the registered absolute divergences are quite low. Hence, considering the average values in focus for all the datasets, we can conclude that our method provides better quality than the initial homographies.

5.5 Conclusion

We presented a novel method for the simultaneous estimation of homographies and plane segmentation from multiple views of a planar scene. Our approach works on a pair-wise fashion in both the homography estimation and plane segmentation but imposes multi-view constraints by transformation synchronization and multi-classifier fusion. This guarantees that each homography in the view graph is transitively consistent with the others while still allowing an high-grade of parallelization of the optimization process which otherwise may be unsustainable in case of a fully-connected view graph. We posed the problem as an energy minimization acting densely (for each pixel) among the images. For these reasons, we are not affected by the limitations of multi-view sparse feature matching nor by the requirement of a calibrated camera. We have shown through synthetically generated data that we can effectively reduce the homography estimation error with respect to sparse estimation techniques. Even if the plane segmentation is focused to avoid false

positives, the obtained results are good enough to provide a good background-foreground segmentation for panoramic or mostly planar scenes.

From/To	Initial	Optimized	Improved
$2 \rightarrow 1$	81.65	77.54	No
$3 \rightarrow 1$	75.14	73.93	No
$4 \rightarrow 1$	70.57	69.87	No
$5 \rightarrow 1$	77.54	76.51	No
$6 \rightarrow 1$	69.66	69.19	No
$1 \rightarrow 2$	75.11	82.83	**Yes**
$3 \rightarrow 2$	86.18	91.57	**Yes**
$4 \rightarrow 2$	80.70	86.07	**Yes**
$5 \rightarrow 2$	81.55	87.29	**Yes**
$6 \rightarrow 2$	77.16	84.42	**Yes**
$1 \rightarrow 3$	116.24	117.06	**Yes**
$2 \rightarrow 3$	118.97	117.80	No
$4 \rightarrow 3$	108.37	108.85	**Yes**
$5 \rightarrow 3$	106.25	107.44	**Yes**
$6 \rightarrow 3$	117.90	119.06	**Yes**
$1 \rightarrow 4$	114.41	113.60	No
$2 \rightarrow 4$	115.73	119.70	**Yes**
$3 \rightarrow 4$	120.01	121.72	**Yes**
$5 \rightarrow 4$	109.55	109.74	**Yes**
$6 \rightarrow 4$	107.77	107.22	No
$1 \rightarrow 5$	116.03	111.89	No
$2 \rightarrow 5$	113.12	120.17	**Yes**
$3 \rightarrow 5$	121.29	122.36	**Yes**
$4 \rightarrow 5$	124.70	124.76	**Yes**
$6 \rightarrow 5$	113.97	109.98	No
$1 \rightarrow 6$	100.82	100.45	No
$2 \rightarrow 6$	104.90	110.11	**Yes**
$3 \rightarrow 6$	121.50	124.25	**Yes**
$4 \rightarrow 6$	118.70	121.76	**Yes**
$5 \rightarrow 6$	124.36	127.65	**Yes**
Mean values:	102.33	103.83	**Yes**

Table 5.1: *Brenner focus measures between initial and optimized homographies for "Synthetic Dataset 1".*

From/To	Initial	Optimized	Improved
$2 \rightarrow 1$	202.21	200.29	No
$3 \rightarrow 1$	185.61	184.45	No
$4 \rightarrow 1$	175.56	176.94	**Yes**
$5 \rightarrow 1$	166.42	167.14	**Yes**
$1 \rightarrow 2$	196.38	201.58	**Yes**
$3 \rightarrow 2$	184.32	187.75	**Yes**
$4 \rightarrow 2$	175.06	180.35	**Yes**
$5 \rightarrow 2$	167.40	170.23	**Yes**
$1 \rightarrow 3$	167.23	190.75	**Yes**
$2 \rightarrow 3$	171.69	190.99	**Yes**
$4 \rightarrow 3$	169.56	182.88	**Yes**
$5 \rightarrow 3$	166.54	170.80	**Yes**
$1 \rightarrow 4$	176.89	177.88	**Yes**
$2 \rightarrow 4$	182.28	179.29	No
$3 \rightarrow 4$	182.97	174.38	No
$5 \rightarrow 4$	182.65	179.17	No
$1 \rightarrow 5$	141.79	152.76	**Yes**
$2 \rightarrow 5$	148.72	154.36	**Yes**
$3 \rightarrow 5$	150.05	149.62	No
$4 \rightarrow 5$	144.34	153.31	**Yes**
Mean values:	171.88	176.25	**Yes**

Table 5.2: *Brenner focus measures between initial and optimized homographies for "Synthetic Dataset 2".*

From/To	Initial	Optimized	Improved
$2 \rightarrow 1$	150.40	162.32	**Yes**
$3 \rightarrow 1$	164.63	163.57	No
$4 \rightarrow 1$	158.60	166.84	**Yes**
$5 \rightarrow 1$	162.61	167.34	**Yes**
$6 \rightarrow 1$	164.26	167.17	**Yes**
$1 \rightarrow 2$	197.87	188.15	No
$3 \rightarrow 2$	200.33	191.56	No
$4 \rightarrow 2$	195.57	197.41	**Yes**
$5 \rightarrow 2$	197.32	193.77	No
$6 \rightarrow 2$	192.45	194.19	**Yes**
$1 \rightarrow 3$	170.49	173.21	**Yes**
$2 \rightarrow 3$	154.71	172.46	**Yes**
$4 \rightarrow 3$	163.88	178.48	**Yes**
$5 \rightarrow 3$	167.52	176.15	**Yes**
$6 \rightarrow 3$	169.07	175.70	**Yes**
$1 \rightarrow 4$	180.31	175.50	No
$2 \rightarrow 4$	170.60	174.82	**Yes**
$3 \rightarrow 4$	178.33	178.80	**Yes**
$5 \rightarrow 4$	188.21	182.97	No
$6 \rightarrow 4$	186.73	183.92	No
$1 \rightarrow 5$	162.45	159.73	No
$2 \rightarrow 5$	148.82	158.04	**Yes**
$3 \rightarrow 5$	158.20	159.92	**Yes**
$4 \rightarrow 5$	160.60	165.32	**Yes**
$6 \rightarrow 5$	180.61	178.25	No
$1 \rightarrow 6$	114.01	116.12	**Yes**
$2 \rightarrow 6$	106.33	111.53	**Yes**
$3 \rightarrow 6$	110.73	113.09	**Yes**
$4 \rightarrow 6$	109.60	115.65	**Yes**
$5 \rightarrow 6$	117.92	121.83	**Yes**
Mean values:	162.77	165.46	**Yes**

Table 5.3: *Brenner focus measures between initial and optimized homographies for "Synthetic Dataset 3".*

From/To	Initial	Optimized	Improved
$2 \rightarrow 1$	182.58	186.83	**Yes**
$3 \rightarrow 1$	176.41	182.44	**Yes**
$4 \rightarrow 1$	168.04	178.29	**Yes**
$5 \rightarrow 1$	158.67	166.70	**Yes**
$6 \rightarrow 1$	155.65	162.83	**Yes**
$1 \rightarrow 2$	192.53	188.92	No
$3 \rightarrow 2$	185.00	187.22	**Yes**
$4 \rightarrow 2$	175.62	182.06	**Yes**
$5 \rightarrow 2$	165.60	170.71	**Yes**
$6 \rightarrow 2$	162.05	166.87	**Yes**
$1 \rightarrow 3$	199.68	198.22	No
$2 \rightarrow 3$	197.01	198.04	**Yes**
$4 \rightarrow 3$	188.19	198.32	**Yes**
$5 \rightarrow 3$	181.65	185.80	**Yes**
$6 \rightarrow 3$	176.43	181.75	**Yes**
$1 \rightarrow 4$	203.17	194.51	No
$2 \rightarrow 4$	200.65	194.63	No
$3 \rightarrow 4$	196.55	196.17	No
$5 \rightarrow 4$	187.98	187.90	No
$6 \rightarrow 4$	185.91	184.86	No
$1 \rightarrow 5$	200.31	198.77	No
$2 \rightarrow 5$	201.04	199.26	No
$3 \rightarrow 5$	200.82	199.83	No
$4 \rightarrow 5$	193.96	200.52	**Yes**
$6 \rightarrow 5$	194.43	195.96	**Yes**
$1 \rightarrow 6$	194.94	191.73	No
$2 \rightarrow 6$	195.02	192.29	No
$3 \rightarrow 6$	192.28	192.81	**Yes**
$4 \rightarrow 6$	189.57	193.79	**Yes**
$5 \rightarrow 6$	189.35	189.70	**Yes**
Mean values:	186.37	188.26	**Yes**

Table 5.4: *Brenner focus measures between initial and optimized homographies for "Synthetic Dataset 4".*

From/To	Initial	Optimized	Improved
$2 \rightarrow 1$	169.74	168.91	No
$3 \rightarrow 1$	157.75	164.37	**Yes**
$4 \rightarrow 1$	160.58	164.55	**Yes**
$5 \rightarrow 1$	169.53	169.26	No
$6 \rightarrow 1$	178.79	177.69	No
$7 \rightarrow 1$	177.57	178.42	**Yes**
$1 \rightarrow 2$	177.20	181.94	**Yes**
$3 \rightarrow 2$	182.87	190.85	**Yes**
$4 \rightarrow 2$	170.79	175.97	**Yes**
$5 \rightarrow 2$	189.62	188.56	No
$6 \rightarrow 2$	181.51	187.15	**Yes**
$7 \rightarrow 2$	190.97	191.54	**Yes**
$1 \rightarrow 3$	180.52	180.38	No
$2 \rightarrow 3$	202.39	196.74	No
$4 \rightarrow 3$	187.25	185.90	No
$5 \rightarrow 3$	189.78	186.40	No
$6 \rightarrow 3$	183.50	185.61	**Yes**
$7 \rightarrow 3$	194.91	192.67	No
$1 \rightarrow 4$	195.89	198.59	**Yes**
$2 \rightarrow 4$	199.66	195.85	No
$3 \rightarrow 4$	201.56	201.25	No
$5 \rightarrow 4$	201.72	202.46	**Yes**
$6 \rightarrow 4$	190.70	196.11	**Yes**
$7 \rightarrow 4$	199.43	197.45	No
$1 \rightarrow 5$	258.89	258.78	No
$2 \rightarrow 5$	264.32	263.19	No
$3 \rightarrow 5$	251.75	256.51	**Yes**
$4 \rightarrow 5$	257.90	259.99	**Yes**
$6 \rightarrow 5$	260.08	264.70	**Yes**
$7 \rightarrow 5$	255.40	257.99	**Yes**
$1 \rightarrow 6$	273.07	270.39	No
$2 \rightarrow 6$	262.56	270.06	**Yes**
$3 \rightarrow 6$	257.68	266.95	**Yes**
$4 \rightarrow 6$	257.65	262.02	**Yes**
$5 \rightarrow 6$	264.48	274.56	**Yes**
$7 \rightarrow 6$	259.95	277.45	**Yes**
$1 \rightarrow 7$	223.52	231.08	**Yes**
$2 \rightarrow 7$	223.46	230.42	**Yes**
$3 \rightarrow 7$	218.66	222.15	**Yes**
$4 \rightarrow 7$	220.30	220.65	**Yes**
$5 \rightarrow 7$	227.56	229.55	**Yes**
$6 \rightarrow 7$	233.99	236.28	**Yes**
Mean values:	212.03	214.56	**Yes**

Table 5.5: *Brenner focus measures between initial and optimized homographies for "Synthetic Dataset No Clutter".*

From/To	Initial	Optimized	Improved
$2 \rightarrow 1$	375.36	370.70	No
$3 \rightarrow 1$	338.79	348.03	**Yes**
$4 \rightarrow 1$	274.29	286.16	**Yes**
$5 \rightarrow 1$	267.67	277.59	**Yes**
$1 \rightarrow 2$	395.16	388.53	No
$3 \rightarrow 2$	427.43	435.93	**Yes**
$4 \rightarrow 2$	390.08	390.00	No
$5 \rightarrow 2$	376.12	384.29	**Yes**
$1 \rightarrow 3$	386.52	367.49	No
$2 \rightarrow 3$	458.80	454.95	No
$4 \rightarrow 3$	369.41	366.49	No
$5 \rightarrow 3$	358.84	364.31	**Yes**
$1 \rightarrow 4$	320.11	305.80	No
$2 \rightarrow 4$	357.62	350.67	No
$3 \rightarrow 4$	324.01	323.75	No
$5 \rightarrow 4$	311.89	336.22	**Yes**
$1 \rightarrow 5$	339.86	324.85	No
$2 \rightarrow 5$	376.64	371.81	No
$3 \rightarrow 5$	338.12	337.69	No
$4 \rightarrow 5$	344.32	354.21	**Yes**
Mean values:	356.55	357.95	**Yes**

Table 5.6: *Brenner focus measures between initial and optimized homographies for a subset of the "castle-P19" dataset.*

From/To	Initial	Optimized	Improved
$2 \rightarrow 1$	905.02	905.24	**Yes**
$3 \rightarrow 1$	860.85	861.60	**Yes**
$4 \rightarrow 1$	896.59	896.49	No
$5 \rightarrow 1$	900.95	900.50	No
$6 \rightarrow 1$	959.59	960.32	**Yes**
$1 \rightarrow 2$	1214.52	1213.13	No
$3 \rightarrow 2$	1102.12	1102.41	**Yes**
$4 \rightarrow 2$	1078.54	1081.07	**Yes**
$5 \rightarrow 2$	1032.79	1032.85	**Yes**
$6 \rightarrow 2$	1214.88	1211.53	No
$1 \rightarrow 3$	1522.66	1521.15	No
$2 \rightarrow 3$	1479.35	1479.87	**Yes**
$4 \rightarrow 3$	1379.52	1380.98	**Yes**
$5 \rightarrow 3$	1315.53	1315.06	No
$6 \rightarrow 3$	1521.37	1519.37	No
$1 \rightarrow 4$	1222.84	1219.37	No
$2 \rightarrow 4$	1184.87	1183.27	No
$3 \rightarrow 4$	1162.97	1162.77	No
$5 \rightarrow 4$	1094.96	1094.83	No
$6 \rightarrow 4$	1226.83	1223.47	No
$1 \rightarrow 5$	1052.79	1053.10	**Yes**
$2 \rightarrow 5$	999.02	999.69	**Yes**
$3 \rightarrow 5$	942.26	942.85	**Yes**
$4 \rightarrow 5$	993.18	992.71	No
$6 \rightarrow 5$	940.85	939.90	No
$1 \rightarrow 6$	534.28	535.32	**Yes**
$2 \rightarrow 6$	505.69	508.30	**Yes**
$3 \rightarrow 6$	480.64	482.26	**Yes**
$4 \rightarrow 6$	485.26	489.07	**Yes**
$5 \rightarrow 6$	485.29	487.30	**Yes**
Mean values:	1023.20	1025.82	**Yes**

Table 5.7: *Brenner focus measures between initial and optimized homographies for "Objects on surface" dataset.*

6

Synchronization over the Birkhoff Polytope for Multi-Graph Matching

In this chapter we address the problem of simultaneously matching multiple graphs imposing cyclic or transitive consistency among the correspondences. This is obtained through a synchronization process that projects doubly-stochastic matrices onto a consistent set. Contrary to most approaches in the literature, we do not set-up an expensive global objective function, but rather try to enforce the consistency as a constraint on the set of doubly-stochastic matrices, a relaxed assignment space used in several graph-matching algorithms. We overcome the lack of group structure of the Birkhoff polytope, *i.e.*, the space of doubly-stochastic matrices, by making use of the Birkhoff-Von Neumann theorem stating that any doubly-stochastic matrix can be seen as the expectation of a distribution over the permutation matrices, and then cast the synchronization problem as one over the underlying permutations. This allows us to transform any graph-matching algorithm working on the Birkhoff polytope into a multi-graph matching algorithm. We evaluate the performance of two[1] classic graph matching algorithms in their synchronized and unsynchronized versions with a state-of-the-art multi-graph matching approach, showing that synchronization can yield better and more robust matches.

6.1 Introduction

Graph-based representations have found widespread application in several domains due to their ability to characterize complex systems in terms of parts and relations, capturing the fundamental state of the system in a way that is invariant to transformations that are irrelevant to the classification task at hand. Concrete examples include the use of graphs to represent shapes [154], metabolic networks [76], protein structure [72], and road maps [78]. However, this enhanced expressive power comes at the cost of the inability to utilize most of the pattern analysis tool set directly and in general in the requirement of using approaches that are computationally more demanding.

[1] In the experimental section we synchronized a third graph matching algorithm actually, but due to its special adjustments to incorporate the proposed method we decided to mention it separately.

Structural pattern recognition in its first 30 years of research has mainly focused its attention to the graph matching problem as the fundamental means of dealing with structural representation and assessing their similarity [36]. In fact, with correspondences at hand, standard similarity-based recognition and classification techniques can be imported to the structural domain. However, graph matching is in general very computationally demanding and can introduce bias in the inference process [172].

Alternatively, graphs can be embedded in a low-dimensional pattern space using either multidimensional scaling, non-linear manifold leaning techniques, or by adopting the famous kernel trick through the definition of graph kernels [18,67,79,152]. One drawback of these approaches is that they neglect the locational information for the substructures in a graph, thus limiting the precision of the resulting similarity measures.

More recently, in an attempt to increase matching performance and reducing the bias in the inference process, some researchers have started to study the problem of simultaneously extracting correspondence information from whole sets of graphs, rather than limiting the analysis to each pair. In this multi-graph matching setting, we aim at improve correspondence estimation by incorporating transitivity constraints among the matches. Namely, if node v_u^a in graph G_a matches node v_v^b in graph G_b and, in turn, the latter node v_v^b matches node v_w^c in graph G_c, then node v_u^a in G_a must match node v_w^c in G_c.

Williams *et al.* [186] impose the transitive vertex-matching constraint in a softened Bayesian manner, favouring inference triangles through fuzzy compositions of pairwise matching functions. Sole-Ribalta and Serratosa [158] extended the Graduated Assignment algorithm [57] to the multi-graph scenario by raising the assignment matrices associated to pair of graphs to assignment hypercube, or tensors, between all the graphs. For computational efficiency, the hypercube is constructed via sequential local pair matching, but still result in a potentially exponential expansion of the state space. More recently, Yan *et al.* [191,192] proposed a new framework explicitly extending the Integer Quadratic Programming (IQP) formulation of pairwise matching to the multi-graph matching scenario. The resulting IQP is then solved through alternating optimization approach. Junchi Yan *et al.* [189, 190] introduced a method to iteratively approximating the global-optimal affinity matching score in a pool of graphs using the consistency between all the pairwise matching as a regularizer for the whole process. Conversely, Xiowei Zhou *et al.* [199] avoided the semi-definite programming formulation (SDP) proposing a method for multi-image matching as a low-rank matrix recovery problem based on the nuclear-norm relaxation. Pachauri *et al.* [123] and Schiavinato *et al.* [1] on the other hand, start from given pairwise correspondence estimations, and synchronize them, that is finding the set of correspondences that satisfy the transitivity constraint that are closer to the given ones in the least squares sense.

The advantage of this permutation synchronization approach is that it can be paierd with any given graph-matching algorithm in the literature it does not require any additional memory other than what is required to store the original $\binom{N}{2}$ correspondences among N graphs. However, it offers only an *ex post* correction through a relaxation process and cannot be fully integrated with an iterative matching process to direct its convergence to a better solution.

6.1.1 Contribution

In this chapter we aim at extending the synchronization approach in such a way that it can be used within well-known graph-matching approaches transforming them into multi-graph matching algorithms. In particular, we aim at defining a synchronization process for doubly-stochastic matrices, a probabilistic relaxation of correspondence matrices commonly used as a state space in several iterative matching processes [30, 57, 193].

The problem with defining a synchronization process over the doubly-stochastic matrices, is that, contrary to the permutation or orthogonal groups used in other approaches, the Birkhoff polytope does not have a group structure necessary even for defining the notion of transitivity.

Here, we use the Birkhoff-Von Neumann theorem stating that any doubly-stochastic matrix can be seen as the expectation of a distribution over the permutation matrices, and synchronize the doubly-stochastic matrices by implicitly constructing a low entropy distribution over synchronized permutations that fit the given observations in a least squares sense.

6.2 Synchronization over the Birkhoff Polytope

The Birkhoff-Von Neumann theorem states that any doubly stochastic matrix can be constructed as the convex linear combination of a set of permutation matrices. This implies that, given an ideal probability distribution $\mathbf{q} = (q_1, q_2, \ldots, q_k, \ldots)^T$ over the group Σ_n of $n \times n$ permutation matrices, the expected value of such distribution

$$\mathbf{O} = \langle \mathbf{q}, \mathbf{P} \rangle = \sum_k q_k \mathbf{P}^k$$

is a doubly-stochastic matrix and that any doubly-stochastic matrix can be constructed in this way. Unfortunately, this construction is not unique and several distributions lead to the same expected value. In general, however, we are interested in sparse, low entropy distributions. We exploit this property to lower the definition of transitivity to that over the permutation group Σ_n and then raise it back to the Birkhoff polytope.

Let $\{\mathbf{P}_{ij} \in \Sigma_n\}_{i,j=1}^{N,N}$ be a set of permutation matrices. We say that they satisfy the *transitivity* property if

$$\mathbf{P}_{ij}\mathbf{P}_{jk} = \mathbf{P}_{ik} \qquad \forall i, j, k = 1, \ldots, N. \tag{6.1}$$

It can be shown [1] that if the matrices $\mathbf{P}_{ij}$ are transitive, then there exist a reference canonical ordering of the vertices and a set $\{\mathbf{Q}_i \in \Sigma_n\}_{i=1}^{N}$ of alignment matrices that map vertices in G_i to the reference order, such that

$$\mathbf{P}_{ij} = \mathbf{Q}_i \mathbf{Q}_j^T \qquad \forall i, j = 1, \ldots, N. \tag{6.2}$$

Let $\mathbf{P}_{ij}^k = \mathbf{Q}_i^k (\mathbf{Q}_j^k)^T$ be a sequence of transitive permutation matrices, where k is the sequence index, while i, j span over the set of graphs. Further, let $\mathbf{q}$ be a distribution over

the sequence, then

$$\mathbf{O}_{ij} = \sum_k q_k \mathbf{P}_{ij}^k \qquad \forall i, j = 1, \dots, N \tag{6.3}$$

forms a set of doubly-stochastic matrices over the given graphs that are composed as expectation of permutations that satisfy the transitivity property. We say that any doubly-stochastic matrix thus constructed is *transitive*. Hence, the problem of synchronization over the Birkhoff polytope can be reduced to that of finding the transitive set of doubly-stochastic matrices closest to a given set in a least squares sense. However, the search space is huge, $O(n!^N)$ where N is the number of graphs and n is the number of nodes in each graph. We solve this by looking for a sparse distribution $\mathbf{q}$ over the set of transitive permutations. This is achieved through the introduction of an entropic regularizer over $\mathbf{q}$ yielding to the minimization problem as follows:

$$\arg\min_{\mathbf{q},\bar{\mathbf{Q}}} \sum_{i,j=1}^{N} ||\mathbf{O}_{ij} - \sum_{k}^{n!^N} q_k \mathbf{Q}_i^k \mathbf{Q}_j^{k^T}||_F^2 + \lambda H(\mathbf{q}), \tag{6.4}$$

where $\lambda \in \mathbb{R}$ is a free scaling parameter and $H(\mathbf{q}) = -\sum_k q_k \ln(q_k)$ denotes the entropy function.

Assuming the sparsity of the resulting $\mathbf{q}$, we find an approximate solution to the problem (6.4) through *Matching Pursuit*.

Let $\mathbf{R}_{ij}^{(t)} = \sum_k^{n!^N} q_k^{(t)} \mathbf{Q}_i^k \mathbf{Q}_j^{k^T}$ with $i = 1, \dots, N$ be the set of synchronized doubly stochastic matrices at iteration t, we can write the solution at the next iteration as

$$\mathbf{R}_{ij}^{(t+1)} = (1-\alpha)\mathbf{R}_{ij}^{(t)} + \alpha \mathbf{Q}_i^{\hat{k}^{(t+1)}} \mathbf{Q}_j^{\hat{k}^{(t+1)^T}}, \tag{6.5}$$

where $\hat{k}^{(t+1)}$ is the index which denotes the optimal residual alignment and α is a value in $[0, 1]$. Moreover, under the sparsity assumption, we can assume that we only bring in new entries over the distribution $\mathbf{q}$, so the update step for the probability distribution $\mathbf{q}$ becomes

$$\mathbf{q}^{(t+1)} = (1-\alpha)\mathbf{q}^{(t)} + \alpha \mathbf{e}^{\hat{k}^{(t+1)}}, \tag{6.6}$$

where $\mathbf{e}^{\hat{k}^{(t+1)}}$ is a vector of zeros where the unique one is placed in position $\hat{k}^{(t+1)}$. This assumption on $\mathbf{q}$ allows us to ignore the entropic term $\lambda H(\mathbf{q})$ from (6.4).

With this formulation, the matching pursuit iteration is computed by solving

$$\min_{\hat{k},\alpha} \sum_{i,j=1}^{N} ||\mathbf{O}_{ij} - (1-\alpha)\mathbf{R}_{ij}^{(t)} - \alpha \mathbf{Q}_i^{\hat{k}} \mathbf{Q}_j^{\hat{k}^T}||_F^2 + \lambda H\big(\mathbf{q}^{(t+1)}\big). \tag{6.7}$$

It is worth to be noted that $\hat{k}$ does not depend by α, thus we can iteratively solve for $\hat{k}$, and then for α given the current set of correspondences introduced into the reconstruction of the doubly stochastic matrices.

6.2.1 Solving for $\hat{k}$

Let the matrix $\mathbf{M}_{ij}^{(t)} = \mathbf{O}_{ij} - (1-\alpha)\mathbf{R}_{ij}^{(t)}$, we can rewrite the objective function in the problem (6.7) without considering the entropic term $\lambda H(\mathbf{q}^{(t+1)})$ as follows:

$$\begin{aligned} & \sum_{i,j=1}^{N} ||\mathbf{O}_{ij} - (1-\alpha)\mathbf{R}_{ij}^{(t)} - \alpha \mathbf{Q}_i^{\hat{k}} \mathbf{Q}_j^{\hat{k}^T}||_F^2 \\ = & \sum_{i,j=1}^{N} ||\mathbf{M}_{ij}^{(t)} - \alpha \mathbf{Q}_i^{\hat{k}} \mathbf{Q}_j^{\hat{k}^T}||_F^2 \\ = & \sum_{i,j=1}^{N} ||\mathbf{M}_{ij}^{(t)}||_F^2 + \alpha^2 n^2 - 2\alpha \mathrm{Tr}\Big(\mathbf{Q}_i^{\hat{k}} \mathbf{Q}_j^{\hat{k}^T} \bar{\mathbf{M}}_{ij}^{(t)}\Big). \end{aligned} \tag{6.8}$$

Note that the optimization over the index $\hat{k}$ in the set of synchronized permutations can be substituted for the direct optimization over the set of synchronized permutation $\bar{\mathcal{Q}} = \{\mathbf{Q}_i\}_{i=1}^N$. Further, under the assumption that α is small, we can set $\bar{\mathbf{M}}_{ij}^{(t)} = \mathbf{O}_{ij} - \mathbf{R}_{ij}^{(t)}$ resulting in the optimization problem

$$\arg\max_{\bar{\mathcal{Q}}} \sum_{i,j=1}^{N} \mathrm{Tr}\Big(\mathbf{Q}_i {\mathbf{Q}_j}^T \bar{\mathbf{M}}_{ij}^{(t)}\Big), \tag{6.9}$$

which can be solved with any approach extracting synchronized permutations, such as [1].

6.2.2 Solving for α

The entropic term $H\big(\mathbf{q}^{(t+1)}\big)$ can be written explicitly as follows:

$$H\big(\mathbf{q}^{(t+1)}\big) = -\sum_{k} \big((1-\alpha)q_k + \alpha\delta_{k\hat{k}}\big) \ln\big((1-\alpha)q_k + \alpha\delta_{k\hat{k}}\big), \tag{6.10}$$

where $\delta_{k\hat{k}}$ denotes the Kronecker delta operator. This can be re-written as:

$$\begin{aligned} H\big(\mathbf{q}^{(t+1)}\big) &= -(1-\alpha)\sum_{k\neq\hat{k}} q_k\big(\ln q_k + \ln(1-\alpha)\big) - \alpha\ln\alpha \\ &= (1-\alpha)H\big(\mathbf{q}^{(t)}\big) - (1-\alpha)\ln(1-\alpha) - \alpha\ln\alpha \\ &= (1-\alpha)H\big(\mathbf{q}^{(t)}\big) + H(\alpha), \end{aligned} \tag{6.11}$$

where $H(\alpha)$ is the binary entropy function in α. The problem (6.7) can be solved by gradient descent of the energy function:

$$E = \sum_{i,j=1}^{N} ||\mathbf{O}_{ij} - (1-\alpha)\mathbf{R}_{ij}^{(t)} - \alpha \mathbf{Q}_i^{\hat{k}^{(t+1)}} \mathbf{Q}_j^{\hat{k}^{(t+1)}{}^T}||_F^2 + \lambda H\big(\mathbf{q}^{(t+1)}\big). \tag{6.12}$$

Differentiating E with respect to α yields:

$$\begin{aligned}\frac{dE}{d\alpha} &= -\lambda\left(H(\mathbf{q}^{(t)}) + \ln\left(\frac{\alpha}{1-\alpha}\right)\right) + 2\sum_{i,j=1}^{N}\Big[\alpha||\mathbf{R}_{ij}^{(t)} - \mathbf{Q}_i^{\hat{k}^{(t+1)}}\mathbf{Q}_j^{\hat{k}^{(t+1)}T}||_F^2 \\ &\quad +\mathrm{Tr}\Big((\mathbf{O}_{ij} - \mathbf{R}_{ij}^{(t)})^T(\mathbf{R}_{ij}^{(t)} - \mathbf{Q}_i^{\hat{k}^{(t+1)}}\mathbf{Q}_j^{\hat{k}^{(t+1)}T})\Big)\Big] \qquad (6.13)\end{aligned}$$

and, with the derivative to hand, we extract the optimal $\alpha \in [0,1]$ to reconstruct the new solution using (6.5). We minimize the energy E by gradient descent as follows:

$$\alpha^{(t+1)} = \alpha^{(t)} - \eta\frac{dE}{d\alpha}\left(\alpha^{(t)}\right),$$

where $\eta \in \mathbb{R}^+$ is a free parameter and $\alpha^0 = 0$. We control the constrain that $\alpha \in [0,1]$ suspending the power iteration by the following rule:

$$\alpha = \begin{cases} 0 & \text{if } \alpha^{(t+1)} < 0 \\ 1 & \text{if } \alpha^{(t+1)} > 1. \end{cases}$$

For numerical reasons may occur sometimes that $\ln\left(\frac{\alpha}{1-\alpha}\right) \notin \mathbb{R}$ and is not finite, we fixed the problem replacing that term with 0. Note that, $\alpha = 0$, or more unlikely $\alpha = 1$, means that the basic pursuit step cannot reduce the entropy-regularized energy and we take that as a stopping criterion for our basis pursuit approach.

6.3 Synchronized Algorithms

We introduce the fundamental algorithms in our work synchronizing two well-known graph matching methods operating in the Birkhoff Polytope, namely Graduated Assignment [57] and Path Following [193]. In particular, we included a synchronization step inside their main updating loops, maintaining the relaxed correspondences consistency among all the graphs throughout the execution.

Since we followed a common scheme to implement both the proposals, we can give a very general description as the Pseudocode 1, where $\mathcal{T}$ is just a wild-card which refers to the underlying method. Our synchronization consists in an on-line process, so we needed to extend the original graph matching method $\mathcal{T}$ in a multi-graph setting from its internal structure. The routine $\mathcal{T}$GRAPHMATCHINGCORE denotes just the principal section of a method which solves a pairwise matching and that depends only by the shared β parameter for all possible couples of graphs.

Algorithm 1 Synchronized Algorithm $\mathcal{T}$ for Multi-graph Matching

Input: $\mathcal{G}$
Output: $\mathcal{P}$

$\triangledown$ *Initialization of* β, λ_s *and* $\mathcal{O}$ *according* $\mathcal{T}$ *(For more details view the text).*

repeat
  for each $\mathbf{O}_{pq} \in \mathcal{O}$ **do**
    $\mathbf{O}_{pq} \leftarrow \mathcal{T}$GRAPHMATCHINGCORE$(G_p, G_q, \beta)$
  end for
  $\lambda \leftarrow \lambda_s \beta$
  $\mathcal{O} \leftarrow$ SYNCHRONIZATION$(\mathcal{O}, \lambda)$
  $\beta \leftarrow \beta_r \beta$
until $\beta \geq \beta_f$
$\mathcal{P} \leftarrow$ DISCRETIZATION$(\mathcal{O})$

where:

- $\mathcal{G} = \{G_p\}_{p=1}^{N}$ is the set which collects all graphs, *i.e.*, the input dataset;
- $\mathcal{O} = \{\mathbf{O}_{pq}\}_{p,q=1}^{N,N}$ is the set which contains all the double stochastic permutation matrices estimated by a graph matching method;
- $\mathcal{P} = \{\mathbf{P}_{pq}\}_{p,q=1}^{N,N}$ is the discrete version related to $\mathcal{O}$, *i.e.*, the set which contains all the binary permutation matrices representing each $\mathbf{O}_{pq}$;
- (λ_s) is the proportionality factor between β and λ;
- (β_r) is the growth rate for β;
- (β_r) is the exit threshold for β.

Considering a single iteration of the main loop we perform the pairwise matching of a method $\mathcal{T}$ for all the graphs computing the doubly stochastic permutation matrices in $\mathcal{O}$. In either case there is a β parameter which governs the whole process, pushing it towards the vertices of the polytope, *i.e.*, towards permutation matrices. After the pairwise computation of all the doubly stochastic permutation matrices in a given β-iteration, we perform our synchronization process resulting in synchronized doubly stochastic matrices. This phase is controlled by our parameter λ, which we set to be proportional to β since the goal of both parameters consists to push the solution towards the vertices of the polytope. The synchronization step is well explained in detail in the Pseudocode 2 (SYNCHRONIZATION routine). When our synchronized algorithm converges, we discretise the solutions applying the typical maximum bipartite assignment problem through the well-know Hungarian algorithm to each doubly stochastic matrix $\mathbf{O}_{pq} \in \mathcal{O}$ (DISCRETIZATION routine). Note that running the algorithm 1 without our synchronization procedure is equal to perform trivially just the original pairwise matching method $\mathcal{T}$ for all couple of graphs.

Algorithm 2 Synchronization

Input: $\mathcal{O}, \lambda$
Output: $\mathcal{O}^s$

$\triangledown$ *Initialization of alignments* $\mathcal{Q}^{(0)} = \left\{\mathbf{Q}_i^{(0)}\right\}_{i=1}^{N}$ *in the transitive alignment space by [1].*

$\mathcal{Q}^{(0)} \leftarrow$ TRANSITIVEALIGNMENT$(\mathcal{O})$

$\triangledown$ *Initialization of residuals* $\mathcal{R}^{(0)} = \left\{\mathbf{R}_{ij}^{(0)}\right\}_{i,j=1}^{N,N}$.

$\mathbf{R}_{ij}^{(0)} \leftarrow \mathbf{Q}_i^{(0)}\mathbf{Q}_j^{(0)^T}$, for each $i, j = 1 \dots N$

$\triangledown$ *Initialization entropy term and time counter.*

$h^{(0)} \leftarrow 0$

$t \leftarrow 0$

repeat

$\triangledown$ *Updates weighted matrices assuming* $\alpha^{(t)} \to 0$.

$\bar{\mathbf{M}}_{ij}^{(t)} \leftarrow \mathbf{O}_{ij} - \mathbf{R}_{ij}^{(t)}$, for each $i, j = 1 \dots N$

$\triangledown$ *Updates alignments by solving (6.9).*

$\mathbf{Q}_i^{(t+1)} = \frac{\sum_{j=1}^{N} \bar{\mathbf{M}}_{ij}^{(t)}\mathbf{Q}_j^{(t)}}{||\sum_{j=1}^{N} \bar{\mathbf{M}}_{ij}^{(t)}\mathbf{Q}_j^{(t)}||_F}$, for each $i = 1 \dots N$

$\triangledown$ *Updates* α *by gradient descent on the energy (6.12).*

$\alpha^{(t+1)} \leftarrow$ LEARNINGALPHA$(\mathcal{O}, \mathcal{Q}^{(t+1)}, \mathcal{R}^{(t)}, h^{(t)}, \lambda)$

$\triangledown$ *Updates entropy by (6.11),* H *is the binary entropy function.*

$h^{(t+1)} \leftarrow (1 - \alpha^{(t+1)})h^{(t)} + \alpha^{(t+1)}H(\alpha^{(t+1)})$

$\triangledown$ *Updates residuals by (6.5).*

$\mathbf{R}_{ij}^{(t+1)} \leftarrow (1 - \alpha^{(t+1)})\mathbf{R}_{ij}^{(t)} + \alpha^{(t+1)}\mathbf{Q}_i^{(t+1)}\mathbf{Q}_j^{(t+1)^T}$, for each $i, j = 1 \dots N$

$t \leftarrow t + 1$

until Convergence of $\mathcal{Q}^{(t)}$ or a maximum number iterations t is reached.

$\triangledown$ *Estimating synchronized permutations* $\mathcal{O}^s = \left\{\mathbf{X}_{ij}^s\right\}_{i,j=1}^{N,N}$.

$\mathbf{X}_{ij}^s \leftarrow \mathbf{Q}_i^{(t)}\mathbf{Q}_j^{(t)^T}$, for each $i, j = 1, \dots, N$

6.4 Experimental Setup and Evaluation

6.4.1 Synthetic Data Experiments

We began the evaluation of our work by comparing the two main synchronized algorithms as section 6.3 to their un-synchronized counterparts and to a state-of-the-art multi-graph matching algorithm: the Consistency-driven Non-Factorized Alternating Optimization (CDAO) [192]. We summarized the main parameter setting for these 5 methods as follows:

GA, **S-GA**. For the Synchronized Graduated Assignment (S-GA) we initialized $\beta = n$ and all the doubly stochastic matrices are initialized as a random perturbation from the barycentre of the polytope. The entropy scale parameter λ_s, *i.e.*, the

proportionality factor between β and λ, was set to 10^{-6}. Moreover, the growth rate and exit threshold for β were set to 1.075 and 200 respectively.

PF, **S-PF**. For the Synchronized Path Following algorithm (S-PF) we initialized the doubly stochastic matrices $\mathbf{O}_{pq}$ as in the original work [193] performing a convex quadratic optimization problem by Frank-Wolfe algorithm. The entropy scale parameter was set to 1 while the increasing rate for β was 0.15.

CDAO. For this method we respected the original setup for the Consistency-driven Non-Factorized Alternating Optimization algorithm in [192] initializing the max number of iteration $T_{max} = 2$ and using the Reweighted Random Walks [30] as pairwise graph matching solver.

We performed tests over several random synthetic graph datasets with different levels of distortion, variations in edge density and proportion of outlier nodes. This evaluation approach followed the widely adopted protocol [30, 191, 192]. The dataset is generated from a set of N root graphs G^r, $r = 1, \ldots, N$, with n_{in} inlier nodes randomly connected with edge density ρ. Edge attributes $a^r_{i_j}$ are randomly drawn from an uniform distribution in $[0, 1]$. According these root graphs, we generate several perturbed sets, by varying (a) edge attributes adding Gaussian noise sampled from $N(0, \sigma^2)$ for increasing values of σ, (b) edge density ρ, and (c) adding a number of outlier nodes.

We introduced another synthetic test as [158] whose aim is to control the topological structure of the graphs. The construction of a synthetic dataset $\mathcal{G}$ in this experiment is based on the generation of an initial seed $P^r = \{([0, 1]; [0, 1])_i\}_{i=1}^n$ of 2D points which are related to the n nodes. Each perturbed graph is generated through a random Gaussian perturbation of the points in P^r, from which we extract a Delaunay triangulation. The computation of the affinity matrix $\mathbf{M}_{pq} = (m_{ia,jb})$ for each pair of graphs (G^p, G^q) is defined as

$$m_{ia,jb} = \exp\left(-\frac{(a^p_{ij} - a^q_{ab})^2}{\sigma^2}\right) \tag{6.14}$$

where σ^2 is a scale factor which we set to 0.15. No single-node weight is considered, so we set the unary affinity as $m_{ia,ia} = 0$. Considering a single experiment on a synthetic dataset all the performed methods share the same affinities matrices.

We present our results in terms of vertex correspondences from the permutations given by the graph-matching methods. The evaluation strategy is based on the computation of a Matching Accuracy (MA) between the common n nodes of two graphs $G^p, G^q \in \mathcal{G}$, which is defined as the ratio between the number of correspondences found (C^{ALG}_{pq}) with respect to those of the ground truth (C^{TRU}_{pq}) and the total number of possible matching as follows:

$$\mathrm{MA}(G^p, G^q) = \frac{|C^{ALG}_{pq} \cap C^{TRU}_{pq}|}{n}$$

We underline that we only calculate the accuracy for common inlier nodes ignoring the matching results over outliers. Given a whole dataset $\mathcal{G}$ of N graphs, the agglomerated

matching accuracy (MA) can be expressed as the mean measure:

$$\mathrm{MA}(\mathcal{G}) = \frac{\sum_{p=1}^{N-1}\sum_{q=p+1}^{N}\mathrm{MA}(G^p, G^q)}{N(N-1)/2} \tag{6.15}$$

In Figure 6.1 we plot the final results of all the synthetic tests varying the parameter of (a) deformation, (b) edge density, (c) number of outlier nodes and (d) topological noise (vertex jitter). All these experiments are repeated over 10 trials, for which we plot average and standard error. Each synthetic dataset has $N = 10$ graphs with $n_{in} = 20$ nodes. For the deformation and edge density tests we set $n_{out} = 0$, for outlier and density tests we set the Gaussian deformation with standard deviation as $\sigma = 0.05$ and $\sigma = 0.2$ respectively. Finally, just for the deformation tests, the edge density is set $\rho = 0.7$.

From Figure 6.1 we can see that in general for high deformations the synchronized algorithms are the best performers regardless of the original algorithm chosen and they generally outperform CDAO as well. It is interesting to not that for deformation, edge density and outlier Graduated Assignment seems to be just as robust as the synchronized algorithms, while is exhibited high sensitivity on the topological noise using Delaunay triangulations. On the other hand, the synchronized version of Graduated Assignment seems to under-perform for low topological noise, going back to very high precision for larger noise. On the other hand, the Synchronized Path Following algorithm is almost always the top performer, even when the original Path Following algorithm appears to be the worst-performing of the lot. This appears to point to the fact that path-following and synchronization provide complementary information. Finally, CDAO, which was built as a multi-graph matching algorithm optimizing a global objective function does not seem to offer a real advantage over the synchronized algorithms, performing generally at the level of the worst-performing non-synchronized algorithms.

6.4.2 Further Implementation Experiments

In this section we present another further version of our algorithms with respect to the implementations described in section 6.3. Our goal consists to overcome even the best results we obtained through the experiments in section 6.4.1, which are related mainly to the implementation based on Path Following (SP) [193] algorithm. We propose a new solution as the synchronized multi-graph matching version of the process based on Reweighted Random Walks (RRW) [30], this choice is motivated for two essential reasons: first, RRW represents state of the art in pairwise graph matching approaches available in literature; second, the relaxed definition of the problem in RRW, which fits very well with the definition of our synchronization scheme over the Birkhoff's polytope as section 6.2.

The implementation of our new algorithm **RRW/S-RRW** follows partially the general structure presented in the Pseudocode 1, but we applied some adjustments which are required to respect the original scheme proposed in [30]. For Synchronized Reweighted Random Walks algorithm (S-RRW) we initialized the doubly stochastic matrices $\mathbf{O}_{pq}$ as the barycentre of the polytope. Unfortunately, we cannot model a parameter β to control

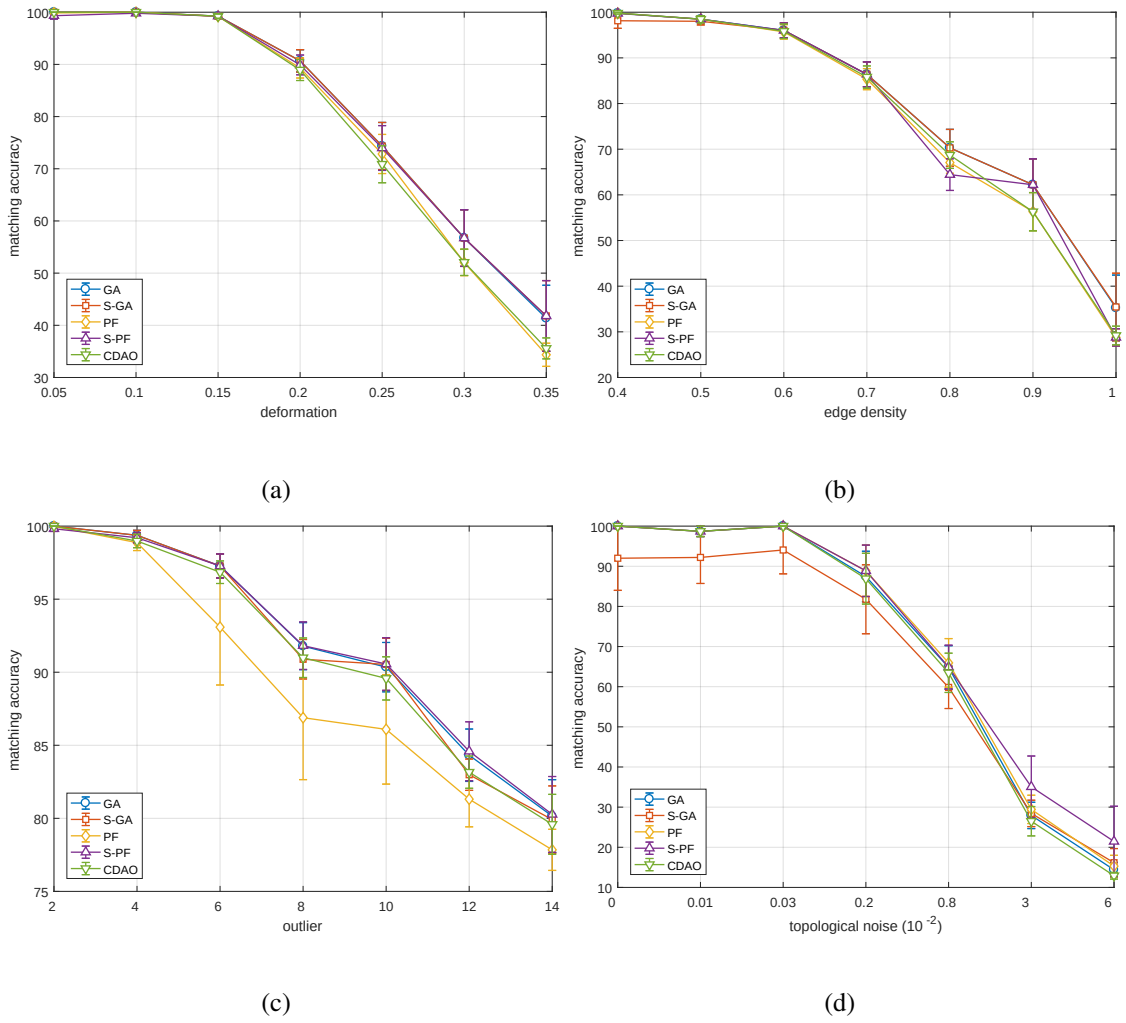


Figure 6.1: *Average results with standard error for synthetic test at varying of the levels of (a) deformation, (b) edge density, (c) number of outlier nodes and (d) topological noise performing Graduated Assignment (GA), Synchronized Graduated Assignment (S-GA), Path Following (PF), Synchronized Path Following (S-PF), and Consistency-driven Non-Factorized Alternating Optimization (CDAO) for Multi-Graph Matching algorithms.*

the learning in this process, hence the final iteration is reached by classical convergence criterion of the solution, *i.e.*, $\mathcal{Q}^{(t+1)} \approx \mathcal{Q}^{(t)}$. For this reason, we does not need to define an increasing rate for β. Moreover, the lack of the latter parameter does not allow us to set a proportional entropy scale parameter λ as proposed in section 6.3, hence we decided to suppress it just setting $\lambda = 0$.

Here, the evaluation strategy follows the same protocol with synthetic datasets and matching accuracy performances introduced in section 6.4.1, but we compare our further novel proposal with respect to the Synchronized Path Following solution only, which is

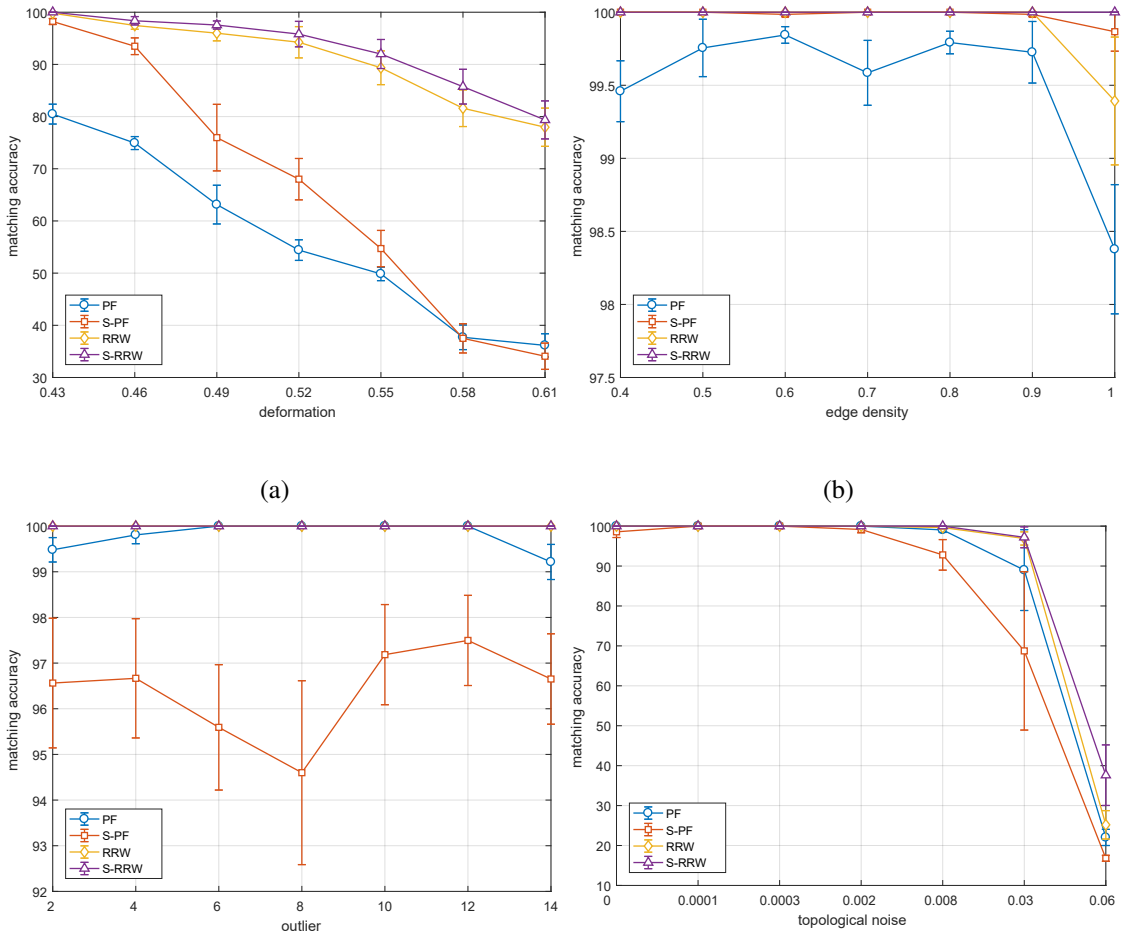


Figure 6.2: *Average results with standard error for synthetic test at varying of the levels of (a) deformation, (b) edge density, (c) number of outlier nodes and (d) topological noise performing Reweighted Random Walks (RRW), Synchronized Reweighted Random Walks (S-RRW), Path Following (PF), and Synchronized Path Following (S-PF).*

the best implementation with respect to state-of-the-art Consistency-driven Non-Factorized Alternating Optimization (CDAO) [192] as well. Moreover, since the Synchronized Reweighted Random Walks (S-RRW) does not use λ, we decide to set the same conditions on Synchronized Path Following (S-PF) in order to guarantee a fair comparison of both methods.

In Figure 6.2 we showed that our novel implementation of Synchronized Reweighted Random Walks (S-RRW) can outperform both the Synchronized Path Following (S-PF) and the unsynchronized graph matching algorithms in all kinds of experiments. The robustness of our proposal is highlighted in test (a) even with high values of graph deformation.

Another interesting aspect is related to the behaviour of Path Following algorithm in (c) and (d) tests which differs with respect to the results in Figure 6.1. The performances with this process do not increase as expected by introducing the synchronization step. In particular on plot (c) the learning even fails totally. In the first place, we guess that such weaknesses is mainly due to a local minimum which is reached activating the synchronization. Generally speaking, we assumed the main explanation about the new limits of our approach with Path Following is related to two key factors. First, the removal of entropy information by suppressing the parameter λ with respect to the original scheme could make worse the synchronization to deal with noise or general uncertainty of the vertex correspondences, which could affect considerably Path Following process. Second, the common aspect in those experiments, *i.e.*, the (c) and (d), is related to the deep changing of the topologies in the synthetic graphs by introducing outlier nodes or noise in Delaunay triangulation. This final consideration is further supported by another marginal experiment as in Figure 6.5, where we generate synthetic graph datasets by removing randomly an increasing number of edges to ruin the original structures. Moreover, we can observe that Path Following is a very sensitive method on graph topologies in principle. Indeed, starting from just removing beyond two edges, the performances of the unsynchronized version fall critically.

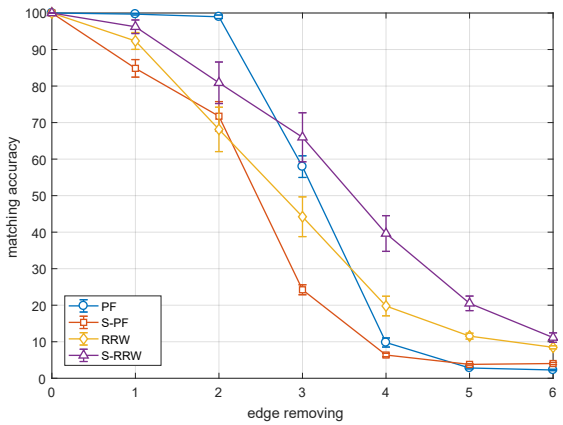


Figure 6.3: *Average results with standard error for synthetic test at removing edges in the graphs performing Reweighted Random Walks (RRW), Synchronized Reweighted Random Walks (S-RRW), Path Following (PF), and Synchronized Path Following (S-PF).*

Since the results of this new implementation are experimentally promising, we decided to include also the Synchronized Reweighted Random Walks algorithm even in the next experiments as Sections 6.4.3 and 6.4.4. Moreover, in the comparison with respect to Graduated Assignment (GA) and Path Following (PF) we maintain their original settings as in Section 6.4.2.

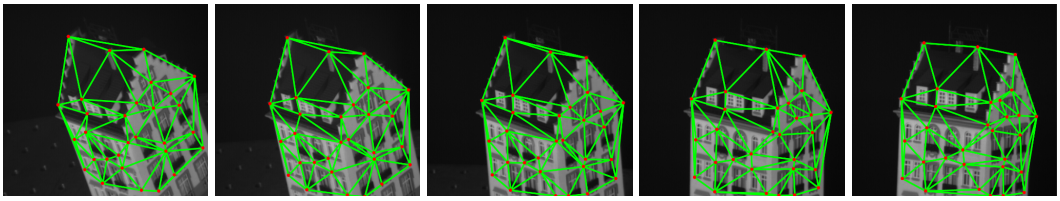


Figure 6.4: *Example of several graphs derived by Delaunay triangulation (green edges) onto 30 landmark feature points (red nodes) from the motion frames in CMU Hotel sequence.*

6.4.3 Real-world Data Experiments

The synthetic evaluation of our algorithms gave us the ability to simulate and control different graph dataset scenarios with good performances. In this section we are aimed to analyse the synchronization even on real-world samples as well. The data used for such tests are the CMU *House* and *Hotel* motion sequences, which consist in two well-known datasets largely employed for Computer Vision tasks [30, 191, 198]. The adjacency matrices to describe the structures of the graphs are obtained by a Delaunay triangulation that is performed onto 30 keypoints manually extracted for each raw image as Figure 6.4. The computation of the affinity edge matrices between two graphs follows the same formulation as (6.14), but considering the divergence of the Euclidean distances between the 2D points associated for each node. The original datasets contain 101 ordered frames which describe very short movements (rotation) of the main subjects. In our experiments we made two reduced versions in order to accentuate the variability of the instances taking one every 10 in the sequence, therefore our resulting datasets contain 10 graphs.

Table 6.1 summaries the final results with our (un)synchronized algorithms and state-of-the-art in terms of matching accuracies as (6.15).

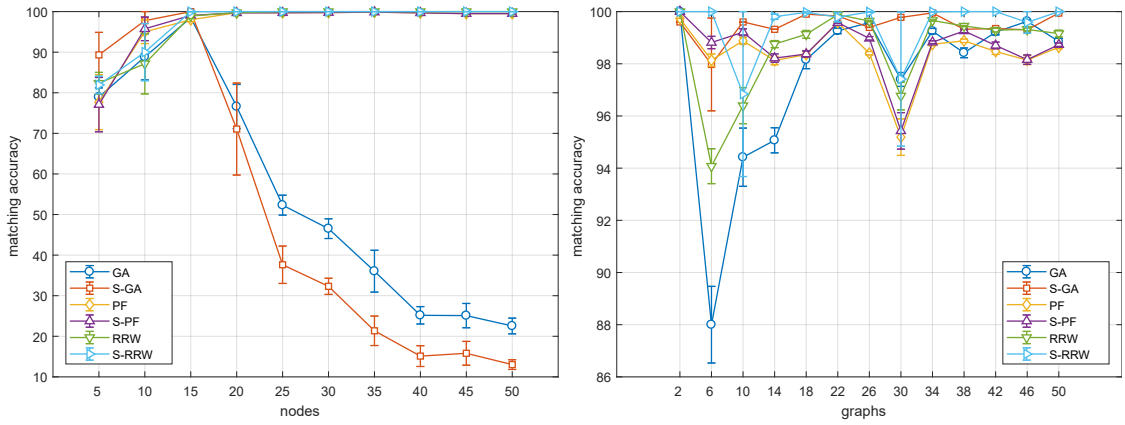
Method	CMU House	CMU Hotel
GA	34.7 ± 2.5	35.8 ± 2.6
S-GA	35.6 ± 2.6	33.6 ± 2.2
PF	46.7 ± 6.2	40.1 ± 7.1
S-PF	59.7 ± 5.6	50.5 ± 6.9
RRW	47.4 ± 4.5	57.0 ± 6.0
S-RRW	$\mathbf{82.7 \pm 1.6}$	68.0 ± 3.0
CDAO	51.1 ± 4.3	$\mathbf{68.1 \pm 4.3}$

Table 6.1: *Mean matching accuracy (%) with standard error for CMU datasets performing Graduated Assignment (GA), Synchronized Graduated Assignment (S-GA), Path Following (PF), Synchronized Path Following (S-PF), Reweighted Random Walks (RRW), Synchronized Reweighted Random Walks (S-RRW), and Consistency-driven Non-Factorized Alternating Optimization (CDAO) for Multi-Graph Matching algorithms.*

The general comment about the results with these real-world datasets consists that applying our method is inclined to improve or remain statistically balanced with the performances by the unsynchronized counterpart algorithms. Moreover, for each dataset our Reweighted Random Walks process registered a greater divergence with respect to the baseline accuracy than Path Following and Graduated Assignment. Finally, we outperformed state-of-the-art performances on CMU House dataset by our Synchronized Reweighted Random Walks algorithm.

6.4.4 Dimensionality Analysis

In the various experiments presented so far we followed a general setting that operates mainly on synthetic (sections 6.4.1 and 6.4.2) and real-world (section 6.4.3) datasets which are quite small both in terms of graphs and nodes magnitudes. Clearly, we are interested to analyse in deep the behaviour of our method with respect to the dimensionality of the sample data as well. In Figure 6.5 we presented the results of two experiments employing our synchronized algorithms, respectively increasing the number of (a) nodes and (b) graphs of the synthetic datasets. The common setting for these two experiments consists to generate random graphs with edge density $\rho = 1$ and deformation $\sigma = 0.2$, while for the graph test the number of nodes is fixed to 15 and for the nodes test the number of graphs is fixed to 20.



(a) (b)

Figure 6.5: *Average results with standard error for synthetic test at varying the number of (a) nodes and (b) graphs of the datasets performing Graduated Assignment (GA), Synchronized Graduated Assignment (S-GA), Path Following (PF), Synchronized Path Following (S-PF), Reweighted Random Walks (RRW), and Synchronized Reweighted Random Walks (S-RRW).*

The results in the (a) tests show a sharp saturation of the average accuracy already beyond a set of graphs with 15 nodes. We can retain that the impact onto the accuracy

by the erroneous correspondences becomes more predominant operating with small graph since that cannot be smoothed in mean terms. Moreover, this trend occurs just for the unsynchronized multi-graph methods, therefore our synchronized extensions could not do better at those high nodes values intuitively. Furthermore, in the same critical point Graduated Assignment (GA) falls critically, we retain that this process is very sensitive with respect to the nodes and the impact of a proper initialization emerges already from low dimensions (at least in our experiments). Since the pairwise matches are corrupted, the synchronization cannot exploit of safe information to correct the bias. The performances measured in the (b) test show that the number of graphs is a parameter that synchronization can tolerate very well and even with low values as well. The algorithms begin to stabilize beyond 18 graphs (the curious fall at 30 is due to randomness surely, *i.e.*, there are no reason that may be a repetitive trend), but we can observe that the gain in performance with synchronization is not proportional with respect the number of graphs. Although, this aspect is quite expectable considering our synthetic setting, since datasets contains the same level of perturbation. Overall, we may deduce by the experiments (a) and (b) that our synchronization works properly increasing the dimension of the problem, but the magnitude does not involve significantly to the accuracy of the learning process: in other terms, if the quality of the data is satisfactory the synchronization is not affected by the scale.

6.5 Conclusion

In this chapter we proposed a synchronization process for doubly stochastic matrices which is set as a basis pursuit over the set of synchronized permutations. Through this approach we can transform any graph-matching algorithm working over the Birkhoff polytope into a multi-graph matching scheme simply maintain the states synchronized throughout the execution. We used this approach to create multi-graph versions of the Graduated Assignment and Path Following methods, and show that the resulting synchronized algorithms outperform not only the original unsynchronized processes, but also state-of-the-art in multi-graph matching. Finally, we proposed a third experimental work based on Reweighted Random Walks for multi-graph matching, which is revealed to be very competitive by relaxing the definition in the proposed method.

7

Subgraph Generalization on Transitive Correspondence

In several real-world contexts reliable techniques of inlier selection are extensively required due to the presence of noise in learning data. These methods help to refine properly parameter model or more generally identify wrong features that not satisfy desired properties in structured objects. We consider just the applications exploiting binary correspondences as essential information to filter outliers, whose reference schema can be yielded to a subgraph matching problem. In fact, due to complexity of this task, the current solutions are just approximations of the original problem to infer partial matching between pair of graphs with different size. Recently, graph matching has been generalized to deduce a consistent solution even on multiple data. This new paradigm aims to reduce bias in local matches, but the underlying formulation is unsuitable to run directly subgraph matching, since it assumes complete correspondences between nodes. In this chapter we present a generalization for the problem to estimate the set of consistent partial constrains over multiple graphs with arbitrary size, which can be employed as a synchronization framework for multi-subgraph matching.

7.1 Introduction

In several fields of science as Machine Learning, Pattern Recognition, Computer Vision and Statistics, the quality and quantity of data are two aspects that affect deeply on the inference task. Although, high magnitude of information does not yield to derive better knowledge perforce, consequently the overall increasing of noise in real-world observations. The latter refers to the well-known Curse of Dimensionality phenomenon, which supports the principle that would be sufficient a reduced set of representative features to describe the instances actually, which may be related to distinguish between inlier and outlier entities as well. Moreover, from another prospective, the requirement to infer on subsets of features can be motived when some global problem cannot be solved totally, but looking for a partial solution is still reasonable.

In the panorama of learning techniques that use correspondences between structured data, we find several applications which deal with noise problems. In Computer Vision

are widely employed Bundle Adjustment [175, 188] processes to refine camera parameters or RANSAC-based methods [32, 44] to improve accuracy for coarse point registration [29, 135]. Furthermore, considering matches just as binary constraints between labelled entities [148], we introduce the generalization of the Graph Matching [36] problem, whose correspondences can be inferred even between a reduced number of inlier nodes. There are two fundamental advantages by solving this general task: first, working with graph of different sizes, the solution can be inferred directly without adding dummy nodes; second, in presence of noisy data graphs, the best candidate matches may help to identify which are potential outlier nodes.

Subgraph matching between two graphs can be treated as the well-known subgraph isomorphism [42] relaxing the space of constrains and setting other heuristic assumptions [45]. Generally, the search space is based on a branch-and-bound strategy defined over the domain of all possible matches, which is typically represented as a tree structure, *i.e.*, the space search tree (SST). The main works of subgraph matching are differentiated just as their way to traverse such SST and pruning the unfair branches [38, 179]. Other approaches overcome such representations formulating the process as a Quadratic Assignment Problem (QAP), where the relaxed permutations are solved by maximizing non linear cost function over the edges [30, 57, 193].

The majority of techniques which solve matching problems encloses the inference task just assuming pairwise formulation with respect to the number of available objects independently. Nevertheless, the introduction of further graphs during the learning could be a useful strategy to smooth the bias in local solutions. Indeed, in graph matching community there is still a limited selection of works that formulate graph matching problems with multiple data globally. There two fundamental strategies that deal with these problems: solving the global mapping optimizing a consistency cost function [3, 157, 191] or synchronizing the constrains imported by out-of-the-box pairwise solvers [1, 123, 192]. These approaches assume the existence of an unknown reference ordering whose transitive constrains can be derived. Although, the common drawback of such works consists in the rigid one-to-one formulation of the consistent correspondences, *i.e.*, they solve just (full) multi-graph matching problem with graphs of the same size.

7.1.1 Contribution

In this chapter we present a novel generalization to derive consistent partial matches given a set of multiple objects. Our work consists in an experimental synchronization framework for set of partial constraints between graphs with arbitrary dimensions, which can be addressed in applications of multi-subgraph matching. Our strategy is mainly inspired with respect to rectification methods [1, 123], therefore we synchronized partial permutations of the graphs independently the specific matching solver. Moreover, we relax the assumption of two-way constrains in order to deal with subgraph problems. Indeed, conversely classical graph matching processes that derive full vertex correspondences from a double stochastic definition [30, 57, 193], we estimate the transitive alignments enforcing the matching likelihoods for the nodes at least in one-direction. In fact, our multi-graph

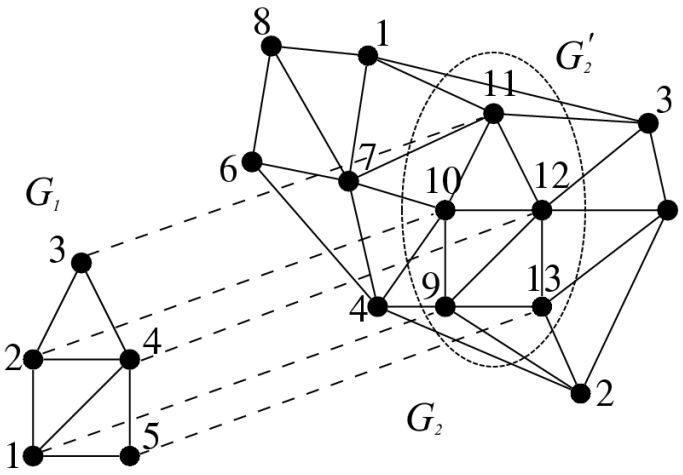


Figure 7.1: *Example of subgraph matching problem between two graphs G_1 and G_2. The picture shows a possible matching of G_1 with respect to the a subgraph G_2' in G_2 which maintains the topology of structures.*

matching approach share several aspects with respect to the current state-of-the-art for these tasks [192], which are the independence of the initial solution and the global consistency by transitive alignments. Although, in our novel formulation we can deal with subgraph problem directly, hence avoiding eventual artificial noise introduced by padding the uneven graphs before the learning.

7.2 Preliminaries

In this section we formalize the fundamental aspects of the subgraph matching problem with respect to the model we followed to devise our work. Moreover, we introduce the reference approach from which we aim to conceive our multi-graph formulation and highlighting some possible obstacles to describe this special generalization.

7.2.1 Subgraph Matching Problem

We introduce the conventional representation of a graph as the pair $G = (V, E)$, where V is the set of nodes and $E = V \times V$ the set of edges. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ two different graphs with respectively number of nodes $n_1 = |V_1|$ and $n_2 = |V_2|$, assuming the case $n_1 \leq n_2$ the subgraph matching consists in the problem to derive the partial injective mapping $f : V_1 \rightarrow V_2$ between the nodes of the (pattern) graph G_1 with respect to a subgraph G_2' in the (target) graph G_2, which is optimal according a certain criterion (as in Figure 7.1). Typically, algorithms exploit of the notion of subgraph isomorphism to infer a solution which reflects the topological similarity between the pair of graphs. We could represent the problem with another formulation just assuming the existence of

a complete (sub)map $f' : V1 \rightarrow V_2'$, which is a bijection that preserves the topologies between the graph G_1 and the subgraph in G_2 (*i.e.*, graph isomorphism). In this way, if $n_1 = n_2$, the approach yields to the typical full graph matching (*i.e.*, unique complete map $f' = f$). There exist another weaker form of subgraph isomorphism which allows the presence of extra edges applying the mapping: the subgraph monomorphism. Although, we specify that in this work we are not interested to model such abstraction, considering always symmetric matching configurations between graphs.

The several approaches of subgraph matching can be divided in three specific categories: *testing*, verifying the existence of at least one match between the graphs; *counting*, estimation of all the occurrences of a graph inside the other one; *listing*, reporting the exact locations of each occurrences. Our method treats the problem of subgraph matching returning one (and only one) solution which is the optimal occurrence of the pattern graph and its exact place with respect to the target graph. We encode such solution as a partial $n_1 \times n_2$ permutation matrix $\mathbf{P} = (p_{ij})$, which models the one-to-one correspondence of the nodes $i \in G_1$ and $j \in G_2$ with the assignment $p_{ij} = 1$ (while is 0 otherwise).

7.2.2 Multi-graph Matching Generalization

Graph matching problem can be generalized extending the inference process beyond the pairwise case, in other terms, working in a set of graphs with cardinality $N \geq 2$ simultaneously. The hard aspect that arises in this new paradigm consists to support the well-known cycle consistency, *i.e.*, the matches between all possible pairs of graphs have to be consistent globally, which is a condition that classical pairwise solvers can not guarantee. Our approach is inspired to the classical formulation [123], which is rooted to introduce a common alignment set $\{\mathbf{X}_i \in \Sigma_n\}_{i=1}^N$ posing the transitive permutation matrices as $\mathbf{P}_{ij} = \mathbf{X}_i \mathbf{X}_j^T, \ \forall i, j = 1, \ldots, N$. Although, the latter is feasible when each graph share the same number of nodes $n = |V_i|, \ \forall i = 1, \ldots, N$ since the alignments has to be defined in the common permutation space Σ_n. The general trick to generalize multi-subgraph matching consists just to add dummy nodes in uneven graphs to exploit of ordinary techniques. However, such halfway practice could enlarge heavy the problem space downgrading the accurateness of the inference task.

7.3 Generalized Transitive Correspondence

Given a couple of graphs G_i and G_j with number of nodes n_i and n_j respectively, we formulate in matricial form the general pairwise solution from any (sub)graph matching technique as the $n_i \times n_j$ cost matrix $\mathbf{P}_{ij}$ whose each entry $(\mathbf{P}_{ij})_{ab} \in [0, 1]$ denotes the weight of the match between their nodes v_a^i and v_b^j.

Let $G_1, G_2, \ldots, G_N$ be a set of N graphs with an arbitrary number of nodes and assuming that all the possible raw pairwise solutions are known, our generalization for the multi-subgraph matching problem consists to estimate each partial $n_i \times n_j$ permutation

matrix $\overline{\mathbf{P}}_{ij}$ which is the nearest in least squares sense to $\mathbf{P}_{ij}$ and transitive as follows

$$\overline{\mathbf{P}}_{ij}\overline{\mathbf{P}}_{jk} = \overline{\mathbf{P}}_{ik} \qquad \forall i,j,k = 1,\dots,N.$$

Defining a direct global transitive formulation is challenging without padding all graphs to the same size, since the space $[0,1]^{n_i \times n_j}$ depends by the dimensions of each couple of graphs. We can generalize this problem just replacing in "one direction" the node dimension n_j by introducing a common universe of nodes $m \geq n_i \ \forall i = 1,\dots,N$ which is encoded as the special space $S_{n_i \times m} \subset [0,1]^{n_i \times m}$ of the matrices whose rows sum to unity. In this way we can retrieve an unknown reference canonical order by defining the alignment $\mathbf{Q}_i \in S_{n_i \times m} \ \forall i = 1,\dots,N$ that express for each vertex of G_i the best candidate matches to such relaxed space, in other terms, each row of $\mathbf{Q}_i$ yields to an element in the common standard simplex Δ_m. Finally, with these matrices on hand we set $\overline{\mathbf{P}}_{ij} = \mathbf{Q}_i\mathbf{Q}_j^T \ \forall i = 1,\dots,N$, that lead back to the real node dimension $n_i \times n_j$.

Therefore, we need to estimate the set of transitive alignments $\mathcal{Q} = \{\mathbf{Q}_i \in S_{n_i \times m}\}_{i=1}^N$ whose reconstructions minimise the difference with respect to the original solution, yielding in the optimization problem as follows

$$\begin{aligned}
&\underset{\mathcal{Q}\in\{S_{n_i\times m}\}^N}{\arg\min} \sum_{i,j=1}^{N} \left\|\mathbf{P}_{ij} - \mathbf{Q}_i\mathbf{Q}_j^T\right\|_F^2 = \\
&\underset{\mathcal{Q}\in\{S_{n_i\times m}\}^N}{\arg\min} \sum_{i,j=1}^{N} \left(\|\mathbf{P}_{ij}\|_F^2 + \left\|\mathbf{Q}_i\mathbf{Q}_j^T\right\|_F^2 - 2\mathrm{Tr}(\mathbf{Q}_j\mathbf{Q}_i^T\mathbf{P}_{ij})\right) = \\
&\underset{\mathcal{Q}\in\{S_{n_i\times m}\}^N}{\arg\min} \sum_{i,j=1}^{N} \left(k_{ij} - \mathrm{Tr}\big(\mathbf{Q}_i^T(2\mathbf{P}_{ij} - \mathbf{Q}_i\mathbf{Q}_j^T)\mathbf{Q}_j\big)\right) = \\
&\underset{\mathcal{Q}\in\{S_{n_i\times m}\}^N}{\arg\min} \sum_{i,j=1}^{N} k_{ij} - \sum_{i,j=1}^{N} \mathrm{Tr}\big(\mathbf{Q}_i^T(2\mathbf{P}_{ij} - \mathbf{Q}_i\mathbf{Q}_j^T)\mathbf{Q}_j\big),
\end{aligned} \tag{7.1}$$

where $\|\cdot\|_F$ is the Frobenius matrix norm and Tr is the linear trace operator. Our proposal consists to relax (7.1) in the final maximization problem

$$\underset{\mathcal{Q}\in\{S_{n_i\times m}\}^N}{\arg\max}\ E = \sum_{i,j=1}^{N} \mathrm{Tr}\big(\mathbf{Q}_i^T(2\mathbf{P}_{ij} - \mathbf{Q}_i\mathbf{Q}_j^T)\mathbf{Q}_j\big). \tag{7.2}$$

We decide to solve (7.2) by gradient descent, whose derivative of the energy E with respect to $\mathbf{Q}_k$ is the formula as follows:

$$\begin{aligned}
\frac{dE}{d\mathbf{Q}_k} &= 2\textstyle\sum_{j\neq k}^{N}\left(\mathbf{P}_{kj}\mathbf{Q}_j - \mathbf{Q}_k\mathbf{Q}_j^T\mathbf{Q}_j\right) + 2\sum_{i\neq k}^{N}\left(\mathbf{P}_{ik}^T\mathbf{Q}_i - \mathbf{Q}_k\mathbf{Q}_i^T\mathbf{Q}_i\right) + 4\mathbf{Q}_k - 4\mathbf{Q}_k\mathbf{Q}_k^T\mathbf{Q}_k \\
&= 2\textstyle\sum_{i=1}^{N}\left((\mathbf{P}_{ki} + \mathbf{P}_{ik}^T)\mathbf{Q}_i - 2\mathbf{Q}_k\mathbf{Q}_i^T\mathbf{Q}_i\right).
\end{aligned}$$

The relaxed formulation as above models the constraint $\mathbf{Q}_i\mathbf{1}_m = \mathbf{1}_{n_i}$ only, while it skips $\mathbf{Q}_i^T\mathbf{1}_{n_i} \leq \mathbf{1}_m$ which should be required $\forall i = 1, \ldots, N$. Although, once we have estimated the transitive cost matrix $\overline{\mathbf{P}}_{ij}$, we can produce the related partial permutation matrix by discretization with Hungarian algorithm [21].

7.4 Experimental Setup and Evaluation

7.4.1 Synthetic Experiments

In this section we introduce our test protocol to analyse and evaluate the performances of our method. Since we derive the transitive partial alignments from an initial source of solutions independently, we synchronized two well-known graph matching algorithms: Reweighted Random Walks (RRW) [30] and Graduated Assignment (GA) [57]. Moreover, we decided to compare our results with respect to another approach, the Consistency-driven Alternating Optimization (CDAO) [192], for two main reasons: first, it works similarly as our process by solving alignments from a common unknown reference ordering; second, this method is recognized being the state-of-the-art algorithm for multi-graph matching in literature. Although, CDAO does not model directly subgraph matching problems, so we ran such process by adding dummy nodes when is required for the execution only. In Table 7.1 we describe in deep the whole set of algorithms employed in our experiments in terms of parameters and implementation details.

We tested these algorithms on synthetic dataset adjusting classical protocols [30, 158, 191, 192] for multi-graph matching task. The dataset is created according the interval of nodes $[n_{MIN}, n_{MAX}]$ (with $n_{MAX} \geq n_{MIN}$), which controls the generation of the graphs in two different ways strategies:

1. **Weighted Graphs**. We generate a root graph G^r with n_{MAX} nodes of edge density ρ whose weights are uniformly distributed random numbers in $[0, 1]$. Each graph G^i is generated perturbing the edges of the root graph by adding Gaussian deformation $N(0, \sigma^2)$ just in the subgraph of n_{MIN} nodes (*i.e.*, edges between the inliers nodes). Finally, the final graph is cropped choosing its dimension by a random number of node such as $n_i \in [n_{MIN}, n_{MAX}]$. In other terms, each graph contains a different number $n_{MAX} - n_i$ of outlier nodes.[1]

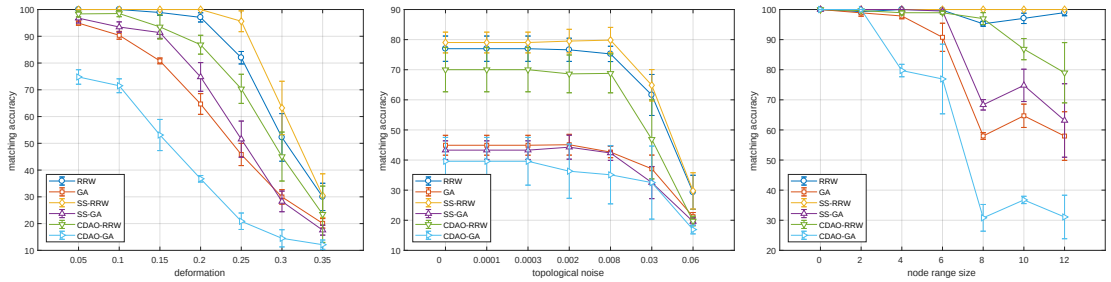
2. **Triangulated Graphs**. We control the topologies of the graph defining a root set of 2D points as $S_{G^r} = \{([0, 1]; [0, 1])\}_{k=1}^{n_{MAX}}$. The structure of each graph G^i is obtained by Delaunay triangulation perturbing the root set with additional random noise in $[0, 1]$. Finally, the graph is cropped to a random dimension in $[n_{MIN}, n_{MAX}]$.

[1]The generation of random graphs of different size gets difficult to control properly edge density and outlier nodes parameters, for this reason in this work we skipped experiments for their deep analysis. Generally, if there is not specified in the context, we fixed the default levels of edge density $\rho = 1$ and deformation $\sigma = 0.2$ in the presented tests.

Methods for Pairwise (Sub)graph Matching
GA. According the original work [57], initializing the parameter $\beta = \max(n_i, n_j)$ as the maximum number of nodes of two graphs. The initial point of the process is set as a rectangular $n_i \times n_j$ matrix of ones. The incrementation ratio for β is set to 1.075, while the related maximum value to 200.
RRW. According the original work [30], initializing the parameters $\alpha = 0.2$ and $\beta = 30$ and defining the initial point as the uniform rectangular $n_i \times n_j$ matrix of $1/(n_i n_j)$.
D-GA, **D-RRW** Refers respectively to original **GA** and **RRW**, but adding dummy nodes to the graphs of different size. More specifically, once we retrieved the maximum graph dimension in a given dataset, we even the remaining smaller graphs with the sufficient number of disconnected nodes.
Methods for Muti-(Sub)graph Matching
SS-GA, **SS-RRW**. We initialized the parameter $m = \max(n_1, \ldots, n_N)$ as the maximum number of nodes with respect to all the graphs. The initial pairwise solutions are derived from Graduated Assignment or Reweighted Random Walks techniques (without padding the graphs), which are related respectively as our **S**ynchronized **S**ub-graph methods **SS-GA** and **SS-RRW**. We implemented the non-linear problem (7.2) according the well-known Manopt optimization toolbox [20] to search the solution in the particular manifold of multi-simplex of m for each graphs. We set the maximum number of iteration at 500 and gradient norm with tolerance 10^{-14}.
CDAO-GA, **CDAO-RRW**. According the original work for the Non-Factorized version [192], initializing the max number of iteration $T_{max} = 2$ and using the Reweighted Random Walks [30] as internal solver for the QAP. We imported the external initial solution from Graduated Assignment or Reweighted Random Walks (padding the graphs), which are denoted respectively as **CDAO-GA** and **CDAO-RRW**.

Table 7.1: *Parameter settings and implementation details of the algorithms used for pairwise and multi-(sub)graph matching experiments.*

We evaluated the performances in terms of Matching Accuracy (MA) between two graphs (G^p, G^q), *i.e.*, the ratio of the correct matches with respect to the ground-truth X_{pq}^{TRU}, which is defined as the identity set $\{1, \ldots, \min(n_p, n_q)\}$ since our synthetic graphs are aligned from a common structural seed. Since the discretised solution for each pair $\overline{\mathbf{P}}_{pq}$ is defined as the partial $n_p \times n_q$ permutation matrix, if $n_p \geq n_q$ the node $1 \leq a \leq n_q$ matches with the node indexed in the position $1 \leq b \leq n_p$ of the unique 1 in the column $(\overline{\mathbf{P}}_{pq})_a$, otherwise if $n_p < n_q$ the matches is read $(\overline{\mathbf{P}}_{pq}^T)_b$. Considering X_{pq}^{ALG} the set of



(a) (b) (c)

Figure 7.2: *Average results with standard error for synthetic test at varying the levels of (a) deformation, (b) topological noise and (c) node range size performing all the methods in Table 7.1.*

matches retrieved in this way from some graph matching solver, we computed formally:

$$\mathrm{MA}(G^p, G^q) = \frac{|X_{pq}^{ALG} \cap X_{pq}^{TRU}|}{\min(n_p, n_q)}.$$

Finally, considering the whole set of N graphs $\mathcal{G} = \{G_1, \ldots, G_N\}$, we computed the overall measure with respect to all possible not trivial configurations as:

$$\mathrm{MA}(\mathcal{G}) = \frac{\sum_{p=1}^{N-1} \sum_{q=p+1}^{N} \mathrm{MA}(G^p, G^q)}{N(N-1)/2}.$$

In order to avoid dependence from a specific dataset we generated for each test 10 random graph datasets. Therefore, the final performance presented is the average of cumulative accuracy from all the trial datasets, from which we added a measure of standard error to visualize properly the boundaries of the mean values. Several algorithms as Graduated Assignment, Reweighted Random Walk and Consistency-driven Alternating Optimization require to operate on the $n_p n_q \times n_p n_q$ affinity matrix $\mathbf{W} = (w_{ia,jb})$, whose entry are solved by the rule

$$w_{ia,jb} = \exp\left(-\frac{(a_{ij}^p - a_{ab}^q)^2}{\sigma^2}\right), \tag{7.3}$$

where a_{ij}^p and a_{ab}^q are the related edge weights of the two graphs, and σ^2 the scaling parameter set to 0.15 experimentally. Moreover, the results are obtained sharing always the same affinity matrices to guarantee a faithful comparison of the algorithms.

In Figure 7.2 we presented the results from three experiments at varying the levels of (a) deformation, (b) topological noise, and (c) size of the range nodes $n_{MAX} - n_{MIN}$ with all the algorithms as introduced in Table 7.1. For experiments (a) and (b), the several random dataset are generated with a fixed interval of nodes $[15, 25]$. Moreover, the graphs are

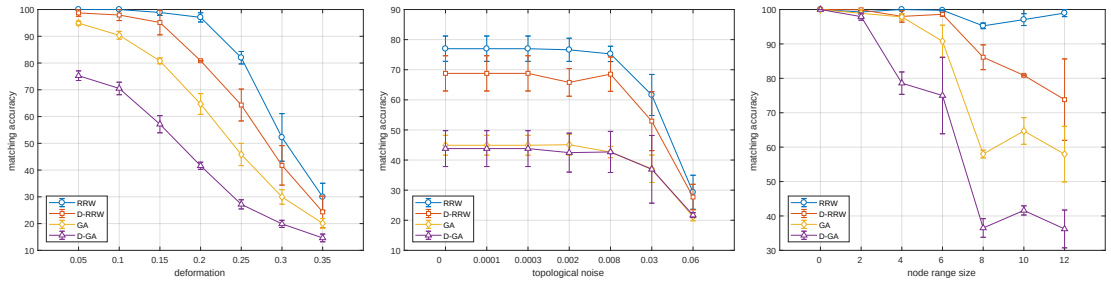
weighted in (a) and (c), while triangulated in (b). These selection of experiment is aimed to analyse the performances in the special scenario of subgraph matching problems with multiple data. We can observe several common aspects from all these experiments. In general, the baseline Reweighted Random Walks works better with respect to Graduated Assignment, whose weakness is propagated clearly in the depending algorithms as well. Our synchronized SS-GA and SS-RRW methods improve the accuracy with respect to the unsynchronized subgraph matching solutions of GA and RRW respectively. Then, we outperformed the performances with the state-of-the-art CDAO-RRW and CDAO-GA. Moreover, the latter tends to be the worst method in our experiments. We explain this behaviour very likely for the internal solver of CDAO since treats a QAP by RRW algorithm, which could be less suitable to synchronize GA-based solutions. Generally, our algorithms work properly in condition of perturbation of graphs (a,b) and increasing the variability of the number of nodes (c). Although, in experiment (b) our SS-GA algorithm seems to remain on the baseline GA. We could motivate the latter aspect guessing two reasons: first, the performances with GA are quite low generally, our synchronization has no sufficient information to improve further the accuracy; second, we are working with discrete triangulated graphs (with respect to real-valued representations), whose contribute in the topological discrimination may affect too crisp the weighted constrains in GA.

Finally, we conclude this evaluation just reasoning about the free parameter m in our formulation as section 7.3. We decided to set that as the maximum number of nodes of the graphs in a dataset, which is clearly the baseline condition otherwise some node matches could not be modelled. Regardless, it would be possible to set even greater value than such lower bound. Without adding trivial plots, we have tried different tests but the accuracy has never been affected. We expected such behaviour, since the process should not need to use further unconnected nodes beyond those would be required to fill all the graphs. This is in general a good aspect, since our method can operate with good performances without increasing excessively the dimension of the problem.

7.4.2 Dummy Nodes and Full Graph Matching

We are interested to investigate further about the impact of dummy nodes during the learning to solve subgraph matching. Even if we could skip directly the introduction of temporary nodes in certain graph matching methods, we forced such condition to measure how they involve on the general performances.

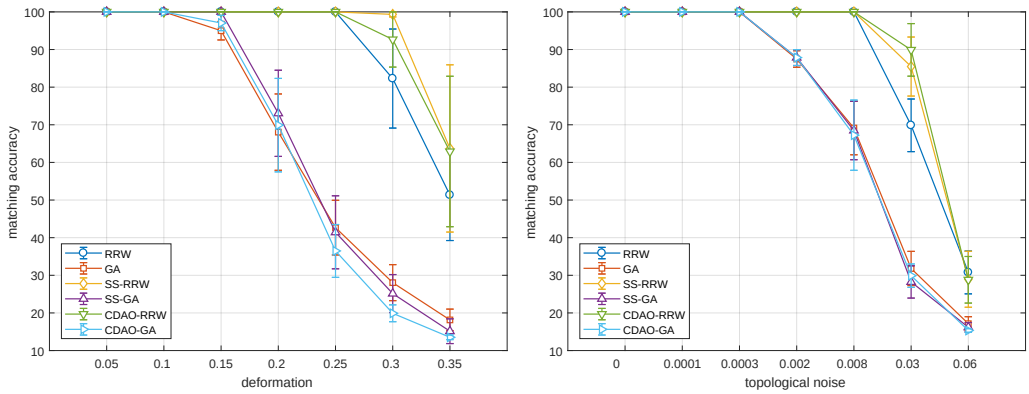
In Figure 7.3 we presented the behaviour to solve pairwise subgraph matching with classical RRW/GA algorithms and by adding dummy nodes as D-GA/D-RRW. The experiment setting (*i.e.*, the employed datasets) followed the main indications as in section 7.4.1. We can observe mainly a sharp decreasing of the general accuracy just in the performances with dummy nodes, therefore even if such nodes are disconnected with respect to the parent graphs, their impact on the learning is pretty relevant. Moreover, this aspect could explain better why the state-of-the-art CDAO in the previous experience as in Figure 7.2 is particularly weak with respect to our method. The former is forced by definition to pad the graphs before the learning, that is synchronizing a resulting initial



(a) (b) (c)

Figure 7.3: *Average results with standard error from synthetic data as Figure 7.2 at varying the levels of (a) deformation, (b) topological noise and (c) node range size performing subgraph matching with dummy nodes according methods as Table 7.1.*

solution which is just disadvantaged at the beginning.



(a) (b)

Figure 7.4: *Average results with standard error for synthetic test at varying the levels of (a) deformation and (b) topological noise with graphs of equal sized performing the graph matching methods as Table 7.1.*

The introduction of dummy nodes by another point of view consists just to solve full graph matching problem, since all graphs share the same number of nodes. We address to analysis such condition in general terms, in order to evaluate the behaviour of our method even in this special case. Figure 7.4 presents an experiment which is similar to Figure 7.2, but setting the condition $n_{MAX} = n_{MIN} = 30$, which is just an application of full graph matching. These new results confirm the conclusions presented in the experiments as section 7.4.1, wherein prevails the power of RRW algorithm working both in deformation test and topological noise experiments. Moreover, it is worth to be noted the fundamental trend that even by solving a typical task of full graph matching our method remains stati-

cally aligned with the state-of-the-art performances (which we remind it assumes graphs of equal sizes). This behaviour is pretty required, since we need to remain competitive with respect to the performances of classical graph matching solvers. This further experiment showed that our generalized formulation includes properly the discrimination accuracy to deal with full graph matching as well.

7.4.3 Constrain Violation Study

We are interested to analyse in deep how the violation of the constrain

$$\mathbf{Q}_i^T \mathbf{1}_{n_i} \leq \mathbf{1}_m \qquad \forall i = 1, \dots, N$$

in our formulation acts with respect to different conditions. Considering the i-th alignment as the $n_i \times m$ matrix $\mathbf{Q}_i$, we introduce the indicator function

$$\sigma_{\mathbf{Q}_i,k} = \begin{cases} 1 & (\mathbf{Q}_i)_k^T \mathbf{1}_{n_i} > 1 \\ 0 & \text{otherwise,} \end{cases}$$

which returns 1 if the k-th column in $\mathbf{Q}_i$ does not respect the constrain. We define a rough overall measure to compute the average Constrain Violation Ratio (CVR) considering all the columns and alignments as follows

$$\sigma_{\mathcal{Q}} = \frac{1}{m} \sum_{k=1}^{m} \left[\frac{1}{N} \sum_{i=1}^{N} \sigma_{\mathbf{Q}_i,k} \right] \in [0, 1].$$

Moreover, there may be interesting to measure the amount of violation in quantitative terms. We define the real-valued Average Constrain Violation Value (ACVV) measure as

$$\delta_{\mathcal{Q}} = \frac{1}{m} \sum_{k=1}^{m} \left[\frac{\sum_{i=1}^{N} \sigma_{\mathbf{Q}_i,k} \big((\mathbf{Q}_i)_k^T \mathbf{1}_{n_i} \big)}{\sum_{i=1}^{N} \sigma_{\mathbf{Q}_i,k}} \right] - 1 \in \mathbb{R}^+.$$

which computes the overall deviation value that exceeds the maximum of 1 in a column. Finally, we introduce a further measure to combine the ratio of violation event for the columns with respect to the magnitude of a given violation value $v > 1$. Considering the indicator function

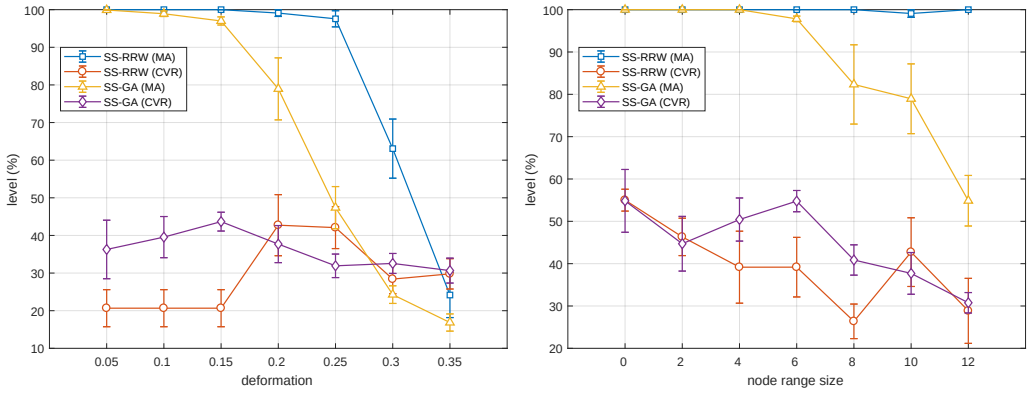
$$\sigma_{\mathbf{Q}_i,k,v} = \begin{cases} 1 & (\mathbf{Q}_i)_k^T \mathbf{1}_{n_i} \leq v \\ 0 & \text{otherwise} \end{cases}$$

we define the Constrain Violation Value Ratio (CVVR) as follows

$$\sigma_{\mathcal{Q},v} = \frac{1}{m} \sum_{k=1}^{m} \left[\frac{1}{N} \sum_{i=1}^{N} \sigma_{\mathbf{Q}_i,k,v} \right] \in [0, 1].$$

In Figure 7.5 we performed two tests with (a) deformation and (b) node range size for random graph datasets according the setting of the previous experiments as in Figure 7.2.

We added the related average measures of Constrain Violation Value (ACVV) in Table 7.2 and the Constrain Violation Value Ratio (CVVR) in Figure 7.6 for both experiments and methods. The common behaviour we can observe from this analysis consists that the constrain violation tends to remain quite low and without some tangible correlation with the decreasing in matching accuracy. The latter aspect is quite important since it means that supporting at least one constrain may be a sufficient condition to estimate correct matches as well. Indeed, the results in Table 7.2 highlight that the magnitude of the violation that happens during the learning is very low in average, considering that the possible upper bound for an alignment matrix is $n_i - 1$. After that, the overall picture of these tests in Figure 7.6 shows that the majority of column violations have values concentrated just beyond 1, in other terms, they are very small. Moreover, contrary to CVVR, we observe that the ACVV value tends to increase with deformation or node range size, *e.g.*, when matching accuracy decreases. This aspect may suggest that even if the ratio of the violation event is quite low, the weak accuracy may be further enhanced by the bias over the columns.



(a) Deformation Tests. (b) Node Range Size Tests.

Figure 7.5: *Average results of matching accuracy (MA) and constrain violation ratio (CVR) with standard error for synthetic test at varying the levels of (a) deformation and (b) node range size performing Syncronized Reweighted Random Walks and Graduated Assignment for multi-subgraph matching as Table 7.1.*

7.5 Conclusion

We have presented a novel generalization of the subgraph matching problem, which exploits of multiple graphs during the learning with arbitrary sizes. Our approach models the transitive partial permutation from combination of alignments which are defined according a common universe of node that embraces all the graphs. The proposed method

Table 7.2: *Average constrain violation value (ACVV) related to experiments of (a) deformation and (b) node range size presented in Figure 7.5.*

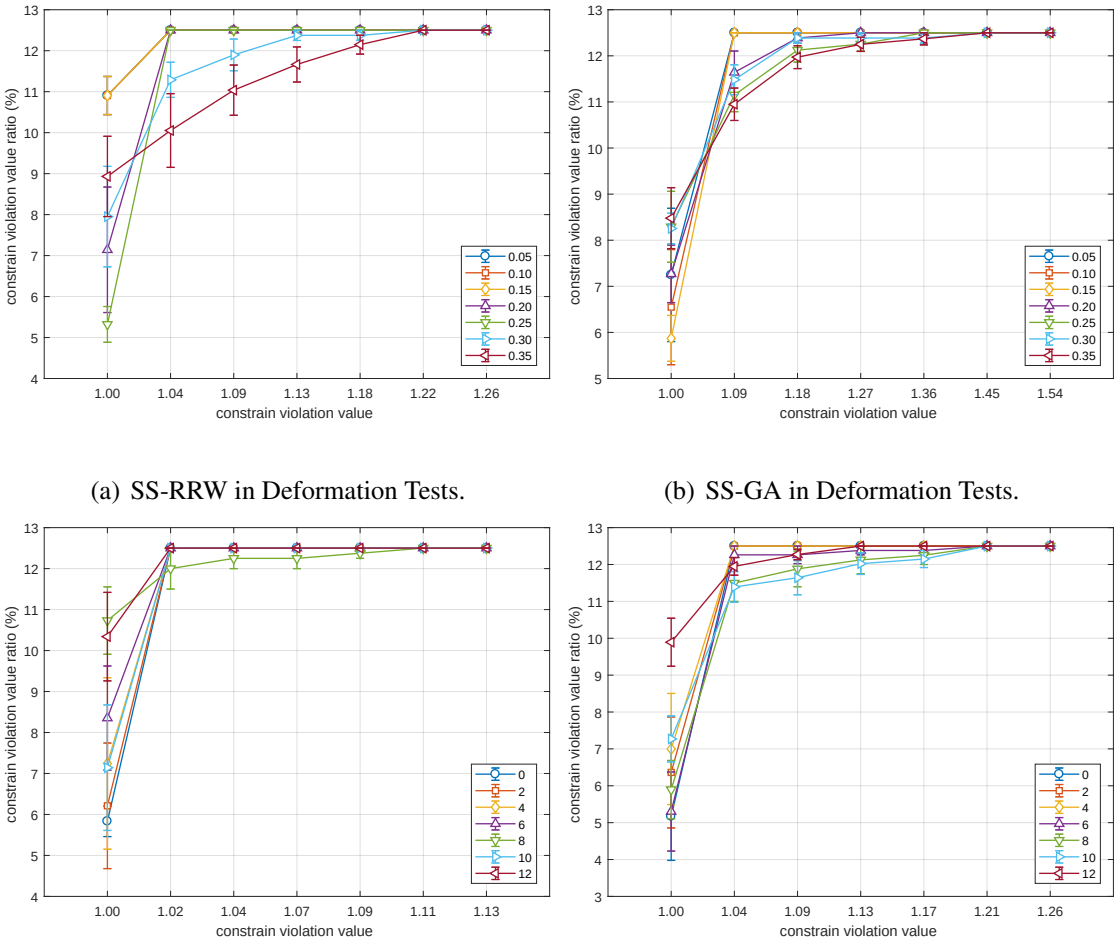
Test	SS-RRW	SS-GA
0.05	$5.0567 \cdot 10^{-11}$	0.0070605
0.10	$5.0567 \cdot 10^{-11}$	0.015083
0.15	$5.0567 \cdot 10^{-11}$	0.027939
0.20	0.0022064	0.056128
0.25	0.019011	0.075443
0.30	0.066212	0.078494
0.35	0.086735	0.08292

(a) Deformation Tests.

Test	SS-RRW	SS-GA
0	$8.0742 \cdot 10^{-11}$	$1.9842 \cdot 10^{-8}$
2	0.0014549	0.0006046
4	0.00031551	0.0080815
6	0.0009673	0.01879
8	0.0020278	0.044988
10	0.0022064	0.056128
12	0.0003098	0.06391

(b) Node Range Size Tests.

works as a off-line rectification process according an initial pairwise solution from external solvers. In our experiments we improved the matching accuracy of the unsynchronized methods based on Graduated Assignment and Reweighted Random Walks. Moreover, we even outperformed the performance of the state-of-the-art Consistency-driven Alternating Optimization (CDAO) algorithm in the application of multi-(sub)graph matching.



(a) SS-RRW in Deformation Tests.

(b) SS-GA in Deformation Tests.

(c) SS-RRW in Node Range Size Tests.

(d) SS-GA in Node Range Size Tests.

Figure 7.6: *Constrain Violation Value Ratio (CVVR) for each tests and methods presented in Figure 7.5.*

III

Conclusion and Future Work

8

Conclusion

In the first part of this thesis we gave an exhaustive introduction about data representations in Machine Learning, with particular focus to stress the implications that may arise operating on classical vector spaces with respect to further complex unstructured paradigms, such as data graphs or more general feature collections. We presented the basis of the well-known Kernel Methods, which suggest approaches to generalize a positive semidefinite kernel with universal objects, including some classical example of kernel functions. We presented foundations of Computer Vision, treating in detail the main characteristics of the pinhole camera model and related rigid or affine geometrical transformations. Moreover, we showed several well-known applications and problems related to feature extraction as well as method to reconstruct 3D scene from general set of 2D views. The last and fundamental topic was about Matching Problem, highlighting the specific distinction between Graph Matching with respect to Point Set Matching. We introduced for both approaches the classical pairwise formulation and principal related solutions from the literature. Furthermore, we presented some recent methods to generalize these matching problems with multiple data objects as well.

In the second part of this text we presented our fundamental contribution, describing four different proposals addressed to solve multi-way matching problems by synchronization of structural transformations.

In *Chapter 4* we introduced an approach which combines Multi-Graph Matching and Graph kernel method for classification task. This work consists in a framework which can be suitable paired with any external pairwise Graph Matching solver in out-of-the-box fashion. In other terms, our solution operates off-line with respect to the whole estimation of the synchronized permutations between graphs, hence resulting in a synchronization process which is an independent rectification step to a primal estimated solution. We devised our approach by a formulation based on the space of orthogonal permutation matrices and solving the transitive transformations according an underlying set of alignments related to an unknown reference order. The consequent integer quadratic program is relaxed by an iterative optimization in the unit sphere and performing a discretization step of the final estimated solution. Our graph kernel is inspired to the class of assignment graph kernel, since we exploit of the aligned permutations of the graphs to retrieve the similarity measurements. The fundamental constrain of transitivity which is supported in

our Multi-graph matching approach gets the novel kernel positive semidefinite as well. Our experimental results in several classification tasks on graph datasets showed that introducing the global consistency in the transformations the performances can increase significantly, in particular we overcame state-of-the-art accurateness with respect to the conventional pairwise Graph Matching solvers.

In *Chapter 5* we introduced a process for the estimation of synchronized homographic transformations through a set of dense planar views, which simultaneously learns a pixel-wise classifier to distinguish the regions which contain the planar surface. Our approach is inspired on typical bundle adjustment tasks in Computer Vision, since it consists in an independent refinement operation onto an external set of rough homographies (whose consistency is not guaranteed at the primal stage). The iterative multi-view optimization is governed by an energy cost function, which combines homographic transformations and the plane masks used by the classifier as cumulative contributions of sub pairwise problems. For each iteration, the homographies are rectified by affine transformation synchronization in closed-form and the classifier is trained by graph-cut segmentation separately. Our combination of methodologies yields in a suitable setup, since the whole process could be easily parallelizable in case of fully-connected view graph. Furthermore, the method operates densely over the views, therefore does not require feature extraction or calibration task to lead the optimization. Our experimental results proofed that the introduction of global consistency in the transitive homographies had a tangible impact to reduce the general error with respect to other sparse homographic estimation techniques, both through the plane classifier performances and qualitative/quantitative tests on the transformed images.

In *Chapter 6* we introduced a further work for Multi-Graph Matching which solved some drawbacks in our previous approach as *Chapter 4*. The fundamental innovation in this solution lies in the active synchronization strategy, which works inside an underlying Graph Matching technique totally. This aspect is crucial to overcome a typical limit in off-line rectification methods, which suffer to be excessively sensitive by the raw solution imported to initialize the synchronization, whose accurateness is related to the external solver which cannot be controlled actively. Moreover, our proposal does not operate in the orthogonal permutation space, but considers transformations defined as double-stochastic matrices, which are points in the well-known Birkhoff's Polytope. In order to maintain the transitive constraints by the permutations, we just reformulated the Birkhoff-Von Neumamnn theorem integrating alignments structures. Furthermore, we exploited of the Matching Pursuit paradigm to get feasible the high magnitude of the resulting cost function by defining a suitable approximation. Our work could be employed as a general component to extend potentially any pairwise Graph Matching algorithm based on double-stochastic permutation matrices to a Multi-Graph Matching approach. We realized three algorithms which integrate our method based on Graduated Assignment, Path Following and Reweighted Random Walks for Graph Matching. The experiments demonstrated the effectiveness of our formulations on several synthetic and real-world graph dataset, since we outperformed not only the performances with respect to the unsynchronized counterpart algorithms, but even the state of the art in Multi-Graph matching field.

Finally, in *Chapter 7* we introduced a method which generalizes the problem of (sub)graph matching considering multiple graphs with arbitrary dimensions. This work is particularly inspired to the previous approaches of (full) Multi-Graph Matching as *Chapter 4* and *Chapter 6*, respectively, for initializing the synchronization from an external pairwise solution and relaxing the transitive alignments as stochastic matrices (at least in one-way sense). However, this new approach overcomes the common limit of previous works by solving multi-subgraph matching without padding the graphs with artificial dummy nodes directly. Our generalization consists to formulate the transitive alignments through the introduction of a further common dimension of nodes, which is set in the multiple standard simplex in order to embrace the several magnitudes of any graph. Therefore, we can define constrains against this virtual universe, which are reduced to real dimensions once the partial permutation are reconstructed. We evaluated our method synchronizing the solutions from two well-known graph matching solvers such as Graduated Assignment and Reweighted Random Walks on synthetic datasets by controlling in several ways noise and variability of the graph size. The resulting performances of these experiments showed a tangible improvement in our generalization both with respect to the unsynchronized solutions and state-of-the-art technique for Multi-Graph matching.

8.1 Future Work

In this thesis we widely analysed through our four proposed methods that exploiting of further data in pairwise matching problems can be an effective strategy to reduce significantly the uncertainty in local correspondences; although, our investigation to generalize this task is not terminated yet.

The problem to guarantee global consistency of the transformations is in general quite difficult to treat, in particular with high dimensionality on data. Despite classical pairwise matching problems, to enforce transitivity during the learning and in particular for on-line processes, there is required the active maintenance of several data structures for each iteration, resulting critically expensive both in memory and computational time. Moreover, there exist several application contexts which involve huge data, especially with graph-based representations. Hence, there becomes reasonable to plan a future study in order to devise more scalable formulations of our synchronization processes, *e.g.*, proposing factorization schemes of the main structures. Nevertheless, we have already presented in sections 4.5 and 6.4.4 some very general dimensionality analyses of respectively off-line and on-line learning, which address to reconsider the importance of dataset magnitudes with respect to synchronization tasks. Indeed, further generalization on huge datasets could lead to saturate the accuracy, since estimating a big set of alignments which have to be also very discriminative there could be very hard to solve numerically. For sure, this conjecture deserves further investigation to be sharper confirmed.

Regardless the dimensions of the problem, there is required further study in order to understand what are the properly conditions whose synchronization is reasonable gener-

ally. After that, we need to establish reliable assumptions and preconditions to exploit of this technique, *e.g.*, number of the samples, structural dimensions of the objects involved, quality of the available data and context of application. Indeed, there may happen that synchronization generates itself disturb when the unsynchronized solution has already the highest quality which is possible for a given problem; in addition, since the principle of global consistency consists to spread information to reduce local ambiguities, if the latter is not sufficiently sharp or biased at the beginning, then the process could even downgrade some local accurate transformations. In our work we simulated these different conditions mainly with synthetic data, but trusting just on such incomplete experience of this topic may lead us to pose weak conclusions by considering applications in real-world scenarios.

Our optimization proposals depend strictly by various parameters to control the learning phase: some are expected by our formulations, but some else are imported by the integration of additional solvers in our works. We initialized these values following both the settings found in the original works and experimentally by our coarse observation of the optimization dynamics. The latter represents a weak point in our analysis, since the current parameters should be unsuitable in further scenarios which are still out of our experience. Therefore, we need to study in deep such manner with the purpose to formulate some general criterion to tune these parameters properly. Regardless, we have already tried further alternatives for method in section 6.4.4 without obtaining significant differences on the final performances. This suggest that the overall impact of some parameters may be negligible actually, in particular working on low scale applications.

The most recent methods presenting in this thesis are the Synchronized Reweighted Random Walks algorithm in section 6.4.2 and the Generalization for (sub)graph matching as Chapter 7. In the former work we just adjusted our general model for multi-graph matching task obtaining a process which registered very high results with respect to our earlier experience with Synchronized Graduated Assignment and Path Following. Even if we are quite sure about the effectiveness of our new proposal, the algorithm may be further analysed with an overall comparison with respect to any multi-graph matching methods. In the latter and final work we just presented a preliminary study of a possible generalization for the problem of multi-subgraph matching. We implemented the process by gradient descent in the manifold of multi-dimensional standard simplex, but another formulation may be possible relaxing the alignments in the unit sphere, *i.e.*, similarly to the projection in transitive space as in section 4.2. Moreover, our experiments are limited with small and synthetic graph datasets, but we need to analyse the method in real-world scenarios as well. Finally, we illustrated in deep that the constrain violation over the alignment columns seems to be an event which does not effect heavily the learning accurateness. Although, we are interested to fix this potential weak point by introducing a hinge loss term in the current formulation, which should tolerate better the lack of the second constrain.

Bibliography

[4] AHN, Y.-K., PARK, Y.-C., CHOI, K.-S., PARK, W.-C., SEO, H.-M., AND JUNG, K.-M. 3d spatial touch system based on time-of-flight camera. *WSEAS Trans. Info. Sci. and App. 6* (2009), 1433–1442.

[5] AIGER, D., MITRA, N. J., AND COHEN-OR, D. 4-points congruent sets for robust surface registration. *ACM Transactions on Graphics 27*, 3 (2008), #85, 1–10.

[6] ALMOHAMAD, H. A., AND DUFFUAA, S. O. A linear programming approach for the weighted graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell. 15*, 5 (May 1993), 522–525.

[7] ARROSPIDE, J., SALGADO, L., NIETO, M., AND MOHEDANO, R. Homography-based ground plane detection using a single on-board camera. *IET Intelligent Transport Systems 4*, 2 (June 2010), 149–160.

[8] BALLARD, D. H., AND BROWN, C. M. *Computer Vision*, 1st ed. Prentice Hall Professional Technical Reference, 1982.

[9] BATLLE, J., MOUADDIB, E. M., AND SALVI, J. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern Recognition 31*, 7 (1998), 963–982.

[10] BAY, H., ESS, A., TUYTELAARS, T., AND GOOL, L. J. V. Surf: Speeded-up robust features. *Computer Vision and Image Understanding 110*, 3 (2008), 346–359.

[11] BELLMAN, R. E. *Dynamic Programming*. Dover Publications, Incorporated, 2003.

[12] BELONGIE, S., MALIK, J., AND PUZICHA, J. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4 (Apr 2002), 509–522.

[13] BERNARD, F., THUNBERG, J., GEMMAR, P., HERTEL, F., HUSCH, A., AND GONCALVES, J. A solution for multi-alignment by transformation synchronisation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 2161–2169.

[14] BESL, P. J., AND MCKAY, N. D. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 2 (Feb. 1992), 239–256.

[15] BIASOTTI, S., MARINI, S., MORTARA, M., PATANÈ, G., SPAGNUOLO, M., AND FALCIDIENO, B. 3d shape matching through topological structures. In *DGCI* (2003), I. Nyström, G. S. di Baja, and S. Svensson, Eds., vol. 2886 of *LNCS*, Springer, pp. 194–203.

[16] BIRKHOFF, D. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucuman Revista , Serie A 5* (1946), 147–151.

[17] BORGWARDT, K., AND KRIEGEL, H. Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on* (2005), IEEE, pp. 8–pp.

[18] BORGWARDT, K. M., AND KRIEGEL, H.-P. Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM 2005)* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 74–81.

[19] BOSCH, A., ZISSERMAN, A., AND MUNOZ, X. Image classification using random forests and ferns. In *Proc. 11th IEEE International Conference on Computer Vision – ICCV '07.* (2007), pp. 1–8.

[20] BOUMAL, N., MISHRA, B., ABSIL, P.-A., AND SEPULCHRE, R. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research 15* (2014), 1455–1459.

[21] BOURGEOIS, F., AND LASSALLE, J.-C. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Commun. ACM 14*, 12 (Dec. 1971), 802–804.

[22] BRONSTEIN, A., BRONSTEIN, M., AND KIMMEL, R. *Numerical Geometry of Non-Rigid Shapes*, 1 ed. Springer Publishing Company, Incorporated, 2008.

[23] BROWN, D. C. Close-range camera calibration. *Photogrammetric Engineering 37*, 8 (1971), 855–866.

[24] BROWN, M., AND LOWE, D. G. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision 74*, 1 (2006), 59–73.

[25] CAELLI, T., AND KOSINOV, S. An eigenspace projection clustering method for inexact graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 4 (April 2004), 515–519.

[26] CAPEL, D. P. An effective bail-out test for ransac consensus scoring. In *BMVC* (2005).

[27] CHAUDHURY, K. N., KHOO, Y., SINGER, A., AND COWBURN, D. Global registration of multiple point clouds using semidefinite programming. *CoRR abs/1306.5226* (2013).

[28] CHEN, C.-S., HUNG, Y.-P., AND CHENG, J.-B. Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. *IEEE Trans. Pattern Anal. Mach. Intell. 21*, 11 (1999), 1229–1234.

[29] CHEN, Y., AND MEDIONI, G. Object modelling by registration of multiple range images. *Image and Vision Computing 10*, 3 (1992), 145 – 155. Range Image Understanding.

[30] CHO, M., LEE, J., AND LEE, K. Reweighted random walks for graph matching. In *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6315 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2010, pp. 492–505.

[31] CHUI, H., AND RANGARAJAN, A. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding 89*, 2 (2003), 114 – 141. Nonrigid Image Registration.

[32] CHUM, O., AND MATAS, J. Matching with prosac " progressive sample consensus. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01* (Washington, DC, USA, 2005), CVPR '05, IEEE Computer Society, pp. 220–226.

[33] CHUM, O., MATAS, J., AND KITTLER, J. *Locally Optimized RANSAC*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 236–243.

[34] CHUNG, D. H., YUN, I. D., AND LEE, S. U. Registration of multiple-range views using the reverse-calibration technique. *Pattern Recognition 31*, 4 (1998), 457–464.

[35] CONTE, D., FOGGIA, P., SANSONE, C., AND VENTO, M. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence 18*, 03 (2004), 265–298.

[36] CONTE, D., FOGGIA, P., SANSONE, C., AND VENTO, M. Thirty years of graph matching in pattern recognition. *IJPRAI 18*, 3 (2004), 265–298.

[37] COOTES, T. F., AND TAYLOR, C. J. *Active Shape Models — 'Smart Snakes'*. Springer London, London, 1992, pp. 266–275.

[38] CORDELLA, L. P., FOGGIA, P., SANSONE, C., AND VENTO, M. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 10 (Oct 2004), 1367–1372.

[39] COUR, T., SRINIVASAN, P., AND SHI, J. Balanced graph matching. In *Advances in Neural Information Processing Systems 19*. MIT Press, 2007, pp. 313–320.

[40] DEBNATH, A. K., DE COMPADRE, R. L. L., DEBNATH, G., SHUSTERMAN, A. J., AND HANSCH, C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry 34*, 2 (1991), 786–797.

[41] EGGERT, D., LORUSSO, A., AND FISHER, R. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications 9*, 5 (Mar 1997), 272–290.

[42] EPPSTEIN, D. Subgraph isomorphism in planar graphs and related problems. *CoRR cs.DS/9911003* (1999).

[43] FAUGERAS, O. D., AND TOSCANI, G. The calibration problem for stereo. In *Proceedings, CVPR '86 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, June 22–26, 1986)* (1986), IEEE, pp. 15–20.

[44] FISCHLER, M. A., AND BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (June 1981), 381–395.

[45] FOGGIA, P., PERCANNELLA, G., AND VENTO, M. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence 28*, 01 (2014), 1450001.

[46] FRANK, M., AND WOLFE, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly 3*, 1-2 (1956), 95–110.

[47] FRÖHLICH, H., WEGNER, J. K., SIEKER, F., AND ZELL, A. Optimal assignment kernels for attributed molecular graphs. In *ICML* (2005), pp. 225–232.

[48] FRÖHLICH, H., WEGNER, J. K., SIEKER, F., AND ZELL, A. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)* (Bonn, Germany, August 2005), L. de Raedt and S. Wrobel, Eds., ACM Press, pp. 225–232.

[49] GAERTNER, T., FLACH, P., AND WROBEL, S. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop* (August 2003), Springer-Verlag, pp. 129–143.

[50] GAO, X., XIAO, B., TAO, D., AND LI, X. A survey of graph edit distance. *Pattern Analysis and Applications 13*, 1 (Feb 2010), 113–129.

[51] GAREY, M. R., AND JOHNSON, D. S. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[52] GASPARETTO, A., MINELLO, G., AND TORSELLO, A. A non-parametric spectral model for graph classification. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods* (2015), pp. 312–319.

[53] GAVRIL, F. Generating the maximum spanning trees of a weighted graph. *J. Algorithms 8*, 4 (Dec. 1987), 592–597.

[54] GHAHRAMAN, D. E., WONG, A. K. C., AND AU, T. Graph optimal monomorphism algorithms. *IEEE Transactions on Systems, Man, and Cybernetics 10*, 4 (April 1980), 181–188.

[55] GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H., AND SEITZ, S. M. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision* (Oct 2007), pp. 1–8.

[56] GOLD, S., AND RANGARAJAN, A. A graduated assignment algorithm for graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 18*, 4 (Apr 1996), 377–388.

[57] GOLD, S., AND RANGARAJAN, A. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell. 18*, 4 (Apr. 1996), 377–388.

[58] GOLD, S., RANGARAJAN, A., LU, C.-P., PAPPU, S., AND MJOLSNESS, E. New algorithms for 2d and 3d point matching. *Pattern Recognition 31*, 8 (1998), 1019 – 1031.

[59] GOWER, J. C., AND DIJKSTERHUIS, G. B. *Procrustes Problems*. Oxford University Press, Oxford, 2004.

[60] GRANGER, S., PENNEC, X., AND ROCHE, A. Rigid point-surface registration using an em variant of icp for computer guided oral implantology. In *MICCAI* (London, UK, 2001), Springer-Verlag, pp. 752–761.

[61] HALL, E. L., TIO, J. B. K., MCPHERSON, C. A., AND SADJADI, F. A. Measuring curved surfaces for robot vision. *Computer 15*, 12 (1982), 42–54.

[62] HARRIS, C., AND STEPHENS, M. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conference* (1988), pp. 147–151.

[63] HARTLEY, R. I. In defence of the 8-point algorithm. In *Proceedings of IEEE International Conference on Computer Vision* (1995), IEEE Comput. Soc. Press, pp. 1064–1070.

[64] HARTLEY, R. I., TRUMPF, J., DAI, Y., AND LI, H. Rotation averaging. *International Journal of Computer Vision 103*, 3 (2013), 267–305.

[65] HASHIM MIR, PETER XU, P. V. B. An extensive empirical evaluation of focus measures for digital photography, 2014.

[66] HAUSSLER, D. Convolution kernels on discrete structures. Tech. rep., Technical report, UC Santa Cruz, 1999.

[67] HAUSSLER, D. Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, University of California at Santa Cruz, Santa Cruz, CA, USA, 1999.

[68] HEIKKILÄ, J. Geometric camera calibration using circular control points. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 10 (2000), 1066–1077.

[69] HOCHREITER, S., AND OBERMAYER, K. Support vector machines for dyadic data. *Neural Comput. 18*, 6 (June 2006), 1472–1510.

[70] HORN, B. K. P. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A 4*, 4 (Apr 1987), 629–642.

[71] HOWE, N. R., AND DESCHAMPS, A. Better foreground segmentation through graph cuts. *CoRR cs.CV/0401017* (2004).

[72] ITO, T., CHIBA, T., OZAWA, R., YOSHIDA, M., HATTORI, M., AND SAKAKI, Y. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences 98*, 8 (2001), 4569.

[73] JAIN, B. J., AND GEIBEL, WYSOTZKI, F. Svm learning with the sh inner product. In *12th European Symposium on Artificial Neural Networks* (Bruges, Belgium,).

[74] JAIN, B. J., GEIBEL, P., AND WYSOTZKI, F. Svm learning with the schur-hadamard inner product for graphs. *Neurocomput. 64* (Mar. 2005), 93–105.

[75] JENSEN, L. J., KUHN, M., STARK, M., CHAFFRON, S., CREEVEY, C., MULLER, J., DOERKS, T., JULIEN, P., ROTH, A., SIMONOVIC, M., ET AL. String 8–a global view on proteins and their functional interactions in 630 organisms. *Nucleic acids research 37*, suppl 1 (2009), D412–D416.

[76] JEONG, H., TOMBOR, B., ALBERT, R., OLTVAI, Z., AND BARABÁSI, A. The large-scale organization of metabolic networks. *Nature 407*, 6804 (2000), 651–654.

[77] KAICK, O. V., ZHANG, H., HAMARNEH, G., AND COHEN-OR, D. A Survey on Shape Correspondence. *Computer Graphics Forum* (2011).

[78] KALAPALA, V., SANWALANI, V., AND MOORE, C. The structure of the united states road network. *Preprint, University of New Mexico* (2003).

[79] KASHIMA, H., TSUDA, K., AND INOKUCHI, A. Marginalized kernels between labeled graphs. In *ICML* (2003), pp. 321–328.

[80] KASHIMA, H., TSUDA, K., AND INOKUCHI, A. Marginalized kernels between labeled graphs. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (2003), T. Fawcett and N. Mishra, Eds., pp. 321–328.

[81] KE, Y., AND SUKTHANKAR, R. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition – CVPR '04* (2004), vol. 2, pp. 506–513.

[82] KOOPMANS, T. C., AND BECKMANN, M. Assignment problems and the location of economic activities. *Econometrica 25*, 1 (1957), 53–76.

[83] KRIEGE, N., AND MUTZEL, P. Subgraph matching kernels for attributed graphs. In *ICML* (2012), icml.cc / Omnipress.

[84] KRISHNAN, S., LEE, P. Y., MOORE, J. B., AND VENKATASUBRAMANIAN, S. Global registration of multiple 3d point sets via optimization-on-a-manifold. In *Proceedings of the Third Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2005), SGP '05, Eurographics Association.

[85] KUHN, H. W. *Lectures on the Theory of Games*. Princeton University Press, 2003.

[86] KUMAR, M. P., KUTHIRUMMAL, S., JAWAHAR, C. V., AND NARAYANAN, P. J. Planar homography from fourier domain representation. In *Signal Processing and Communications, 2004. SPCOM '04. 2004 International Conference on* (Dec 2004), pp. 560–564.

[87] KUMAR, R. M., AND K, S. A survey on image feature descriptors. *International Journal of Computer Science and Information Technologies 5*, 6 (2014), 7668–7673.

[88] LAZEBNIK, S., SCHMID, C., AND PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2* (Washington, DC, USA, 2006), CVPR '06, IEEE Computer Society, pp. 2169–2178.

[89] LEORDEANU, M., AND HEBERT, M. A spectral technique for correspondence problems using pairwise constraints. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Oct 2005), vol. 2, pp. 1482–1489 Vol. 2.

[90] LEORDEANU, M., HEBERT, M., AND SUKTHANKAR, R. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), pp. 1–8.

[91] LI, G., SEMERCI, M., YENER, B., AND ZAKI, M. J. Effective graph classification based on topological and label attributes. *Statistical Analysis and Data Mining 5*, 4 (2012), 265–283.

[92] LIEBOWITZ, D., AND ZISSERMAN, A. Metric rectification for perspective images of planes. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on* (Jun 1998), pp. 482–488.

[93] LILIENBLUM, E., AND MICHAELIS, B. Optical 3d surface reconstruction by a multi-period phase shift method. *JCP 2*, 2 (2007), 73–83.

[94] LINDEBERG, T. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision 11*, 3 (Dec 1993), 283–318.

[95] LIPING ZHENG, GUANGYAO LI, J. S. The survey of medical image 3d reconstruction, 2007.

[96] LIU, Y. Replicator dynamics in the iterative process for accurate range image matching. *Int. J. Comput. Vision 83*, 1 (2009), 30–56.

[97] LIU, Z.-Y., QIAO, H., AND XU, L. An extended path following algorithm for graph-matching problem. *IEEE Trans. Pattern Anal. Mach. Intell. 34*, 7 (July 2012), 1451–1456.

[98] LIVI, L., AND RIZZI, A. The graph matching problem. *Pattern Analysis and Applications 16*, 3, 253–283.

[99] LOIOLA, E. M., DE ABREU, N. M. M., BOAVENTURA-NETTO, P. O., HAHN, P., AND QUERIDO, T. A survey for the quadratic assignment problem. *European Journal of Operational Research 176*, 2 (2007), 657 – 690.

[100] LONGUET-HIGGINS, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature 293*, 5828 (Sep 1981), 133–135.

[101] LOWE, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2 (Nov. 2004), 91–110.

[102] LUCCHESE, L., AND MITRA, S. K. Using saddle points for subpixel feature detection in camera calibration targets. In *APCCAS* (2002), IEEE, pp. 191–195.

[103] LUO, B., WILSON, R. C., AND HANCOCK, E. R. Spectral embedding of graphs. *Pattern Recognition 36*, 10 (2003), 2213 – 2230.

[104] LUONG, Q.-T., AND FAUGERAS, O. D. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision 17*, 1 (Jan 1996), 43–75.

[105] MAHÉ, P., UEDA, N., AKUTSU, T., PERRET, J.-L., AND VERT, J.-P. Extensions of marginalized graph kernels. In *Proceedings of the twenty-first international conference on Machine learning* (2004), ACM, p. 70.

[106] MARR, D., AND HILDRETH, E. Theory of edge detection. *Royal Soc. of London Proc. Series B 207* (1980), 187–217.

[107] MATAS, J., AND CHUM, O. Randomized ransac with td,d test. *Image and Vision Computing 22*, 10 (2004), 837 – 842. British Machine Vision Computing 2002.

[108] MATAS, J., AND CHUM, O. Randomized ransac with sequential probability ratio test. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (Oct 2005), vol. 2, pp. 1727–1732 Vol. 2.

[109] MATAS, J., CHUM, O., URBAN, M., AND PAJDLA, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing 22*, 10 (2004), 761–767. British Machine Vision Computing 2002.

[110] MCLAUCHLAN, P., AND JAENICKE, A. Image mosaicing using sequential bundle adjustments. In *Proc. BMVC* (2000), pp. 62.1–62.10. doi:10.5244/C.14.62.

[111] MENCHETTI, S., COSTA, F., AND FRASCONI, P. Weighted decomposition kernels. In *Proceedings of the 22Nd International Conference on Machine Learning* (New York, NY, USA, 2005), ICML '05, ACM, pp. 585–592.

[112] MIAN, A. S., BENNAMOUN, M., AND OWENS, R. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Trans. Pattern Anal. Mach. Intell. 28* (October 2006), 1584–1601.

[113] MIKOLAJCZYK, K., AND SCHMID, C. An affine invariant interest point detector. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I* (London, UK, 2002), Springer-Verlag, pp. 128–142.

[114] MIKOLAJCZYK, K., AND SCHMID, C. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 27*, 10 (2005), 1615–1630.

[115] MOHR, J., JAIN, B. J., SUTTER, A., TER LAAK, A., STEGER-HARTMANN, T., HEINRICH, N., AND OBERMAYER, K. A maximum common subgraph kernel method for predicting the chromosome aberration test. *Journal of Chemical Information and Modeling* (2010), 1821–1838.

[116] MOREL, J.-M., AND YU, G. ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci. 2*, 2 (2009), 438–469.

[117] NENE, S. A., NAYAR, S. K., AND MURASE, H. Columbia object image library (coil-20). *Dept. Comput. Sci., Columbia Univ., New York.[Online] http://www. cs. columbia. edu/CAVE/coil-20. html 62* (1996).

[118] NEUHAUS, M., AND BUNKE, H. Edit distance-based kernel functions for structural pattern classification. *Pattern Recognition 39*, 10 (2006), 1852–1863.

[119] NEUMANN, J. V. Allgemeine eigenwerttheorie hermitescher funktionaloperatoren. *Mathematische Annalen 102* (1930), 49–131.

[120] NEUMANN, M., PATRICIA, N., GARNETT, R., AND KERSTING, K. Efficient graph kernels by randomization. In *ECML/PKDD (1)* (2012), P. A. Flach, T. D. Bie, and N. Cristianini, Eds., vol. 7523 of *Lecture Notes in Computer Science*, Springer, pp. 378–393.

[121] OLIVA, A., AND TORRALBA, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision 42*, 3 (May 2001), 145–175.

[122] ONG, C. S., CANU, S., AND SMOLA, A. J. Learning with non-positive kernels. In *In Proc. of the 21st International Conference on Machine Learning (ICML)* (2004), pp. 639–646.

[123] PACHAURI, D., KONDOR, R., AND SINGH, V. Solving the multi-way matching problem by permutation synchronization. In *Adv. NIPS 2013* (2013), pp. 1860–1868.

[124] PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference* (Brisbane, Australia, 1998), pp. 161–172.

[125] PARK, S. Y., AND PARK, G. G. Active calibration of camera-projector systems based on planar homography. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (Aug 2010), pp. 320–323.

[126] PARRILO, P. A. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming 96*, 2 (May 2003), 293–320.

[127] PELILLO, M. Replicator equations, maximal cliques, and graph isomorphism. *Neural Comput. 11*, 8 (Nov. 1999), 1933–1955.

[128] PEVZNER, P. A. Multiple alignment, communication cost, and graph matching. *SIAM Journal on Applied Mathematics 52*, 6 (1992), 1763–1779.

[129] PIZARRO, D., AND BARTOLI, A. Global optimization for optimal generalized procrustes analysis. In *CVPR 2011* (June 2011), pp. 2409–2415.

[130] PRINCE, S. J. D., XU, K., AND CHEOK, A. D. Augmented reality camera tracking with homographies. *IEEE Computer Graphics and Applications 22*, 6 (Nov 2002), 39–45.

[131] RAMON, J., AND GÄRTNER, T. Expressivity versus efficiency of graph kernels. In *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences* (2003), pp. 65–74.

[132] REMONDINO, F., AND FRASER, C. Digital camera calibration methods: considerations and comparisons. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* (Dresden, Germany, 2006), Isprs, Ed., vol. Vol. XXXVI.

[133] ROSENFELD, A., HUMMEL, R. A., AND ZUCKER, S. Scene labeling by relaxation operations. *Systems, Man and Cybernetics, IEEE Transactions on SMC-6*, 6 (1976), 420–433.

[134] RUDEMO, M. Statistical shape analysis. i. l. dryden and k. v. mardia, wiley, chichester 1998. no. of pages: xvii+347. price: £60.00.isbn 0-471-95816-6. *Statistics in Medicine 19*, 19 (2000), 2716–2717.

[135] RUSINKIEWICZ, S., AND LEVOY, M. Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling* (2001), pp. 145–152.

[136] RUTA, D., AND GABRYS, B. An overview of classifier fusion methods. *Computing and Information Systems 7*, 1 (February 2000), 1–10.

[137] SAEZ, J. M., AND ESCOLANO, F. 6dof entropy minimization slam. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* (May 2006), pp. 1548–1555.

[138] SALAHAT, E., AND QASAIMEH, M. Recent advances in features extraction and description algorithms: A comprehensive survey. *CoRR abs/1703.06376* (2017).

[139] SALVI, J., ARMANGUE, X., AND BATLLE, J. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition 35*, 7 (2002), 1617–1635.

[140] SALVI, J., MATABOSCH, C., FOFI, D., AND FOREST, J. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput. 25*, 5 (2007), 578–596.

[141] SALVI, J., PAGÈS, J., AND BATLLE, J. Pattern codification strategies in structured light systems. *Pattern Recognition 37* (2004), 827–849.

[142] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM JOURNAL OF RESEARCH AND DEVELOPMENT 3*, 3 (July 1959), 210–229.

[143] SCHARSTEIN, D., AND SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithm. *Int. J. of Comp. Vision 47*, 1 (2002), 7–42.

[144] SCHIETGAT, L., RAMON, J., BRUYNOOGHE, M., AND BLOCKEEL, H. An efficiently computable graph-based metric for the classification of small molecules. In *Proceedings of the 11th International Conference on Discovery Science* (Berlin, Heidelberg, 2008), DS '08, Springer-Verlag, pp. 197–209.

[145] SCHOLKOPF, B., AND SMOLA, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[146] SCHOMBURG, I., CHANG, A., EBELING, C., GREMSE, M., HELDT, C., HUHN, G., AND SCHOMBURG, D. Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Research 32*, suppl 1 (2004), D431–D433.

[147] SCOTT, G. L., AND LONGUET-HIGGINS, H. C. An algorithm for associating the features of two images. *Proceedings of the Royal Society of London B: Biological Sciences 244*, 1309 (1991), 21–26.

[148] SHAPIRO, L. G., AND HARALICK, R. M. Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-3*, 5 (Sept 1981), 504–519.

[149] SHAPIRO, L. S., AND BRADY, J. M. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing 10*, 5 (1992), 283 – 288. BMVC 1991.

[150] SHEN, D., AND DAVATZIKOS, C. Hammer: hierarchical attribute matching mechanism for elastic registration. *IEEE Transactions on Medical Imaging 21*, 11 (Nov 2002), 1421–1439.

[151] SHERVASHIDZE, N., SCHWEITZER, P., VAN LEEUWEN, E. J., MEHLHORN, K., AND BORGWARDT, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research 12* (2011), 2539–2561.

[152] SHERVASHIDZE, N., SCHWEITZER, P., VAN LEEUWEN, E. J., MEHLHORN, K., AND BORGWARDT, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research 12* (2011), 2539–2561.

[153] SHERVASHIDZE, N., VISHWANATHAN, S., PETRI, T., MEHLHORN, K., AND BORGWARDT, K. Efficient graphlet kernels for large graph comparison. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics* (2009).

[154] SIDDIQI, K., SHOKOUFANDEH, A., DICKINSON, S., AND ZUCKER, S. Shock graphs and shape matching. *International Journal of Computer Vision 35*, 1 (1999), 13–32.

[155] SIMON, G., FITZGIBBON, A. W., AND ZISSERMAN, A. Markerless tracking using planar structures in the scene. In *Augmented Reality, 2000. (ISAR 2000). Proceedings. IEEE and ACM International Symposium on* (2000).

[156] SINKHORN, R. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics 35*, 2 (1964), 876–879.

[157] SOLÉ-RIBALTA, A., AND SERRATOSA, F. Graduated assignment algorithm for multiple graph matching based on a common labeling. *International Journal of Pattern Recognition and Artificial Intelligence 27*, 01 (2013), 1350001.

[158] SOLÉ-RIBALTA, A., AND SERRATOSA, F. Models and algorithms for computing the common labelling of a set of attributed graphs. *Computer Vision and Image Understanding 115*, 7 (2011), 929–945.

[159] SRINIVASAN, V., LIU, H. C., AND HALIOUA, M. Automated phase-measuring profilometry: a phase mapping approach. *Appl. Opt. 24*, 2 (1985), 185–188.

[160] STEWART, J. Positive definite functions and generalizations, an historical survey. *Rocky Mountain J. Math. 6*, 3 (09 1976), 409–434.

[161] STRECHA, C., VON HANSEN, W., GOOL, L. V., FUA, P., AND THOENNESSEN, U. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008), pp. 1–8.

[162] SUN, J., OVSJANIKOV, M., AND GUIBAS, L. A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP '09, Eurographics Association, pp. 1383–1392.

[163] SURREL, Y. Design of algorithms for phase measurements by the use of phase stepping. *Appl. Opt. 35*, 1 (1996), 51–60.

[164] SUTHERLAND, I. E. Three-dimensional data input by tablet. *Proceedings of the IEEE 62*, 4 (April 1974), 453–461.

[165] SZELISKI, R. *Computer Vision: Algorithms and Applications*, 1st ed. Springer-Verlag New York, Inc., New York, NY, USA, 2010.

[166] SZPAK, Z. L., CHOJNACKI, W., AND VAN DEN HENGEL, A. Robust multiple homography estimation: An ill-solved problem. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 2132–2141.

[167] TAKEDA, M., AND MUTOH, K. Fourier transform profilometry for the automatic measurement of 3-d object shapes. *Appl. Opt. 22*, 24 (1983), 3977–3982.

[168] TAREL, J.-P., CIVI, H., AND COOPER, D. B. Pose estimation of free-form 3d objects without point matching using algebraic surface models. In *Proceedings of IEEE Workshop Model Based 3D Image Analysis* (Mumbai, India, 1998), pp. 13–21.

[169] TORDOFF, B., AND MURRAY, D. W. *Guided Sampling and Consensus for Motion Estimation*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 82–96.

[170] TORR, P., AND ZISSERMAN, A. Robust computation and parametrization of multiple view relations. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision* (Washington, DC, USA, 1998), IEEE Computer Society, p. 727.

[171] TORR, P. H. S., AND ZISSERMAN, A. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding 78* (2000), 138–156.

[172] TORSELLO, A. An importance sampling approach to learning structural representations of shape. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA* (2008), IEEE Computer Society.

[173] TORSELLO, A., GASPARETTO, A., ROSSI, L., BAI, L., AND HANCOCK, E. Transitive state alignment for the quantum jensen-shannon kernel. In *Structural, Syntactic, and Statistical Pattern Recognition*, P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo, Eds., vol. 8621 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2014, pp. 22–31.

[174] TORSELLO, A., RODOLÀ, E., AND ALBARELLI, A. Multiview registration via graph diffusion of dual quaternions. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011* (2011), IEEE Computer Society, pp. 2441–2448.

[175] TRIGGS, B., MCLAUCHLAN, P., HARTLEY, R., AND FITZGIBBON, A. Bundle adjustment – a modern synthesis. In *"Vision Algorithms: Theory and Practice* (2000), B. Triggs, A. Zisserman, and R. Szeliski, Eds., vol. 1883 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 298–372.

[176] TRIGGS, B., MCLAUCHLAN, P. F., HARTLEY, R. I., AND FITZGIBBON, A. W. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice* (London, UK, UK, 2000), ICCV '99, Springer-Verlag, pp. 298–372.

[177] TSIN, Y., AND KANADE, T. *A Correlation-Based Approach to Robust Point Set Registration*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 558–569.

[178] UESHIBA, T., AND TOMITA, F. Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (Oct 2003), pp. 966–973 vol.2.

[179] ULLMANN, J. R. An algorithm for subgraph isomorphism. *J. ACM 23*, 1 (Jan. 1976), 31–42.

[180] UMEYAMA, S. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 5 (Sep 1988), 695–703.

[181] VERT, J.-P. The optimal assignment kernel is not positive definite. *CoRR abs/0801.4061* (2008).

[182] VINCENT, E., AND LAGANIERE, R. Detecting planar homographies in an image pair. In *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.* (2001), pp. 182–187.

[183] VISHWANATHAN, S. V. N., BORGWARDT, K. M., AND SCHRAUDOLPH, N. N. Fast computation of graph kernels. In *Proceedings of the 19th International Conference on Neural Information Processing Systems* (Cambridge, MA, USA, 2006), NIPS'06, MIT Press, pp. 1449–1456.

[184] WEN, G., ZHU, D., XIA, S., AND WANG, Z. Total least squares fitting of point sets in m-d. In *Proceedings of the Computer Graphics International 2005* (Washington, DC, USA, 2005), CGI '05, IEEE Computer Society, pp. 82–86.

[185] WIENER, H. Structural determination of paraffin boiling points. *Journal of the American Chemical Society 69*, 1 (1947), 17–20.

[186] WILLIAMS, M. L., WILSON, R. C., AND HANCOCK, E. R. Multiple graph matching with bayesian inference. *Pattern Recognition Letters 18* (1997), 080.

[187] WILSON, R. C., HANCOCK, E. R., AND LUO, B. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 7 (July 2005), 1112–1124.

[188] WU, C. Towards linear-time incremental structure from motion. In *Proceedings of the 2013 International Conference on 3D Vision* (Washington, DC, USA, 2013), 3DV '13, IEEE Computer Society, pp. 127–134.

[189] YAN, J., CHO, M., ZHA, H., YANG, X., AND CHU, S. M. A general multi-graph matching approach via graduated consistency-regularized boosting. *CoRR abs/1502.05840* (2015).

[190] YAN, J., LI, Y., LIU, W., ZHA, H., YANG, X., AND CHU, S. M. *Graduated Consistency-Regularized Optimization for Multi-graph Matching*. Springer International Publishing, Cham, 2014, pp. 407–422.

[191] YAN, J., TIAN, Y., ZHA, H., YANG, X., ZHANG, Y., AND CHU, S. M. Joint optimization for consistent multiple graph matching. In *Proceedings of the 2013 IEEE International Conference on Computer Vision* (Washington, DC, USA, 2013), ICCV '13, IEEE Computer Society, pp. 1649–1656.

[192] YAN, J., WANG, J., ZHA, H., YANG, X., AND CHU, S. Consistency-driven alternating optimization for multigraph matching: A unified approach. *Image Processing, IEEE Transactions on 24*, 3 (March 2015), 994–1009.

[193] ZASLAVSKIY, M., BACH, F., AND VERT, J.-P. *A Path Following Algorithm for Graph Matching*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 329–337.

[194] ZEINIK-MANOR, L., AND IRANI, M. Multiview constraints on homographies. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 2 (Feb 2002), 214–223.

[195] ZHANG, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 11 (2000), 1330–1334.

[196] ZHANG, Z. Camera calibration with one-dimensional objects. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 7 (2004), 892–899.

[197] ZHANG, Z., DERICHE, R., FAUGERAS, O., AND LUONG, Q.-T. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artif. Intell. 78*, 1-2 (1995), 87–119.

[198] ZHOU, F., AND LA TORRE, F. D. Factorized graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 38*, 9 (Sept 2016), 1774–1789.

[199] ZHOU, X., ZHU, M., AND DANIILIDIS, K. Multi-image matching via fast alternating minimization. *CoRR abs/1505.04845* (2015).