POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica e Informazione

Image Mosaicing: An approach based on Synchronization and Game Theory

Relatore: Prof. Luca Magri

Tesi di Laurea di:
Simone Francavilla, matricola 920142

Anno Accademico 2019-2020

Alla mia famiglia

Sommario

In Computer Vision, i metodi per allineare le immagini in fotomosaici senza soluzione di continuità sono stati ampiamente utilizzati nel corso degli anni. Per esempio, l'allineamento delle immagini al frame rate è usato in ogni videocamera che ha una funzione di "stabilizzazione dell'immagine". Nella comunità della fotogrammetria, metodi più intensivi manualmente basati su punti di controllo a terra rilevati o punti di legame registrati a mano sono stati a lungo utilizzati per registrare le foto aeree in fotomosaici su larga scala. Mentre la maggior parte delle tecniche di cui sopra lavora minimizzando direttamente le dissimilarità da pixel a pixel, una diversa classe di algoritmi lavora estraendo un insieme sparso di feature per poi abbinarle tra loro. Gli approcci basati sulle feature hanno il vantaggio di essere più robusti contro il movimento della scena e sono potenzialmente più veloci, se implementati nel modo giusto. Il loro più grande vantaggio è la capacità di "riconoscere i panorami", cioè di scoprire automaticamente le relazioni di adiacenza (sovrapposizione) tra un insieme non ordinato di immagini, il che li rende ideali per lo stitching completamente automatizzato di panorami presi da diversi utenti. Un problema correlato, noto come matching multi-vista, è la ricostruzione di tracce multi-feature che identificano lo stesso punto materiale da un insieme di immagini prese dalla stessa scena. In questa tesi presentiamo due tecniche per affrontare il problema dello stitching delle immagini e del matching multi-vista. La prima combina il concetto di warping as-projective-as-possible con la coerenza della registratura. In particolare, si ispira al concetto di warping flessibile dell'immagine mentre tiene conto degli errori che si accumulano quando si aggiunge un'immagine alla volta al mosaico, per mezzo di un approccio globale formalmente noto come sincronizzazione di gruppo, attraverso tutte le immagini. La seconda propone un approccio basato sulla teoria dei giochi per trovare le corrispondenze combinando ancora la proprietà di coerenza, ma applicato, invece, alle corrispondenze di feature, affrontando la limitazione presente nei precedenti lavori di non avere un indicatore affidabile per la congruità delle tracce multi-feature.

Abstract

In Computer Vision, methods for aligning and stitching images into seamless photomosaics have been widely used through the years. For example, frame-rate image alignment is used in every camcorder that has an “image stabilization” feature. In the photogrammetry community, more manually intensive methods based on surveyed ground control points or manually registered tie points have long been used to register aerial photos into large-scale photo-mosaics. While most of the above techniques work by directly minimizing pixel-to-pixel dissimilarities, a different class of algorithms works by extracting a sparse set of features and then matching these to each other. Feature-based approaches have the advantage of being more robust against scene movement and are potentially faster, if implemented the right way. Their biggest advantage is the ability to “recognize panoramas”, i.e., to automatically discover the adjacency (overlap) relationships among an unordered set of images, which makes them ideally suited for fully automated stitching of panoramas taken by casual users. A related problem, known as multi-view matching, is the reconstruction of multi-feature tracks that identifies the same material point from a set of images taken from the same scene. In this thesis we present two techniques to tackle the problem of both image stitching and multi-view matching. The former combines concept from as-projective-as-possible warping with registration consistency. Specifically, it takes inspiration from the concept of flexible image warping while accounting for the errors that accumulate when adding an image at a time to the mosaic, by means of a global approach formally known as group synchronization, across all images. The latter proposes a game-theoretical approach for finding matches and still combining the consistency property, but applied, instead, to matching correspondences, addressing the limitation present in the previous related works of not having a reliable indicator for the adequacy of multi-feature tracks.

Contents

Chapter 1

Introduction

1.1 Overview

Algorithms for aligning images and stitching them into seamless photomosaics are among the oldest and most widely used in Computer Vision as for instance, frame-rate image alignment is used in every camcorder that has an "image stabilization" feature. Image stitching algorithms create the high-resolution photo-mosaics used to produce today's digital maps and satellite photos. They also come bundled with most digital cameras currently being sold, and can be used to create beautiful ultra wideangle panoramas [45].

In the photogrammetry community, more manually intensive methods based on surveyed ground control points or manually registered tie points have long been used to register aerial photos into large-scale photo-mosaics [36]. One of the key advances in this community was the development of bundle adjustment algorithms that could simultaneously solve for the locations of all of the camera positions, thus yielding globally consistent solutions [21, 47]. One of the recurring problems in creating photo-mosaics is the elimination of visible seams, for which a variety of techniques have been developed over the years.

While most of the above techniques work by directly minimizing pixel-to-pixel dissimilarities, a different class of algorithms works by extracting a sparse set of *features* and then matching these to each other. Feature-based approaches have the advantage of being more robust against scene movement and are potentially faster, if implemented the right way. Their biggest advantage, however, is the ability to "recognize panoramas", i.e., to automatically discover the adjacency (overlap) relationships among an unordered set of images, which makes them ideally suited for fully automated stitching of panoramas taken by casual users [11].

1.2 Research Background

Aligning and stitching images into seamless mosaics is a procedure usually composed by three main steps: image registration, color correction and blending. Image mosaicing can be performed independently from (and prior to) the structure-from-motion and dense matching phases, that are instead required to generate orthophotos. The goal of mosaic creation is, in fact, to visualize a wide area on a single image under perspective projection, whereas orthophotos are orthographic projections.

In the last decades several methods for automatic image mosaicing appeared in the literature, proposing a complete pipeline for the final mosaic generation [17, 30, 12] or focusing the attention on the optimization of one of the previously cited steps [41, 33, 27].

1.2.1 Image registration

Algorithms for image alignment can be divided into two broad categories [44]: direct (pixel-based) and feature-based. Direct methods exploit the entire image data, thus providing very accurate registration but requiring at the same time a close initialization. Feature-based algorithms, instead, do not require initialization and can be computationally less expensive. Moreover, since the introduction of invariant features (e.g., SIFT, [28]) and robust feature matching, feature-based methods have gained increasing attention and are nowadays widely used. [12] proved that, formulating stitching as a multi-image matching problem and using invariant local features to find matching between the images, lead to a method insensitive to the ordering, orientation, scale and illumination of the input images.

1.2.2 Color correction

To obtain a clean, pleasant looking mosaic, a robust alignment process must be followed by color correction. Neighboring images can indeed show color and appearance differences due to exposure level variation, changes in lighting condition and different camera settings. Color correction methods proposed in the literature can be divided into model-based parametric approaches and non parametric ones [52]. The former assume that the relation between two images can be described by a color transfer function, whereas the latter consider no particular parametric format of the color mapping function and typically use a look-up table to directly record the mapping of the color levels. [52] evaluated the performance of various color correction approaches, showing how the gain compensation method by [12] and the local color transfer approach by [50] are fast, effective and general (applicable in various scenarios).

1.2.3 Blending

Once the seams have been placed and unwanted object removed, we still need to blend the images to compensate for exposure differences and other misalignments. However, it is difficult in practice to achieve a pleasing balance between smoothing out low-frequency exposure variations and retaining sharp enough transitions to prevent blurring (although using a high exponent does help).

Among the contributions in image blending, it is worth mentioning:

Laplacian Pyramid

An attractive solution to this problem was developed by Burt and Adelson [13]. Instead of using a single transition width, a frequency-adaptive width is used by creating a bandpass (Laplacian) pyramid and making the transition widths a function of the pyramid level.

Gradient Domain Blending

An alternative approach to multiband image blending is to perform the operations in the *gradient domain*. Reconstructing images from their gradient fields has a long history in computer vision [23], starting originally with work in brightness constancy [22], shape from shading [6], and photometric stereo [51]. Pérez et al. [35] showed how gradient domain reconstruction can be used to do seamless object insertion in image editing applications.

Exposure Compensation

Uyttendaele et al. [48] iteratively estimate a local correction between each source image and a blended composite and, as their results demonstrate, this does a better job of exposure compensation than simple feathering, and can handle local variations in exposure due to effects like lens vignetting.

High Dynamic Range Imaging

A more principled approach to exposure compensation is to estimate a single *high dynamic range* (HDR) radiance map from the differently exposed images [29, 18, 32, 37]. Most techniques assume that the input images were taken with a fixed camera whose pixel values are the result of applying a parameterized *radiometric transfer function* to scaled radiance values.

1.3 The Problem of Mosaicing

The problem of image mosaicing can be decomposed in what are called multi-view matching and image stitching. The former aims to find correspondences among image features which are supposed to belong to the same material point in the same scene. This is achieved by extracting and collecting significant features from every image thus obtain what are meant to be their feature descriptors. This descriptors are essential for finding correspondences through their similarities since each of them represents an important role in identifying significant image features. The property being exploited when extracting such feature descriptors is their invariance especially in scale and rotation but also illumination and noise.

When feature correspondences among images are found, we need to remark the difference in generating the so called multi-feature tracks, i.e., set of feature belonging to the same material point, and using them for alignment and stitching, which require estimating projective transformations—as already explained in the previous section—for composing an image mosaic.

1.4 Aim and Contributions

The purpose of this thesis is to present a novel technique in both context of image stitching and multi-view matching by still treating them as two decoupled problem.

Concerning image stitching we describe in this thesis a technique that combines the work presented in [53] and [40] about projective transformations, which we are going to refer to as APAP Synchronization. The reason behind this arises from the idea of exploiting the averaging property, known as synchronization, from a set pairwise measurements that need to agree on consistency, but applied on local image deviation which do not strictly follow the global projective trend. The main assumption behind the traditional image alignment relates to views that differ purely by rotation and not by translation, or that the imaged scene is effectively planar.

The idea proposed for the multi-view matching problem, here presented as Evolutionary MatchEIG, follow a game-theoretical approach described in [15], that try to solve the problem as a non-cooperative game in which two strategy are drawn from a population of individuals. Such individuals are interpreted as the extent of selecting certain image feature to represent a material point of the scene from which images are taken. This idea takes into account the descriptor space similarity among feature descriptors in order to find correspondences but lacks of the consistency property which is

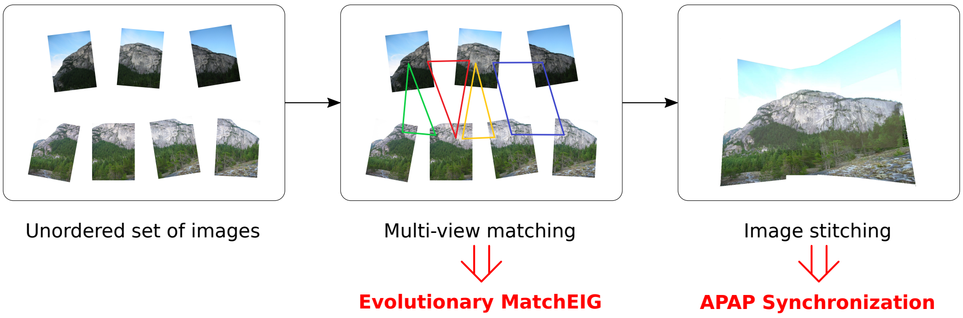


Figure 1.1: Mosaicing pipeline overview showing our contribution methods

vital when presenting the MatchEIG method in paper [31]. The contribution brought by [31] for presenting the proposed method, concerns the same consistency property based still on the one from the synchronization formulation but—as we will see later on—being applied on permutations of features. Specifically, given a set of noisy pairwise correspondences, jointly updates them so as to maximize their consistency, based on a spectral decomposition method.

1.5 Thesis Structure

This thesis is structured as follows:

- Chapter 2 introduces and describe all the main related works and their research field which play an important role in carrying out the work presented in this thesis;

- In Chapter 3 we describe the APAP Synchronization method, in particular its main idea followed by all the criteria necessary to fully represent it;

- In Chapter 4 we discuss Evolutionary MatchEIG along with its considerations and the general pipeline.

- Chapter 5 shows and evaluates the experiments being performed;

- Chapter 6 draws the final conclusions considering the results obtained from the experiments.

Chapter 2

State of the Art

2.1 Classification of the Main Related Works

Here we present the related work carried out in both field of image stitching and multi-view matching. The main stream of research are classified in the following way:

- **Image alignment and stitching**: the procedure of taking the alignment estimates produced by feature registration algorithms and blending the images in a seamless manner, while taking care to deal with potential problems such as blurring or ghosting caused by parallax and scene movement as well as varying image exposures.
  - **As-Projective-As-Possible warps** [53]: an estimation technique that is able to tweak or fine-tune the projective warp to accommodate the deviations of the input data from the idealized conditions. This significantly reduces ghosting without compromising the geometric realism of perspective image stitching.
  - **Mosaicing via Synchronization** [40]: a method to create high-quality seamless planar mosaics. It uses a global approach, known as synchronization, for image registration and color correction. correction.
- **Multi-view matching**: the problem of feature registration from a set of multiple images, in which the features grouped together refer to the same material point.
  - **Permutation Synchronization**: the feature matching between images can be represented by means of permutation that associates to each feature in one image the corresponding one in the

second image. By solving a synchronization problem where all partial permutations between image pairs are taken into account, it is possible to identify all the features that correspond to the same 3D point in the scene (see Appendix D) This problem has been addressed in [31] which introduce the MatchEIG method.

- **Game Theory for hypothesis validation**: the matching problem is formulated as a simultaneous optimization over the entire image collection, without requiring previously computed pairwise matches to be given as input. This formulation operates directly in the space of feature across multiple images, resulting in the final matches being consistent by construction, and has a natural interpretation as a non-cooperative game, which allows to leverage tools and results from Game Theory (see Appendix F). [15] proposes a formulation and realization of this problem by means of what are referred to as multi-feature matching games.

2.2 Image Alignment and Stitching

2.2.1 As-Projective-As-Possible warps

The authors of [53] introduced an estimation technique called *Moving Direct Linear Transformation* (Moving DLT) that is able to tweak or fine-tune the projective warp to accommodate the deviations of the input data from the idealized conditions. This produces *as-projective-as-possible* image alignment that significantly reduces ghosting without compromising the geometric realism of perspective image stitching. This technique thus lessens the dependency on potentially expensive post-processing algorithms. In addition, they describe how multiple as-projective-as-possible warps can be simultaneously refined via bundle adjustment to accurately align multiple images for large panorama creation.

Moving DLT

When images I and I' are obtained by translating cameras or the scene is not planar using a basic homographic warp inevitably yields misalignment or parallax errors. To alleviate this problem, [53] introduced the *Moving DLT* method. The idea is to warp each $\mathbf{x}_*$ using a *location dependent* homography

$$\tilde{\mathbf{x}}'_* \propto H_* \tilde{\mathbf{x}}_* \,,$$

where H_* is estimated from the weighted problem

$$\mathbf{h}_* = \arg\min_{\mathbf{h}} \sum_{i=1}^{N} \|w_*^i \mathbf{a}_i \mathbf{h}\|^2 \quad \text{s.t.} \quad \|\mathbf{h}\| = 1\,. \tag{2.1}$$

The scalar weights $\{w_*^i\}_{i=1}^N$ give higher importance to data that are closer to $\mathbf{x}_*$, and the weights are calculated as

$$w_*^i = \exp(-\|\mathbf{x}_* - \mathbf{x_i}\|^2/\sigma^2)\,. \tag{2.2}$$

Here, σ is a scale parameter, and $\mathbf{x}_i$ is the coordinate in the source image I of one-half of the i-th point match $\{\mathbf{x}_i, \mathbf{x}'_i\}$.

Intuitively, since Eq. (2.2) assigns higher weights to data closer to $\mathbf{x}_*$, the projective warp H_* better respects the local structure around $\mathbf{x}_*$ in contrast to Eq. (A.4)—from original DLT formulation on page 62—which uses a single and global H for all $\mathbf{x}_*$. Moreover, $\mathbf{x}_*$ is *moved* continuously in its domain I, the warp H_* also varies smoothly. This produces an overall warp that adapts flexibly to the data, yet attempts to preserve the projective trend of the warp, i.e., a flexible projective warp.

The problem in Eq. (2.1) can be written in the matrix form

$$\mathbf{h}_* = \arg\min_{\mathbf{h}} \|W_* A\mathbf{h}\|^2 \quad \text{s.t.} \quad \|\mathbf{h}\| = 1\,, \tag{2.3}$$

where the weight matrix $W_* \in \mathbb{R}^{2N \times 2N}$ is composed as

$$W_* = \text{diag}([\, w_*^1 \;\; w_*^1 \;\; w_*^2 \;\; w_*^2 \;\; \ldots \;\; w_*^N \;\; w_*^N \,])\,,$$

and diag$(\cdot)$ creates a diagonal matrix given a vector. This is a weighted SVD[1] (WSVD) problem, and the solution is simply the least significant right singular vector of $W_* A$.

Problem Eq. (2.3) may be unstable when many of the weights are insignificant, e.g., when $\mathbf{x}_*$ is in a data poor (extrapolation) region. To prevent numerical issues in the estimation, the authors of [53] propose to offset the weights with a small value γ within 0 and 1

$$w_*^i = \max(\exp(-\|\mathbf{x}_* - \mathbf{x_i}\|^2/\sigma^2), \gamma)\,. \tag{2.4}$$

This also serves to regularize the warp, whereby a high γ reduces the warp complexity. In fact as γ approaches 1 the resultant warp loses its flexibility and reduces to the original homographic warp.

[1] Singular value decomposition: a factorization of a real or complex matrix that generalizes the eigendecomposition of a square normal matrix to any $m \times n$ matrix.

Cells partitioning

The assumption underlying Moving DLT is that the input consists in inlier matches, but in practice mismatches and outliers exist among the data. For this reason, invoking Moving DLT, we remove outliers using RANSAC [19] with DLT as the minimal solver. Although [53] considers data where the inliers themselves may deviate from the projective trend, in practice, the outlier errors are orders of magnitude larger than the inlier deviations [46], thus RANSAC can be effectively used.

Solving Eq. (2.3) for each pixel position $\mathbf{x}_*$ in the source image I is unnecessarily wasteful, since neighboring positions will yield very similar weights Eq. (2.2) and hence very similar homographies. They thus uniformly partition the 2D domain I into a grid of $C_1 \times C_2$ cells, and take the center of each cell as $\mathbf{x}_*$. Pixels within the same cell are then warped using the same homography. The warp is globally projective for extrapolation, but adapts flexibly in the overlap region for better alignment.

Partitioning into cells effectively reduces the number of WSVD instances to $C_1 \times C_2$. Moreover, each of the WSVD instances are mutually independent, thus a simple approach to speed up computation is to solve the WSVDs in parallel.

A potential concern is that discontinuities in the warp may occur between cells, since cell partitioning effectively downsamples the smoothly varying weights Eq. (2.4). In practice, as long as the cell resolution is sufficiently high, the effects of warp discontinuities are minimal.

Further speedups are possible, for most cells, due to the offsetting many of the weights do not differ from the offset. To exploit this observation a WSVD can be updated from a previous solution instead of being computed from scratch by means of rank-one update [53].

Bundle Adjustment

Bundle adjustment [21, 47] is the problem of refining a visual reconstruction to produce jointly optimal 3D structure and viewing parameter (camera pose and/or calibration) estimates. Optimal means that the parameter estimates are found by minimizing some cost function that quantifies the model fitting error, and jointly that the solution is simultaneously optimal with respect to both structure and camera variations. The name refers to the “bundles” of light rays leaving each 3D feature and converging on each camera centre, which are “adjusted” optimally with respect to both feature and camera positions. Equivalently—unlike independent model methods, which merge partial reconstructions without updating their internal structure—all of the

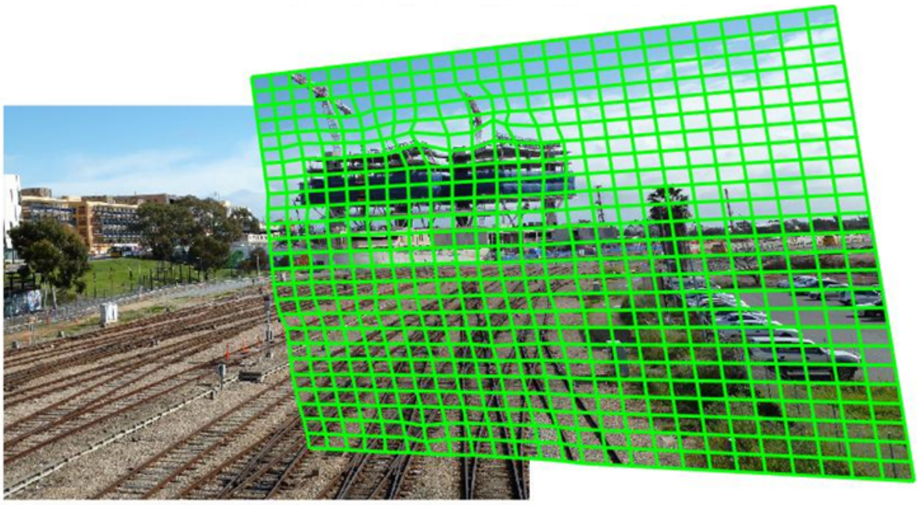


Figure 2.1: APAP stitching with source image partitioned in 25x25 cells

structure and camera parameters are adjusted together “in one bundle”.

To stitch multiple images to form a large panorama, pairs of images can be incrementally aligned and composited onto a reference frame. However, incremental stitching may propagate and amplify alignment errors, especially at regions with multiple overlapping images. Such errors can be alleviated by simultaneously refining the multiple alignment functions, prior to compositing [53, 44]. In [53] is shown how bundle adjustment can be used to simultaneously refine multiple as-projective-as-possible warps.

2.2.2 Mosaicing via Synchronization

The algorithm proposed in [40] tries to overcome some common issues in mosaic generation (e.g., misalignments, color correction, moving objects) thanks to the use of synchronization and the search for an optimal cutting path between overlapping images. The entire process is summarized in [31] and described in detail in the following paragraphs.

Homography estimation

The first step of the proposed procedure is to extract features from all the images (e.g., SIFT features, [28]) and match them. A robust feature matching algorithm should be used in order to avoid wrong matches that can cause strong misalignments between the images. For this reason, the method proposed in [31] has been chosen. Starting from the correspondences between pairs of views, it jointly updates them so as to maximize their consistency.

Pairwise homographies are then robustly estimated using RANSAC [19], computing image transformation parameters through the Direct Linear Transformation (DLT) method [1]. A possible solution to project all the images in the same reference system for mosaic generation is to compose relative transformations multiplying the obtained pairwise homographies. However, this approach accumulates error at each successive multiplication. To solve this problem, synchronization over SL(3) is applied, converting in this way pairwise homographies into absolute ones. This guarantees that all relative information are considered simultaneously, minimizing misalignment errors among the whole dataset.

To improve the accuracy of the synchronization process, a weighting factor can be assigned to each pairwise homography, that describes its reliability. In practice, the unitary elements of the adjacency matrix A contained in Eq. (E.2) are replaces by the estimated weights. In the proposed procedure, these weights are assumed to be proportional to the area of the convex hull that contains the features matched in each image pairs.

Color correction

Changes of the illumination conditions, different camera settings and vignetting are some of the causes that make the seams of the mosaic visible, even when the scene is planar, the images are sharp and the alignment is perfect. Color variations between overlapping images should be modelled by a non-linear function and often involve the three color channels simultaneously. However, the simplified approach that considers the RGB channels independently and that models the transformation with an affinity proved to work well. Thus, in the proposed method the relation between the three color channel of adjacent images (i, j) is assumed to be an affine transformation, that can be written in matrix form as

$$\begin{bmatrix} C \\ 1 \end{bmatrix}_i = \begin{bmatrix} a_c & b_c \\ 0 & 1 \end{bmatrix}_{i,j} \cdot \begin{bmatrix} C \\ 1 \end{bmatrix}_j ,$$

where C is in turn R,G, or B. Formulating the problem in this way corresponds to estimating the parameters of three affine transformations between each pair of overlapping images, that have to be then composed in order to compute a global color correction for each single image. It is easy to see that this problem can be solved via the synchronization over the *Affine Group* Aff(d).

In presence of small residual misalignments, pixel-based estimation of pairwise affine transformations can lead to inaccurate results. An alternative

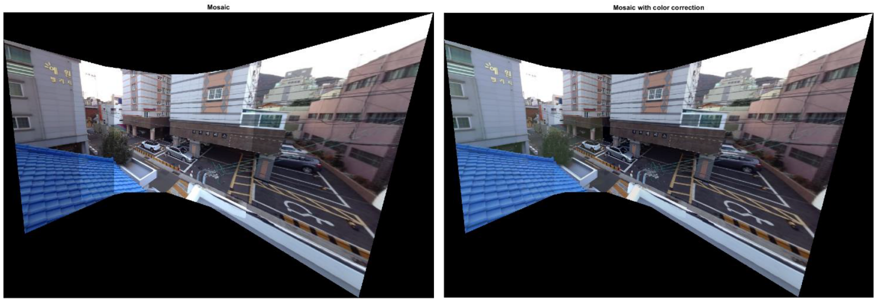


Figure 2.2: Mosaic before (left) and after (right) applying color correction

robust approach, adopted in [40], consists in exploiting the histograms of the overlapping area computed for both images. The parameters of the affine transformation are computed as the angular coefficient and intercept of the straight line that fits the plot of one cumulative histogram versus the other cumulative histogram [16].

Once all the relative affine transformations have been computed for each color channel, the absolute ones can be retrieved via synchronization, as previously explained for the homographies. A weighting matrix can be introduced, where the weights are proportional to the overlapping area size, in order to give more confidence to the most reliable pairwise color transformation. Please note that synchronization retrieves absolute affine transformation, up to a global one. This degree of freedom can be fixed by choosing one image that does not undergo color correction. The unaltered image can be identified automatically as the one that has the best color balance, or it can be defined by the user.

Figure 2.2 shows an application to an image mosaic of color correction with the described method (source: Marearts roof dataset[2]).

2.3 Multi-View Matching

In the context of multi-view matching, and even more in general of feature matching, much research has been carried out. We introduce the ideas reported in [31] and [15] which are essential in introducing the concept developed for this thesis. The former is based on the maximization of the global consistency of pairwise correspondences derived from the permutation synchronization problem, the latter, instead, consists in a game-theoretical approach which operates directly in the space of feature descriptor and for-

[2] `http://study.marearts.com/2013/11/opencv-stitching-example-stitcher-class.html`

mulates the problem as a non-cooperative game, without requiring previously computed pairwise matches to be given as input.

2.3.1 Spectral method

The Spectral method of [34] treats the absolute permutation block-matrix X—from the optimization problem described in Eq. (E.6)—as a real matrix instead of a binary matrix and enforces the columns of X to be orthogonal, resulting in the following optimization problem

$$\max_{U^T U = I_d} \operatorname{trace}(U^T \hat{Z} U)\,, \tag{2.5}$$

where the notation U instead of X is used to underline that, due to the relaxation, the optimal U will not be composed of partial permutation matrices. Equation (2.5) is a generalized Rayleigh problem, whose solution is given by the d leading eigenvectors of $\hat{Z}$. In order to obtain proper correspondences from U, each $m_i \times d$ block is projected onto the nearest permutation matrix via the Kuhn-Munkres algorithm [26], which solves a linear assignment problem, thus returning a set of estimated absolute permutations.

2.3.2 MatchEIG method

The Spectral method is extremely fast, as multi-view matching is solved in one shot via spectral decomposition. However, since absolute permutations are computed, the knowledge of the size of the universe d is required, which is not available in practice. The importance of a correct estimate of d is also demonstrated experimentally in [31].

The authors of paper [31] introduce a novel technique for multi-view matching, dubbed MatchEIG, which inherits the positive aspects of the Spectral method, namely efficiency and simplicity, and at the same time it overcomes its drawback, i.e., the need of the correct value of d as input. The key observation is that relative permutations are independent from d, thus a method that aims at producing relative permutations instead of absolute ones can get by without knowing precisely d. Specifically, this method proceeds as follows. First, the top d eigenvectors of $\hat{Z}$ are computed and collected in a $m \times d$ matrix U, as done by Spectral. Let D be the diagonal matrix containing the corresponding d eigenvalues $\lambda_1, \ldots, \lambda_d$. The matrix

$$\hat{Z}_d = U D U^T$$

is the solution of Eq. (E.5) under the spectral relaxation. In this way we get an estimate of Z—which contains relative permutations, and this is a key difference with respect to the SPECTRAL method that provides an estimate of X—which contains absolute permutations.

Suppose that we are given an estimate $\hat{d}$ of the size of the universe such that $\hat{d} \geq d$ and is computed $\hat{Z}_{\hat{d}}$ accordingly. Since $\hat{Z}$ has approximately rank d, we expect that the least $\hat{d} - d$ eigenvalues $\lambda_{d+1}, \ldots, \lambda_{\hat{d}}$ are smaller than the top d eigenvalues, thus the corresponding eigenvectors in U have a limited impact on $\hat{Z}_{\hat{d}}$, in particular: $\|\hat{Z}_d - \hat{Z}_{\hat{d}}\|_2 = |\lambda_{d+1}|$.

Note that, due to the relaxation, the $m_i \times m_j$ blocks $\hat{Z}_d$ are not guaranteed to be partial permutation matrices. In order to enforce this constraint, two different strategies are analyzed. A first stage common to both consists in setting to zero all the entries smaller than a given threshold t. In the experiments carried out in [31] it has been set $t = 0.25$ in simulations and $t = 0.5$ in real experiments. A higher threshold allows for more missing matches, and this is useful in real datasets to model the presence of isolated features.

Then, a principled approach consists in projecting each block onto the closest partial permutation matrix via the Kuhn-Munkres algorithm. In [31] this method is called MATCHEIG-CP, where CP stands for "closest permutation". This projection, however, slows down the computing time, so in MATCHEIG algorithm a greedy strategy is used so that, if applied to each block, it returns a valid permutation, although not the closest one. This strategy, implemented by the authors of [31] as function `matrix2perm()`, is approximate but it produces no noticeable loss in accuracy, while greatly boosting the speed, as experiments demonstrated.

The `matrix2perm()` function takes a matrix C as input and returns a (partial) permutation matrix P constructed as follows: search among the non-zero entries of C for the ones where the maximum over the corresponding row or column is achieved. These entries are then sorted by decreasing magnitude and examined sequentially starting from the largest element: let (i, j) be the index of the current entry, and let P be the output matrix, initialized to 0; then $[P]_{i,j}$ is set to 1 provided that P remains a partial permutation.

The idea behind this procedure is the following. For a given row i, which corresponds to a feature in one image, each entry $[C]_{i,j}$ represents the extent of pairing between feature i and feature j, and the greatest element in this row can be regarded as the most likely correspondence. The same holds for each column. To these putative matches we need to apply the principle of exclusion, and the authors of [31] propose to do it in a greedy way [42]: the strongest match wins and inhibits other 1s to be placed in its row or column.

Note that, because of noise, $\hat{Z}_{\hat{d}}$ is full, in general, and its size can become large in practical scenarios. However, this matrix needs not to be explicitly computed, for only one block is needed at a time. Specifically, when an image pair (i, j) is considered, the product $U_i D U_j^T$ need to be computed, where U_i denotes the $m_i \times \hat{d}$ block in U corresponding to image j. Therefore we only need to store the matrix $UD^{\frac{1}{2}}$ instead of $\hat{Z}_{\hat{d}}$, and this observation considerably reduces the storage space necessary to run the algorithm.

Note also that the projection step (either via the Kuhn-Munkres algorithm or via the approximate strategy) can be performed in parallel, since each image pair is independent from the others, thus speeding up the process.

2.3.3 Pairwise Game-theoretical Matching

During the last few years, Game Theory has been adopted to perform hypothesis validation within a wide range of scenarios. This is the case, for instance, of the seminal paper by Albarelli et al. [2], where a game-theoretical framework is used for finding correspondences between segments and to perform point-pattern matching. Registration of rigid and deformable shapes have been also addressed in [3] and [38], and object-in-clutter recognition in [39]. Other relevant works are about feature selection [5] and image segmentation [25]. Most of these methods follow a common script:

- A set of initial hypotheses is selected from the solution space of the problem;

- A payoff function is defined between each pair of hypotheses in order to express the level of mutual validation;

- A hypotheses population, represented as a probability distribution, is evolved through some dynamics.

A highly relevant application, is the one presented in [4], called Game Theoretical Matcher (GTM). Here, the initial hypotheses are putative matches between features and the selection process operates according to a payoff that accounts for how well the affine transformation induced by one match can be applied to a competing hypothesis. This addresses the same problem of [15], i.e., to extract coherent feature matches between images in order to enable 3D recovery. Moreover, it also adopts a game theoretical framework, albeit it uses it in a very different way.

As experiments in [15] shows, the pairwise matching process fails to recover some of the correct matches. This is due to the limited mutual support that the individual matches can establish in the simple pairwise setting, which may give rise to visual ambiguities and is in fact more prone to the presence of structured noise in the images and to random outliers.

The next section will describe an approach to sidestep these limitations by leveraging the information contained in the *whole* collection of images.

2.3.4 Multi-Feature Matching Game

Feature matching methods exploiting the game-theoretical framework (e.g., [4], [38], [39]) usually consider matches between two images as hypothesis and validate them to find the set of assignments that are the most suitable. These pairwise solutions can in principle be recomposed so as to form coherent tracks. [15] proposes exactly the opposite approach: validating multiple correspondences of the same feature among several images by finding a mutually coherent set according to the feature descriptor. In this view, each game will produce just a *single* multi-feature match rather than a set of pairwise matches.

The motivating idea is that, if n images are available, searching for a set of landmarks exhibiting strong compatibility between each pair should result in a much stronger validation of the feature descriptors, which are required to be repeatable over all the $n(n-1)/2$ pairs. This, in turn, helps to avoid wrong matches resulting from random descriptor similarities that can easily happen if only two images are involved. Furthermore, the obtained tracks will be inherently multi-way, ruling out the need for an explicit merge of pairwise correspondences. Finally, when the single tracks are grouped together, enforcing their geometrical coherence throughout several images will benefit from the increased dimensionality. These two steps (multi-feature selection and multi-feature validation) are performed through two separate games, using different hypothesis sets and payoff functions, which are going to be described later on.

Multi-Feature Selection Game

The goal of this game is to find features that agree on their descriptor throughout the whole image sequence (or at least images where the feature is visible). The result will be the extraction of a single track characterized (hopefully) by high reliability. Note that, differently from [4], no geometric information is used as this method just relies on the descriptor vectors.

First, query candidates are selected from all the feature points of all the images. This is carried out by estimating the density of the features in the feature-descriptor space and selecting *low-density* (i.e., uncommon) descriptors under the assumption that these descriptors are more distinctive. The density estimation is performed non-parametrically through k-nn density estimation: Let x be a point in the descriptor space, and $B_k(x)$ be the minimal ball centered at x containing k descriptors of k features. Then the k-nn density at x is $d_k(x) = \frac{k}{|B_k(x)|}$, where $|A|$ is the volume of the set A. With the density at hand, select the N least common features (lowest density) as query points and create a selection game for each of them. Given a query point, the selection game is as follows:

Hypotheses

for each image extract a fixed proportion p of the features extracted from that image that are closest (in the descriptor space) to the query point.

Payoff

The payoff is defined as a Gaussian over the descriptor distances. This makes sense, since we are considering descriptors to be originated from the same phenomenon and their drift can be reasonably modeled as a non-biased random error with standard deviation σ_a. Note, however, that features from the same image are incompatible with one another, thus their payoff is set to 0 regardless of their descriptor:

$$\pi(i,j) = \begin{cases} \frac{1}{\sigma_a\sqrt{2\pi}} e^{-\frac{|D(f_i)-D(f_j)|^2}{2\sigma_a^2}} & \text{if } I(f_i) \neq I(f_j), \\ 0 & \text{otherwise.} \end{cases}$$

Parameter σ_a can be used to tune the expected drift, which is clearly dependent on the feature descriptor adopted, on its dimensionality and, finally, on the strictness that is needed to enforce on the selection process. Smaller σ_a values results in a more selective process and vice versa.

Setting payoff 0 between features coming from the same image enforce a very important theoretical property of this method. The theorem proven in [15] showed that the support of a population evolved through a Feature Selection Game contains at most one feature from each image. This theorem is key to the feasibility of the proposed approach since it allows to avoid to include in the final solution two or more hypotheses originating from the same image (which is indeed the case, for instance with highly repeated structure such as walls, facades, and many man-made objects).

This configuration, which is defined as *multi-feature*, collects the most repeatable instances among all the features close to the query point. This process can be repeated for each query point resulting in the computation of exactly N distinct multi-features.

Multi-Feature Validation Game

While the tracks extracted in the selection game are highly similar from a photometric point of view, their extraction does not enforce any form of mutual geometric consistency among them. The authors of [15] propose a validation scheme that selects the *geometrically consistent* multi-features by performing an additional game over them.

Hypotheses

The set of multi-features extracted with the selection games. Each multi-feature refers to a single material point, thus it must contain at most one feature from each image. More formally, multi-features are sets $\alpha = \{f_i, i \in 1, \dots, n \mid f_i, f_j \in \alpha \implies I(f_i) \neq I(f_j)\}$.

Payoff

In order to play this second game, we must define a payoff between multi-features. This differs from GTM [4] as we do not assume the transformations to be locally affine, neither we can perform epipolar validation, since we need to define a payoff between two tracks, and we would need at least 5 to produce a fundamental matrix. Rather, define some property that can be preserved throughout subsequent shots of the same subject and that can be verified between two tracks. Given the 3D position of each tracked point, the distance between two of them would be a suitable measure. Unfortunately, only the projection on the image plane of the observed points is known. However, each feature f_i comes with an observed scale $S(f_i)$ and changes of the depth of the point with respect to a camera would result in inversely proportional changes in the observed scale through a constant k, which is a characteristic of the planar patch responsible for the observed feature. Under the assumption of moderate rotation between views, such constant is related to the size of the original object and can be used to express a point's 3D position: If k is known for the object that generated feature f_i, its position with respect to the reference frame of camera $I(f_i)$ is:

$$P(f_i, k) = \frac{k}{S(f_i)} \begin{bmatrix} U(f_i) & V(f_i) & 1 \end{bmatrix}^T$$

where $U(f_i)$ and $V(f_i)$ are normalized coordinates of f_i on the image plane of the observing camera $I(f_i)$. Let us consider features f_i and f_j extracted from the same image I_i observing two material objects (i.e., belonging to two tracks) α and β. If we know the values of k for such objects, we can compute the estimated length of the segment connecting α and β as $L(f^i_\alpha, f^i_\beta, k_\alpha, k_\beta) = \|P(f^i_\alpha, k_\alpha) - P(f^i_\beta, k_\beta)\|$. If tracks α and β have no outliers and perfectly accurate feature localization, then the variance σ^2_L of the distance L between the 3D points can be used as a measure of geometric inconsistency that is intrinsically multi-way. However, since k_α and k_β are not known, it is not possible to compute a value for σ^2_L, but a lower bound can be computed by minimizing its value over the unknown parameters. To this end, we have to account for an unrecoverable scale factor in the two patch sizes since we can always trade patch size for depth, thus we fix this scale setting $k^2_\alpha + k^2_\beta = 1$, obtaining the following multi-feature variance:

$$\sigma^2_{\alpha\beta} = \min_{k_\alpha, k_\beta} \sigma^2_L \quad \text{s.t.} \quad k^2_\alpha + k^2_\beta = 1\,.$$

With this define the payoff as follows:

$$\pi(\alpha, \beta) = \frac{1}{\sigma_b \sqrt{2\pi}} e^{-\frac{\sigma_{\alpha\beta}}{2\sigma^2_b}}\,.$$

In order to compute this variance, define v_α and v_β as long vectors concatenating all the observed 3D points modulo the parameters k_α and k_β:

$$v_i = \left[\frac{U(f^1_i)}{S(f^1_i)}, \frac{V(f^1_i)}{S(f^1_i)}, \frac{1}{S(f^1_i)}, \dots, \frac{U(f^n_i)}{S(f^n_i)}, \frac{V(f^n_i)}{S(f^n_i)}, \frac{1}{S(f^n_i)}\right]^T\,.$$

With these vectors at hand, we note that $\|k_\alpha v_\alpha - k_\beta v_\beta\|^2 = n(\sigma^2_L + \mu^2_L)$. In order to estimate k_α and k_β substitute in the computation of μ_L the Euclidean distance between the points with their Manhattan distance, obtaining: $n\mu_L \approx s^T(k_\alpha v_\alpha - k_\beta v_\beta)$, where s is a vector satisfying $s_i = \text{sign}(k_\alpha v_\alpha - k_\beta v_\beta)$. While this approximation is a bit rough, use it only to estimate k_a and k_b, and not to compute the variance $\sigma_{\alpha\beta}$ directly:

$$n\sigma^2_L = [k_\alpha, k_\beta] A [k_\alpha, k_\beta]^T, \qquad A = \begin{bmatrix} v^T_\alpha E v_\alpha & v^T_\alpha E v_\beta \\ v^T_\beta E v_\alpha & v^T_\beta E v_\beta \end{bmatrix}\,,$$

where $E = I - \frac{1}{n} s s^T$. Hence, estimate k_α and k_β by initializing them to $\sqrt{2}/2$ and iteratively computing s with the current values and re-estimating $[k_\alpha, k_\beta]^T$ as the eigenvector associated with the smallest eigenvalue of A. With the estimated values of k_α and k_β at hand, we can compute the actual Euclidean distances between feature points (modulo global scale) and thus their variance $\sigma_{\alpha\beta}$ as:

$$\sigma_{\alpha\beta} = \frac{1}{n} \sum_{i=1}^{n} L(f^i_\alpha, f^i_\beta, k_\alpha, k_\beta)^2 - \left(\frac{1}{n} \sum_{i=1}^{n} L(f^i_\alpha, f^i_\beta, k_\alpha, k_\beta)\right)^2\,.$$

As demonstrated by the experiments in [15], the remaining tracks, that include at least one mismatched feature, result in a low mutual payoff which, in turn, drives them to extinction.

Chapter 3

APAP Synchronization

3.1 Overview

The as-projective-as-possible warps [53] allow to to accommodate the deviations of the input data from the idealised conditions reducing artifacts such as misalignments or “ghosting” effects by means of cell partitioning of images. This technique, however, is limited to pairwise image stitching with only one image that adapts to the other. On the other hand, the homography synchronization [40] proposes a solution for synchronizing homography estimations, thus performing only globally projective warps.

We aim at overcoming these limitations by presenting a novel method that handle deviations not modeled by global homographies but at the same time is able to enforce consistency between transformations involving multiple images—as in synchronization methods.

The general idea of APAP Synchronization is to synchronize the homography estimations among cells from all the images, thus building the synchronization graph over them. In other words, the graph will be composed of image cells representing nodes and the homography estimations, needed to warp points of one cell to the other, representing arcs. We formalize the proposed method with the following criteria used for building the synchronization graph:

- **Image adjacency graph**: define the APAP estimations to be computed by determining the pairs of images that have an overlapping region with the global alignment.

- **Overlapping criterion**: describe the way cells in overlapping regions of pairs of images are estimated through synchronization.

- **Non-overlapping criterion**: describe the way cells in non-overlapping regions of pairs of images are estimated through synchronization.
- **Arc criterion**: define the selection of arcs composing the synchronization graph over the cells along with the related homographies as pairwise measure.
- **Weight criterion**: describe the assignment of weight to arcs between cells.

Each node in the synchronization graph represents a cell of certain image, except of the one defined as *root*, i.e., the image selected for not undergoing any transformation (up to translation factor in the final mosaic composition). For this reason, the root image is associated to only one node in the graph since it is not required to be partitioned in cells associated with different transformation.

3.2 Image graph criterion

We need to recall that given a pair of images, the APAP stitching estimation is an asymmetric relation (an image flexibly adapts to the other but the the transformation can not be directly inverted).

After obtaining the (weighted) adjacency matrix among image homographies for classic synchronization, compute the *maximal spanning sub-DAG* thereof. This directed graph is defined over images as nodes instead of cells.

This DAG is obtained as follows:

1. The original image graph is assumed weakly connected (i.e., ignoring arc orientations)
2. Arbitrarily select the image that does not undergo transformation as root
3. Compute the *Minimum Spanning Tree* (e.g., Kruskal algorithm)
4. Then add arcs with only one orientation from the original graph that does not generate a cycle

The resulting DAG define the ordered pairs of images for which is performed APAP estimation (i.e., pairs of source and target images).

For each arc (i, j) in the DAG define target and source image respectively for pairwise APAP estimation.

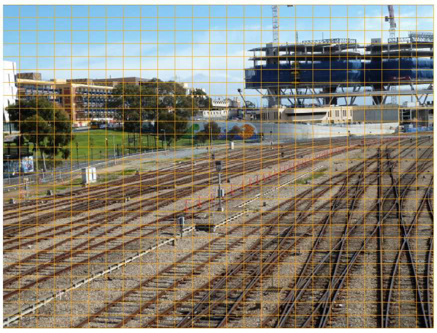


Figure 3.1: Target image

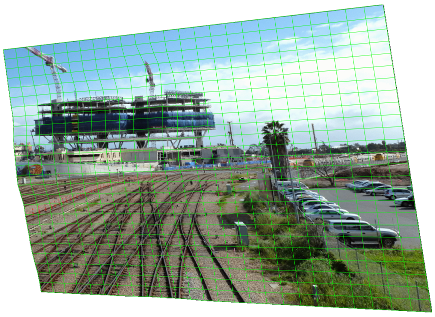


Figure 3.2: Source image

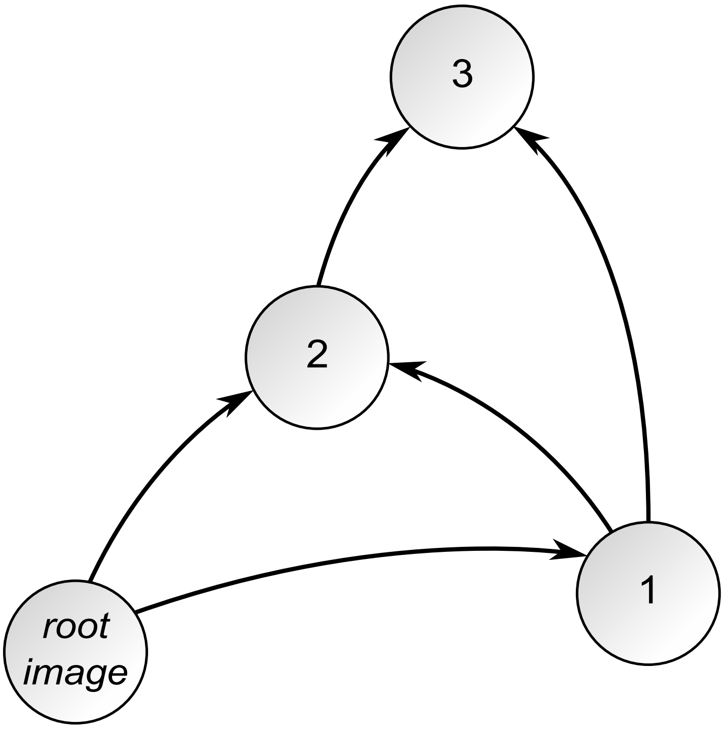


Figure 3.3: Maximal spanning sub-DAG from a set of 4 images. Differently from a tree, each node can be reached from the root node with at least one path so to allow redundancy in noisy measures. In the same way, it guarantees the asymmetric relation of pairwise APAP estimation. Each arc connects the target image to the source one in each APAP estimation to be computed.

3.3 Overlapping criterion

For each ordered pair of images i and j respectively target and source images, from an arc (i, j) the DAG, compute the homographies and the warped meshgrid of the source image.

Arc criterion

for each cell in the source image, defined as a quadrilateral by its vertices, obtain all the overlapping cells in the (un-warped) meshgrid of the target image, thus assign an arc from each of this cells to the current one in the source image. In the case the target image is the root node, assign only one arc from it (see Figs. 3.4 and 3.5).

Weight criterion

assign the weight proportionally to the covered area of the cell of the source with the one in target, so that they sum up to one for one cell of the source image.

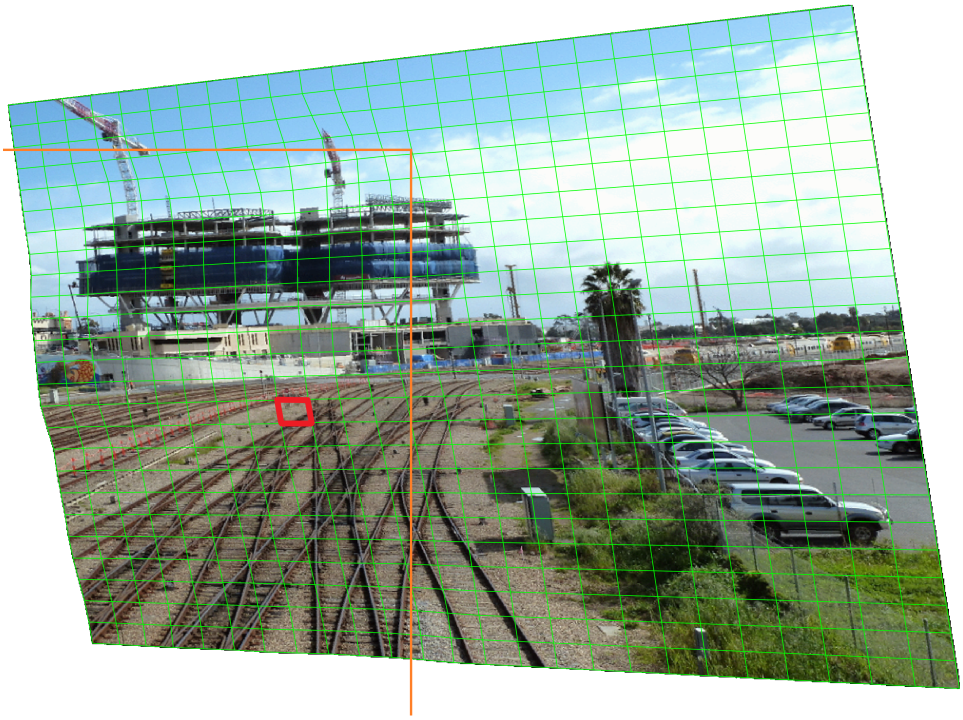
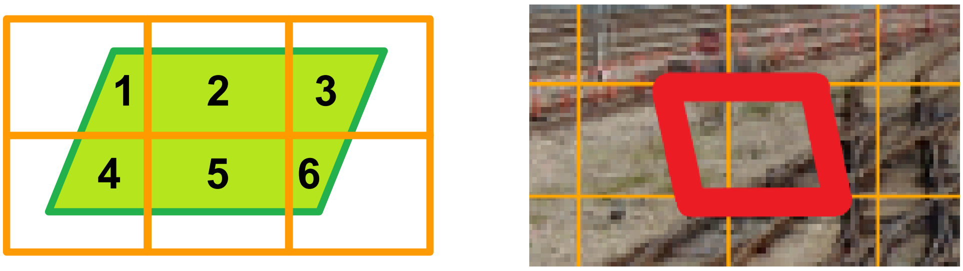


Figure 3.4: Source image: overlapping image boundary (orange) and considered overlapping cell (red)



(a)

(b)

Figure 3.5: One cell of the source image (orange) maps into more cells of target image meshgrid (green on the left, highlighted in red on the right) in different overlapping areas

3.4 Non-overlapping criterion

This criterion is applied in case a cell of the source image does not overlap with any of the cell of the target image, then outside the overlapping region of the pair of images.

This criterion is based combining these two ideas:

1. Cells that are close to the target image are mapped to the closest cell therein. The closest cell accounts for the most information regarding the local deviation of that area of scene that results in the entire mosaic, basing on the principle of space locality the idea behind APAP is built upon (see Fig. 3.6).

2. Cells that are far from the target image directly maps to root image node by means of the global homography between them (see Fig. 3.7).

In this case the closest cell does not bring enough information about the local deviation of the warping for that cell of the source image. Indeed, being far from the matching points the warping tends to be globally projective. Being the information brought by the closest cell is likely to be “misleading” we cover this lack by synchronize the considered cell directly to the root image node through the global homography of the source image required to be known a priori (e.g., obtained by classic synchronization among images).

Apply these two considerations for every non-overlapping cell:

Arc criterion

assign the cell of the source image to the closest one, in terms of Euclidean distance of the centroids, of the target image. Also assign an arc to the root node so to account also for the global transformation (i.e., the one computed without APAP warping). If the target image node is root node, then closest point in the target meshgrid is considered instead.

Weight criterion

for the closest cell of the source image the weight for the arc is computed similarly as the one used for weighting the matches in APAP estimation, with $\mathbf{x}'_*$ and $\mathbf{x}_{\text{closest\_cell}}$ the centroids of current (warped) cell of the source image and the closest cell respectively

$$\text{weight}_{\text{closest\_cell}} = \exp(-\|\mathbf{x}'_* - \mathbf{x}_{\text{closest\_cell}}\|^2/\sigma^2)\,.$$

For the arc with the root node is assigned its complementary.

$$\text{weight}_{\text{root}} = 1 - \text{weight}_{\text{closest\_cell}}\,.$$

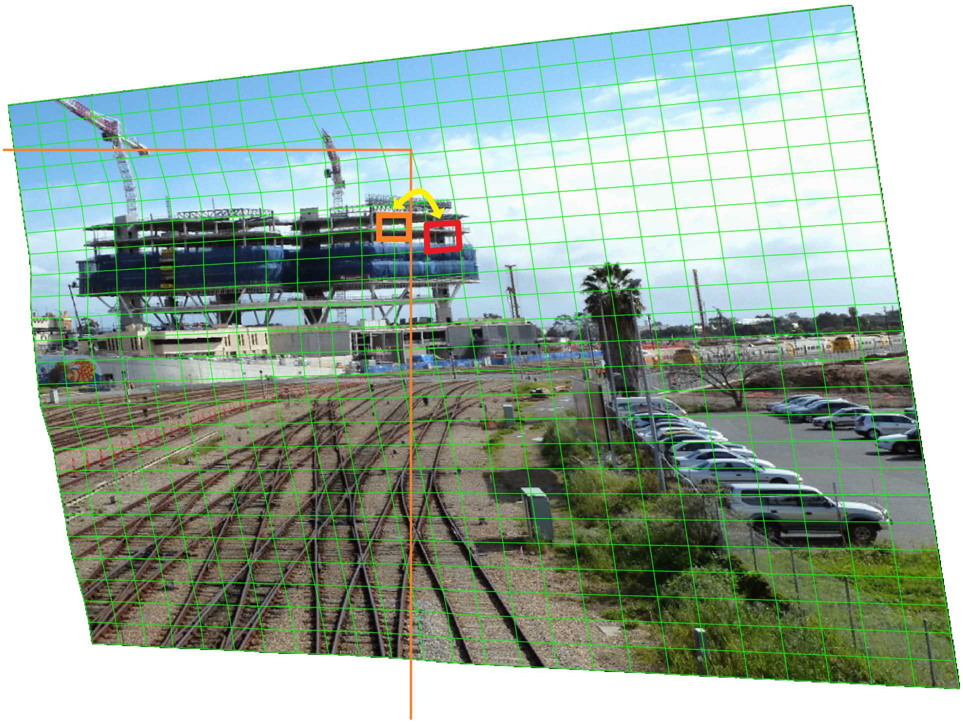


Figure 3.6: Non-overlapping criterion: the considered cell is very close to the target image meshgrid, so the estimation can directly rely on the closest cell present on it.

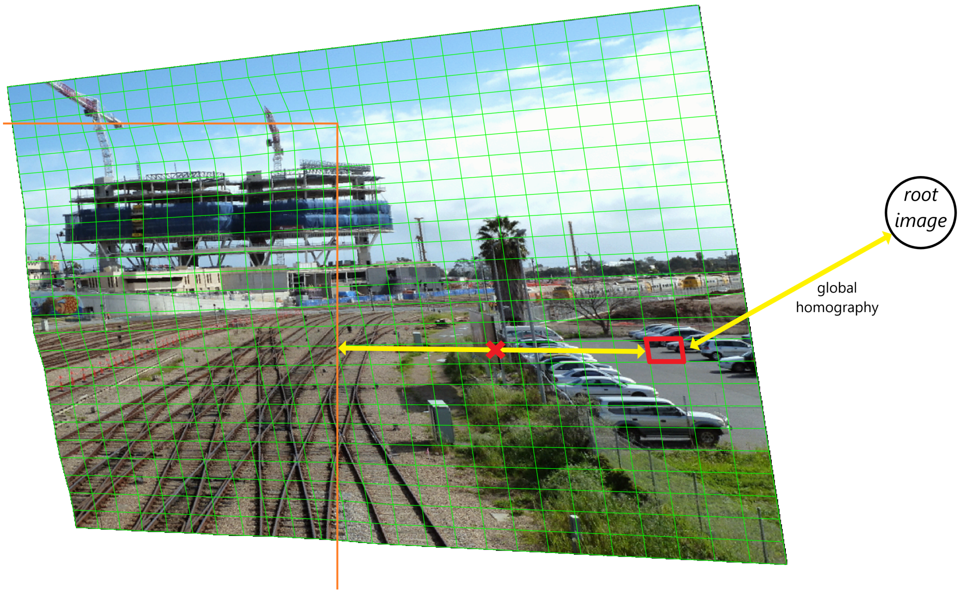


Figure 3.7: Non-overlapping criterion: the considered cell is visibly far from the target image meshgrid, so an estimation based on the closest cell would be inaccurate. On the other hand, the global homography would be a preferable choice.

Moreover, since the arc from the root node is only needed to cover the lack of information concerning the local warping between source and target images, the relation to the global homography should be used only for non-overlapping cells of the source image that are not covered by any other target image in the mosaic image set.

3.5 Discussion

The method presented in this chapter as APAP Synchronization deals with the homography synchronization problem but extended to cell partitions throughout all images. The crucial part is obviously how to compose the graph as well as assigning the corresponding weights to each arc. One node of such graph is chosen to represent the reference image which does not undergo any transformation referred to as root node. All the other nodes are connected in such a way they overlap when applying APAP warp estimation in their respective images. If a cell of the source image does not overlap with any of the target image cell, this one is synchronized both with global homography estimation and the closest target image cell in a weighted average depending on how much is cell distant from the target image meshgrid.

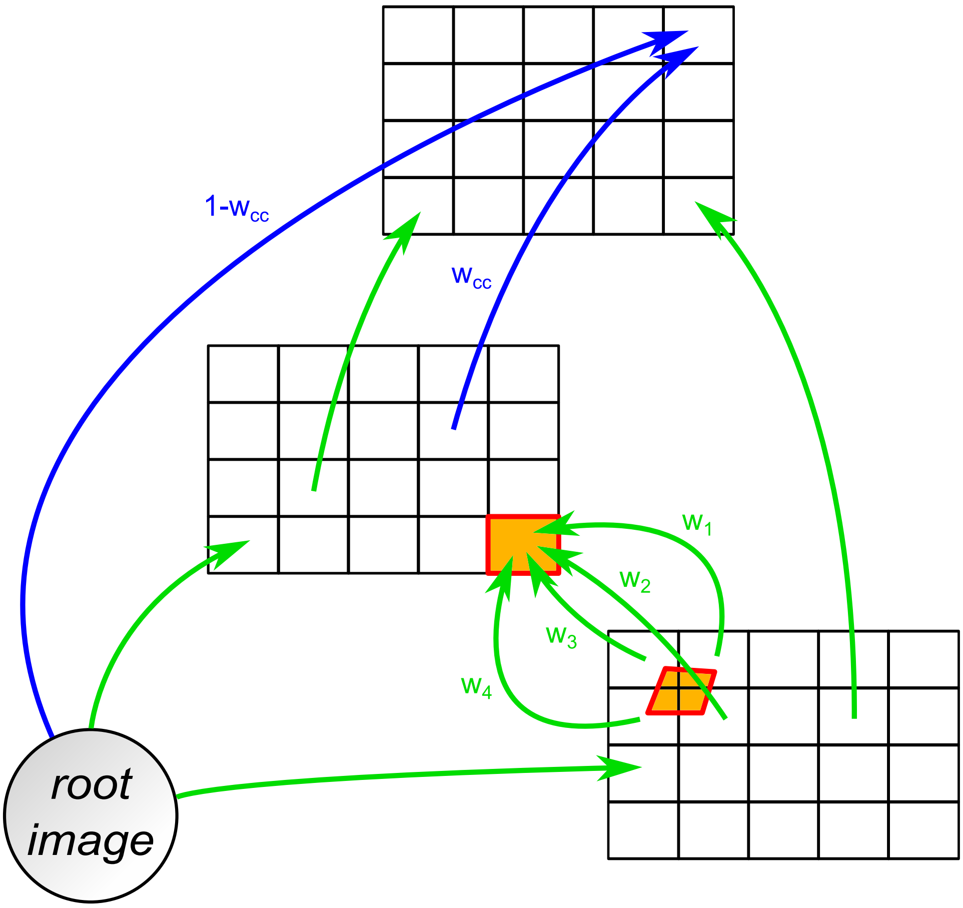


Figure 3.8: APAP Synchronization graph: Arcs in green and blue represent the ones built according to the overlapping and non-overlapping criterion respectively. The four values w_1, w_2, w_3, w_4 (which they sum to one) indicate the arc weights proportional to the overlapping regions of the highlighted source cell mapped into the target meshgrid (the bottom one). To the top meshgrid two arcs are directed according to the non-overlapping criterion which connect its closest cell node and the root image one to a non-overlapping node with two complementary weights, respectively w_{cc} and $1 - w_{cc}$.

Chapter 4

Evolutionary MatchEIG

4.1 Overview

The MatchEIG method [31] propose a novel solution to the multi-view matching problem that, given a set of noisy pairwise correspondences, jointly updates them so as to maximize their consistency. On the contrary multi-feature games [15] are able to produce multi-feature tracks—and thus pairwise correspondences—directly from features itself in terms of descriptor space while possibly lacking of cycle consistency, namely the composition of pairwise matches along any loop should give the identity.

The main concept behind Evolutionary MatchEIG arises from the idea of combine the averaging property from the consistency constraints with the feature similarities in term of descriptor space. To do so, we consider the approach used in [31] regarding the implementation of the function—referred to as `matrix2perm()`—for converting a matrix of real values into a valid permutation one, in which each entry in row i and column j in matrix $\hat{Z}_{\hat{d}}$ represents the *extent* of pairing the features associated to i and j.

Starting from the set of images, all the features (e.g., SIFT) are obtained encompassing both keypoints and descriptor vectors. Next, a set of query points are retrieved from feature previously obtained as in [15]. Features that are not included in any of the computed hypothesis sets are hence discarded. The ESS is computed and for each query point is obtained a track of features (i.e., multi feature) from different images. The number of query points is set to a value less or equal to $\hat{d}$, since the maximum number of distinct tracks must not be more than the size of the universe.

Based on [15], treat each value $x_i(t)$ from a certain game g and feature i

as a (conditioned) probability distribution described as

$$P(i\,|\,g) \overset{\text{def}}{=} P(D(f_i)\,|\,g) = \frac{x_i}{\max_{i' \in g} x_{i'}} \sim Be(\cdot) \qquad \forall i \in g\,, \tag{4.1}$$

The expression shown in Eq. (4.1) represents the probability of i to belong to the feature set described by game g. In general probability independence does not hold for two different features belonging to the same hypothesis set since we need to consider the fact that two distinct features can come from the same image, hence a matching between them will not be possible.

$$P(i,j\,|\,g) = \begin{cases} P(i\,|\,g)P(j\,|\,g) & \text{if } I(f_i) \neq I(f_j), \\ 0 & \text{otherwise.}. \end{cases}$$

Therefore we need to embed this information in matrix $\hat{Z}^{\star}_{\hat{d}}$ for each pair of features. To do so we propose two alternatives: a first approach consists in setting

$$\hat{z}^{\star}_{\hat{d},ij} = \max_{g \in G_{ij}} P(i,j\,|\,g)\,,$$

being G_{ij} the set of games which contain the putative match (i,j). Alternatively, we can also incorporate further information about the reliability of the multi-feature track obtained from game g described as posterior probability $P(g)$—which is going to be introduced later on—so that it can be reformulated as follows

$$\hat{z}^{\star}_{\hat{d},ij} = \frac{\max_{g \in G_{ij}} P(i,j\,|\,g)P(g)}{\max_g P(g)}\,.$$

Finally apply MATCHEIG to matrix $\hat{Z}_{\hat{d},ij}$ thus obtained matrix $\hat{Z}^{\star}_{\hat{d},ij}$ fixing possible consistency errors by spectral relaxation as described in [31].

Subsequently, the last step consists in estimating the absolute permutation matrix—and therefore the set of multi-feature tracks—from the information provided by the relative permutation matrix obtained combined with the game posteriors. At the end extract a valid permutation matrix from both relative and absolute matrix applying `matrix2perm()` block-wise.

4.2 Game priors

Being able to compute sets of features does not ensure us that a particular one is valid and not redundant. The validity of a multi-feature set can be estimated from the convergence rate of the population vector in reaching the ESS, mostly because of incompatibility of features belonging to the hypothesis set. To correctly evaluate the reliability of a multi-feature set obtained from a particular game we consider the excepted payoff as in [15] to evaluate the mutually compatibility of the current population and also a weighting factor, so having

$$P(g) \propto w_g \cdot \mathbf{x}^T \Pi \mathbf{x} \sim Be(\cdot) \qquad \forall g\,,$$

where w_g represents the fact of having as number of features belonging to a certain track no more than the number of images whose features are been included in the hypothesis set. The higher is the number of features in the resulting set, the lower such weight is, as described by the following formula.

$$w_g = \begin{cases} 1 & \text{if } (\max_{i' \in g} x_{i'})^{-1} \leq n_g \\ \frac{|H| - \left(\max_{i' \in g} x_{i'}\right)^{-1}}{|H| - n_g} & \text{otherwise} \end{cases}$$

Once obtained what is intended to be a score associated to one game some normalization can be applied to each of them (e.g., considering the maximum one with certain probability).

4.3 Game posteriors

The concept of game posterior is intended to better evaluate the information retrieved from the game priors. Whereas game priors are obtained considering each single game independently, the idea of game posterior follows the idea of considering all games at once to evaluate one. The idea is inspired by the concept of "posterior equals likelihood times prior"

$$P(g \,|\, \text{data}) \propto P(\text{data} \,|\, g) P(g)\,.$$

Regarding the likelihood function, it needs to represent the maximum evidence that a certain feature is representative for a given multi-feature track, that is, no other track exists, among the ones already considered, such that they (probability-wise) contain it. The idea behind it is to exclude as much as possible multi-feature tracks that potentially contain only a subset of already considered tracks given a certain order. For that reason, consider an enumeration of all the games, hence obtaining the following

$$P_{g,1} = \max_{i \in g} \left\{ P(i \,|\, g) \prod_{g' < g} (1 - P(i \,|\, g)) \right\}, \tag{4.2a}$$

$$P_{g,2} = \max_{i \in g} \left\{ P(i \,|\, g) \prod_{g' > g} (1 - P(i \,|\, g)) \right\}. \tag{4.2b}$$

The reason of having two values in Eq. (4.2) is because of the potential presence of two possible sets of features in which one is a strict subset of the other while not excluding sets that happen to be roughly equivalent. In this way only the greater set of the two is preserved considering both ordering, as expressed by

$$P(\mathrm{data} \,|\, g) = \min(P_{g,1}, P_{g,2})\,.$$

4.4 Absolute permutation estimation

In this phase an attempt to estimate a set of multi-feature tracks is performed in terms of absolute permutation matrices, that means a matrix composed by the pairwise matches between features extracted from each image and their respected feature of the universe from a material point of the scene [31]. The way this is obtained is by means of the game posteriors and the population vector of each game. Firstly, all games with posterior lower than a certain threshold are excluded since they may be wrongfully associated to a multi-feature track that does not exist. The value $\hat{d}'$ is the number of games which posterior is above the threshold.

$$\hat{d}' = |\{g : P(g) \geq t\}|\,.$$

Secondly, a matrix $X_{\hat{d}'} \in \mathbb{R}^{M \times \hat{d}'}$ is created such that

$$[X_{\hat{d}'}]_{i,g} = P(i \,|\, g) = \frac{x_i}{\max_{i' \in g} x_{i'}} \qquad \forall g : P(g) \geq t\,,$$

given the row entry corresponding to feature i. Being $X_{\hat{d}'}$ composed of real values, finally obtain a set of valid permutation matrices by applying `matrix2perm()` [31] to each block thereof.

4.5 Discussion

Evolutionary MatchEIG is a technique presented in this thesis that combines multi-feature games and permutation synchronization by evaluating matching correspondences as probability estimations. Such estimations are firstly obtained from different multi-feature games in same way as [15] which each of them represents a potential multi-feature track. Each game can be seen as a population of feature that compose a certain track while finding an agreement in description similarity as to determine the track they belong to. The obtained multi-feature tracks are consequently discarded depending on their game posterior, filtered by thresholding. Game posteriors are computed as to evaluate each track truthfulness to represent a real and consistent multi-feature track. As a consequence, we are also able to find an estimate to the absolute permutation matrices which map features from one image to the corresponding track related to the material point taken from the scene.

Chapter 5

Experiments

All the experiments have been conducted in MATLAB, with the addition of VLFeat[1] and `export_fig`[2] libraries.

5.1 APAP Synchronization experiments

In order to assess the performance of APAP Synchronization we design the following experiments.

The procedure essentially requires the following parameters:

- `root_img_idx`: root image index taken from the image set enumeration;
- `ratio_test`: this parameter is used to improve the robustness of feature matching. Specifically, it is used to discard those matches that are ambiguous as the distance of a descriptor is closest match is similar to the distance with its second closest. The lower the value (expressed in percentage for the ratio between the first and second lower distances) this parameter is set the less matches are kept;
- `msac_max_distance`: MSAC outlier thresholding used for estimating the set of global homographies. A typical value for it is 1.5.
- `msac_max_distance_apap`: MSAC outlier thresholding used for discarding matches when performing the APAP estimations. This values is expected to be higher than the regular outlier thresholding for the global homographies. The reason behind this is to accommodate more matches which may not follow the global projective trend in the image

[1] `https://www.vlfeat.org/`

[2] `https://www.mathworks.com/matlabcentral/fileexchange/23629-export_fig`

stitching procedure. Because of this, a looser outlier thresholding value needs to be chosen, as for example of one order of magnitude higher than the regular one (e.g., 15, given 1.5 for regular global estimations);

- `avg_cell_width`, `avg_cell_height`: average image cell size (in pixels), used for computing the meshgrid of each image (except root). Depending on each image pixel size, the actual cell size is rounded off so to fit in an integer number of cell in the meshgrid. The higher these values the more resolute the APAP estimations are, trading off for computational time since it would involve a higher number of cells and thus synchronization nodes;

- `sigma`, `gamma`: for APAP weights computation (respectively for σ and γ). The scale parameter σ has to be resized depending on the average pixel resolution in images, while γ (a small value between 0 and 1) is used to regularize the warp by offsetting the APAP weights. A typical value for σ can be 30 on a set of 320×480 pixels and 0.025 can be a good value for γ.

The pipeline is composed of the following steps:

Parameters initialization

Set the aforementioned constants.

Feature extraction and matching

Extract SIFT feature from all the images and find the corresponding matching correspondences by performing MSAC algorithm, a variant of the Random Sample Consensus (RANSAC) algorithm.

Global homography estimation

Compute a set of global homographies through MSAC which are going to be used in case misalignments among cells occur.

Compute APAP matches

Same as done for the global homographies, but here instead a looser inlier threshold for MSAC has been used (namely, $\epsilon_{\text{APAP}} > \epsilon$).

APAP image pair stitching

Given any overlapping pair of images, obtain the target meshgrid and the (warped) source meshgrid. Subsequently, compute the adjacency matrix for all cell according to the overlapping and non-overlapping criteria.

APAP Synchronization

Apply the homography synchronization for the cells of all images. In this way we can exploit the consistency property among multiple measurements, thus reducing the error from misalignments.

Mosaic warping and rendering

Find the common reference frame from the root image and warp all the images for the final mosaic.

Figures 5.1 and 5.2 show an application of APAP Synchronization to a set of three overlapping images with different meshgrid cell sizes. As it can be noted, they produce no significant difference in the final mosaic in this particular example.

To quantify the alignment accuracy of the total set of image warps $f \colon \mathbb{R}^2 \to \mathbb{R}^2$, we adopt the method described in [53], we compute the root mean squared error (RMSE) of f on a set of keypoint matches $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^{N}$, i.e., $\mathrm{RMSE}(f) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|f(\mathbf{x}_i) - \mathbf{x}'_i\|^2}$. Further, for an image pair we randomly partitioned the available SIFT keypoint matches into a “training” and “testing” set. The training set is used to learn a warp, and the RMSE is evaluated over both sets.

We also employed the thresholding of Euclidean distance between a point in one image and its corresponding warped one in an overlapping image as error metric for finding outliers. For this purpose we used a threshold of 3 pixels for all datasets.

We compared APAP Synchronization with a baseline approach in which the images homographies are chained in a tree, thus not synchronized, connecting overlapping pairs of images in order to show the difference in the alignments on the same mosaicing task. The order in which images are estimated in the baseline approach (i.e., the construction of the tree) may affect the final alignment. On a side note, the tree is built so to maximize the total sum of matching points from pairs of images that are connected in the tree. In the end, this method just uses a subtree of the synchronization graph between over global homography. In addition, we extended this setting to pairwise APAP estimations used for comparison with the same error metrics. A further comparison is given by the homography synchronization of [40].

Dataset	APAP Synch			APAP w/o Synch			Synch w/o APAP			Baseline		
	TR	TE	%out	TR	TE	%out	TR	TE	%out	TR	TE	%out
Buildings	**2.23**	**3.16**	**4.92**	2.28	3.24	**4.92**	4.04	4.37	6.15	4.18	4.49	6.15
Buildings(2)	**2.67**	3.10	**4.56**	2.74	**3.01**	**4.56**	4.25	4.65	6.89	4.44	5.01	6.89
G. Earth	**1.80**	**1.94**	**0.81**	1.92	2.03	**0.81**	3.81	3.98	1.13	3.85	3.98	1.13

Table 5.1: Comparison of APAP Synchronization: for each method RMSE on both training set (TR) and test set (TE) has been reported with the total number of outlier matches (%out).

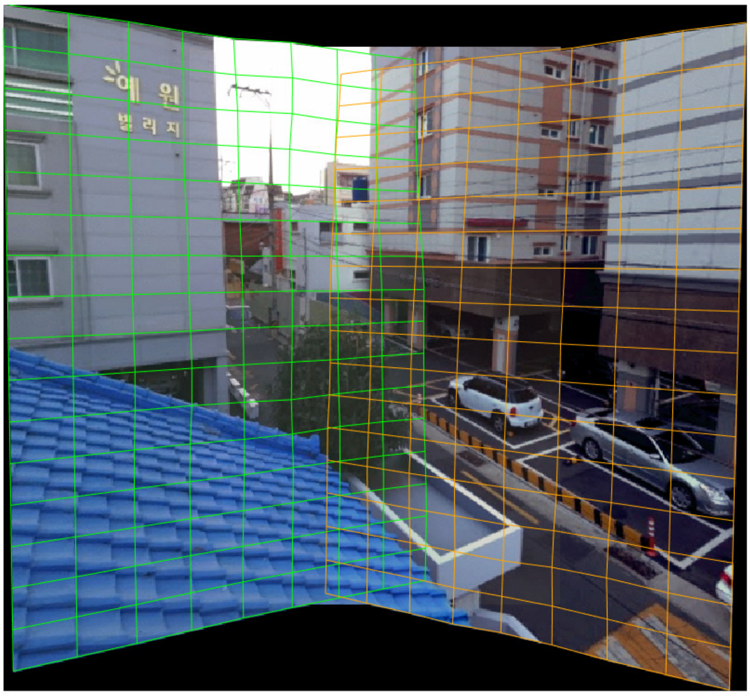


Figure 5.1: APAP Synchronization: example comprising three images, divided in 16x8 cells each on a dataset of images of 320x480 pixels.

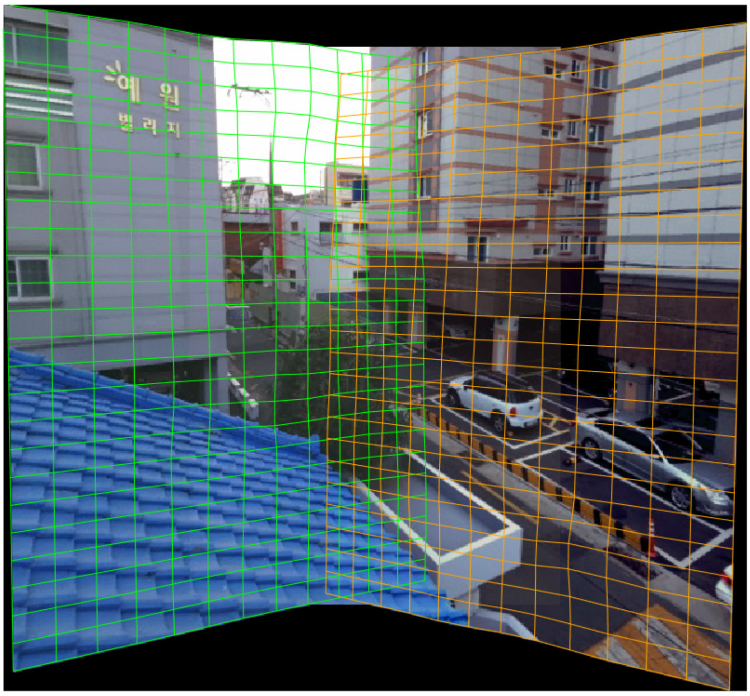


Figure 5.2: APAP Synchronization: example comprising three images, divided in 24x12 cells each on a dataset of images of 320x480 pixels.

5.2 Evolutionary MatchEIG experiments

The procedure essentially requires the following parameters:

- `N_qp`: number of query points, from each of which is generated a multi-feature matching game;
- `d_hat`: estimate $\hat{d}$ of the size of the universe set d. This is going to be used in the MatchEIG routine as in [31] and it has to be greater than the believed number of multi-feature tracks (it can be chosen to be same as `N_qp`);
- `k`: size of query point neighborhood. It is needed to evaluate the ball density of a query point when choosing the set of multi-feature games as in [15]. A typical value can be 4;
- `p`: size of hypothesis set, as a percentage of the total number of feature from all images. The higher this value the larger will be any hypothesis set for each game. A typical value is 20% of the total number of features;
- `sigma_a`: payoff function parameter. Earlier introduced as σ_a, it allows to model the payoff function so to be more or less selective when comparing the descriptor distances from two features. The closer to zero this value is the more selective the matching is;
- `thresh_Z`: threshold for discarding matches from relative permutation matrix when applying MatchEIG. Entry values in the relative permutation block-matrix $\hat{Z}^{\star}_{\hat{d}}$ lower than this threshold are set to zero;
- `thresh_game`: threshold for discarding multi-feature set whose game posterior is low. A typical value can be 0.5 (with normalized game posteriors values);
- `num_RD_iters`: number of iterations when applying the replicator dynamic equation to each game population. A value of 20 can be sufficient for reaching the ESS in the multi-feature selection games.

The pipeline is composed of the following steps:

Parameters initialization

Set the aforementioned constants.

Feature extraction

Extract SIFT feature from all the images.

Compute hypothesis sets

For each query point selected a group of feature which are going to the hypotheses of a certain game. Such feature are chosen so that they are close to the query point, while the total number of them (which is the same for all games) is set as a proportion, set by parameter p, of the total number of feature in all images.

Apply the replicator dynamics

Perform the population evolution through the replicator equation for fixed number of iteration.

Estimate relative permutations

Collect all the probabilities of relative matches in a block-matrix given each pair of images.

Apply MATCHEIG

Obtain a set of permutation matrix maximizing matching consistency. Set also to zero all the entries smaller than a given threshold `thresh_Z`.

Compute game priors and posteriors

Discard less reliable multi-feature sets whose game posterior is below a certain threshold.

Estimate absolute permutations

From the selected games and the related matching probabilities, find an estimate for the absolute permutation matrix of each image.

Figure 5.3a shows the matching correspondences of a pair of images as a result of Evolutionary MATCHEIG performed on a set of five images taken from Google Earth, while Fig. 5.3b shows an example of pairwise matching performed by means of RANSAC. As it can be noted, RANSAC is able to recover more feature correspondences than Evolutionary MATCHEIG given a sample pair of images.

In all the experiments, performances have been measured in terms of *precision* (number of correct matches returned divided by the number of matches returned and *recall* (number of correct matches returned divided by the number of correct matches that should have been returned). In order to provide a single figure of merit we computed the F_1 score (twice the product of precision and recall divided by their sum), which is a measure of accuracy and reaches its best value at 1 and worst at 0.

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}}\,, \quad \text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}\,, \quad F_1 = 2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}\,.$$

Dataset	Evolutionary MatchEIG		w/o MatchEIG		Multi-feature games ([15])		k-d trees	
	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
Graffiti	**94.5**	45.3	93.9	43.1	92.0	**97.3**	84.6	-
G. Earth	**92.9**	53.6	92.8	53.6	89.0	**95.6**	78.2	-

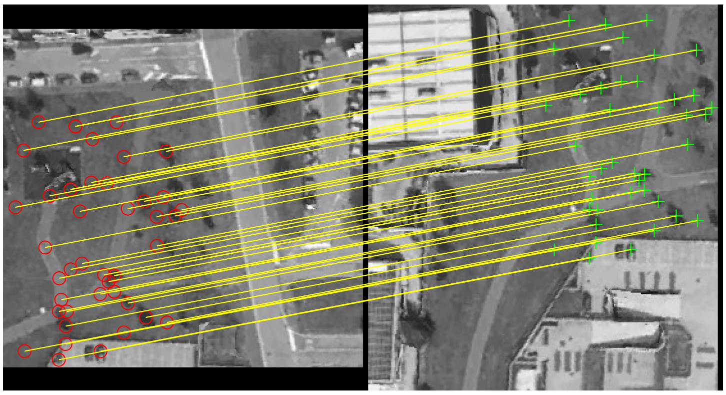
Table 5.2: Comparison from the experiment conducted reporting precision and recall. Note that the recall has not been reported for k-d trees since the correct matches obtained have been considered as the total number that should have been returned for all the other methods.

In order to estimate the precision we need to describe the criterion for considering an certain match correspondence correct. Matches are considered correct if the corresponding point is located within a given distance threshold from what is predicted. To have an idea of the recall we considered all the matches found by means of k-d trees and then polishing them through RANSAC from all pairs of images. This matches are going to be deemed as an approximation—being limited to the number of feature the SIFT algorithm can find—of the set of relevant matches that should have been returned by the proposed method.

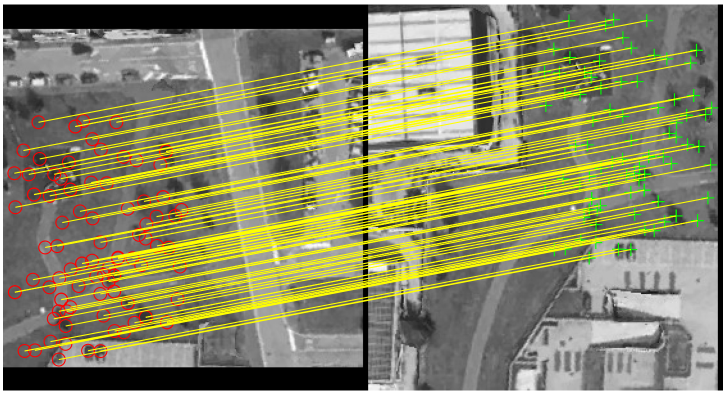
5.2.1 Synthetic experiments

Besides using real images, experiments have been also carried from synthetic data. For creating synthetic data, we set the following constants:

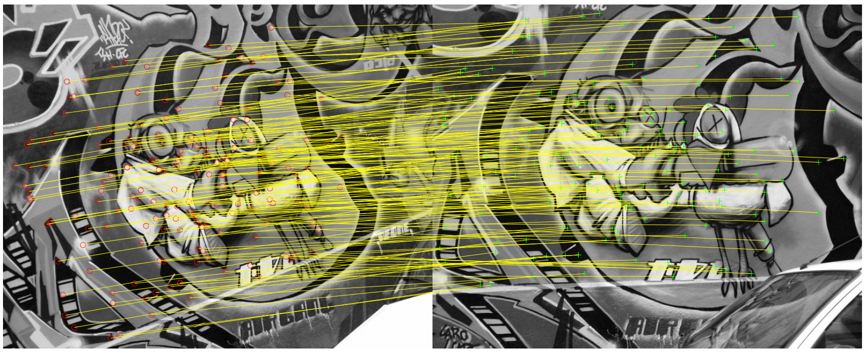
- `n`: number of views, namely the number of nodes in the permutation synchronization problem;
- `d`: size of the universe set. Note that we also use constant `d_hat` to represent the upper bound estimation for d;
- `obs_ratio`: observation ratio, i.e., the probability that a feature is seen in a view;
- `err_rate`: input error (ratio of observations corrupted), i.e., the ratio of mismatches in the relative permutations by switching two matches, removing true matches or adding false ones;
- `sigma_DD`: descriptor variance of matching pairs, for generating random descriptor distance out of a normal distribution (generally lower than `sigma_a`).



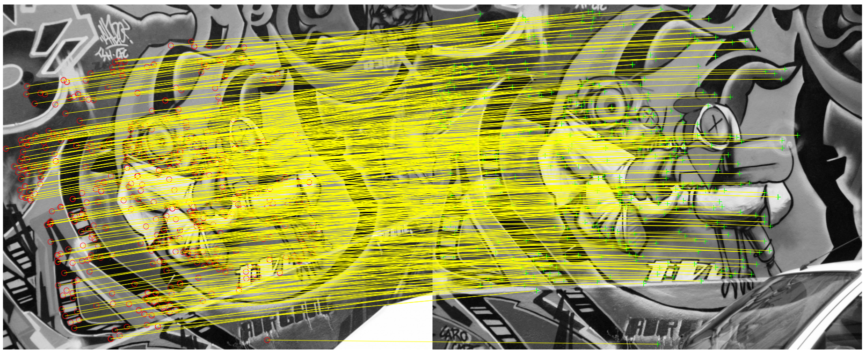
(a) Evolutionary MatchEIG *pair matching example on Google Earth images*



(b) RANSAC pair matching example on Google Earth images



(a) Evolutionary MatchEIG *pair matching example on Graffiti dataset*



(b) RANSAC pair matching example on Graffiti dataset

Firstly a random ground-truth absolute and relative permutation matrix is generated which satisfies the consistency constraint. Secondly, for each query point, random hypothesis sets are computed. The hypothesis sets, each corresponding to one game, are generated by grouping all features indices belonging to a one multi-track feature from the ground-truth absolute matrix.

Concerning the payoff functions, a matrix with the same size of the ground-truth permutation matrix is created; each entry of such matrix contains a randomly generated descriptor distance:

- if the feature pair corresponds to a match in the ground-truth matrix, their descriptor distance is generated from a zero-mean normal distribution of variance σ_{DD} (in absolute value);
- otherwise the distance is obtained from a uniform distribution ranged in $(0, u)$ where $u = \frac{3|H|\sigma_{DD}}{r_o n}$ (being r_o the observation ratio).

In order to analyze the effectiveness of the proposed method using the synthetic data, we evaluate the F_1 score alongside the precision and recall indices which in turn are obtained by the total number of true/false positive and true/false negative, being the problem a two-class classification, i.e., deciding if every entry in the relative permutation block-matrix is a match or not from the ground-truth matrix. We show that this evaluation can be performed when estimating both the relative permutations and the absolute ones. We recall that the the relative permutations matrices can be obtained back from the absolute ones by matrix multiplication as shown in Eq. (E.3).

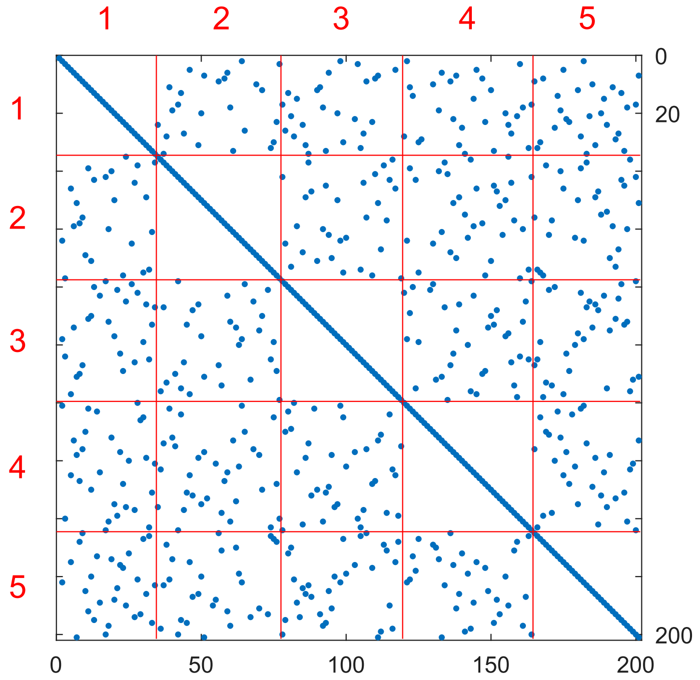


Figure 5.5: Relative permutation block-matrix resulted from Evolutionary MatchEIG *on synthetic data composed of 5 nodes, which represent images in real dataset. Every blue dot in each block represents a matching corresponding to the relative pair of features belonging to their respective images. Along the diagonal it consists of all the identity matrices representing the reflexive self-matching of features belonging to the same node.*

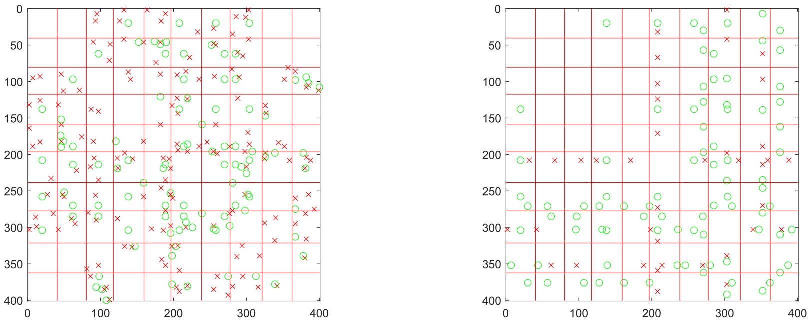
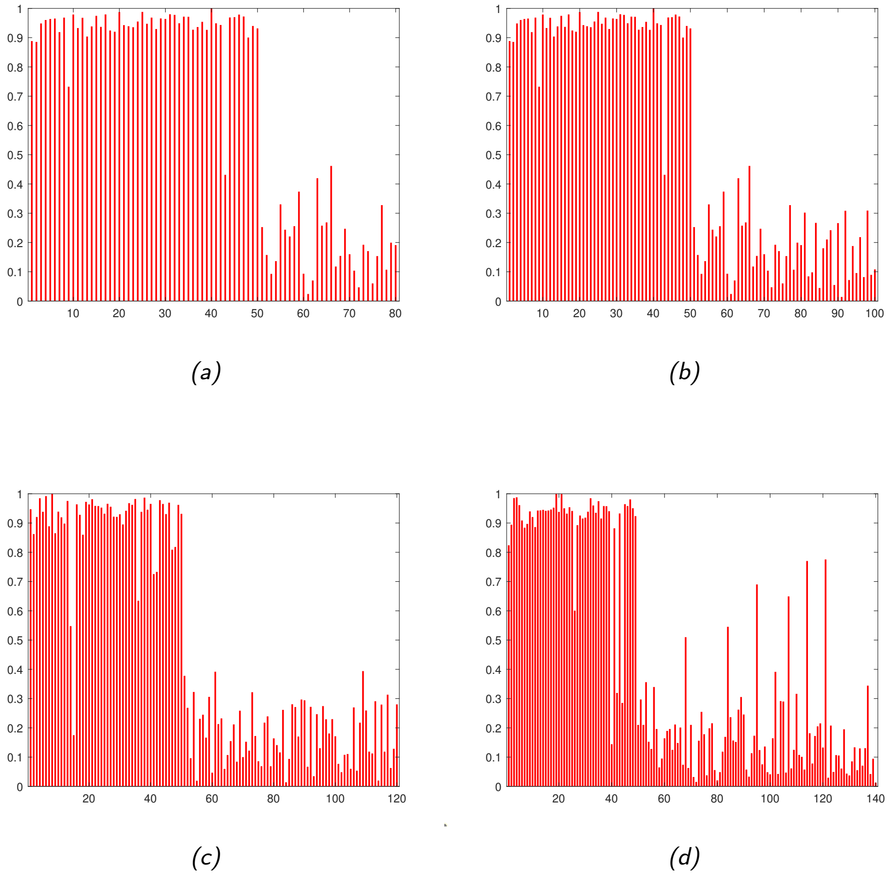


Figure 5.6: False positives (crosses) and false negative (circles) compared with ground-truth of the relative permutation block-matrix. Left from obtaining directly the relative permutation block-matrix, right from reassembling it from absolute permutations. The synthetic data here is generated with 10 nodes. A quick comparison of the two figures shows that the reassembling of the relative permutation block-matrix from the absolute one lead to more accurate results.



(a)

(b)

(c)

(d)

Figure 5.7: Game posteriors: results of $d = 50$ *(leftmost) ground-truth universe size (leftmost posterior values) on a total of* $\hat{d}$ *number of games taken from the grid of values* $[80\ \ 100\ \ 120\ \ 140]$ *(resp. (a), (b), (c), (d)). Values have been normalized by the maximum. It can be noted that the game posterior selection with a threshold of* $t = 0.5$ *is able to select most part of the 50 leftmost correct multi-feature tracks generated from synthetic data so to assess its robustness to outlier games (i.e., partially, or in no way, resembling a complete multi-feature track).*

5.3 Final Considerations

There is still room for improvements: as the experiments show, Evolutionary MatchEIG fails to recover a considerable part of multi-feature tracks which otherwise would be obtainable by simply performing RANSAC for each pair of images. Consequently, APAP Synchronization, that needs dense and distributed feature keypoints in overlapping regions, may in turn not be able to find correct estimates of local warping alignments. When APAP Synchronization still produces incorrect homographies the algorithm relies on a global homography.

After conducting the experiments, we observed that the current implementation for eigenvector decomposition (tested in MATLAB) fails to compute the result in some cases, undermining the reliability of the proposed algorithm on a generic set of image samples. This problem is due to the limitation of eigendecomposition algorithms in presence of noisy input matrices. APAP Synchronization allows to trade off between alignment granularity and computational time by leveraging on scalability of cell size.

In addition, we say that the estimated size of the universe $\hat{d}$ needs to be carefully selected due to it being an upper bound of the unknown actual value d, which represented the total number of multi-feature tracks from all images. We noticed that using a too large estimation may lead to returning too many wrongly evaluated multi-feature sets, i.e., that they do not represent an actual track, despite thresholding of game posteriors. For that reason, we chose to select a lower value than $\hat{d}$ as number of query points, but still using $\hat{d}$ when performing MatchEIG.

Despite all the challenges, both the presented methods can achieve reasonable performance on the provided datasets thanks to the averaging procedure provided by the synchronization of noisy measurements.

Chapter 6

Conclusions & Future Directions

The work presented in this thesis focuses on the challenge of both finding feature correspondences among a set of images taken from the same scene and stitching overlapping image region together so to obtain a mosaic. The inspiration taken for both ideas derives from the concept of averaging sets of noisy pairwise measures, being a common factor on both APAP Synchronization and Evolutionary MATCHEIG. The former aims to find a consistent agreement among as-projective-as-possible warps from a set of overlapping images corresponding to views that may not differ purely by rotation. The latter, instead, attempts to reconstruct feature tracks related to the same material point from a certain scene, hence resulting in solving the problem known as of multi-view matching. Solving this problem involves retrieving pairwise feature correspondences that needs to be consistent in terms of cycle consistency and descriptor similarity.

Some of the paths that can be explored for further development are finding a solution to the eigenvector decomposition problem in case of noisy data when performing APAP Synchronization, as well as the possibility of leveraging on geometrical information about the images taken from the scene during the process of multi-view matching in Evolutionary MATCHEIG. A further possibility is essentially to join the pipeline which through matching data coming from Evolutionary MATCHEIG go to APAP Synchronization. In other words, the objective is to exploit some additional information about the correspondences retrieved by Evolutionary MATCHEIG to be passed to APAP Synchronization so to better adapt the image stitching procedure.

Bibliography

[1] Y. I. Abdel-aziz and H. Karara. "Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry". In: *Photogrammetric Engineering and Remote Sensing* 81 (1971), pp. 103–107.

[2] A. Albarelli et al. "Matching as a non-cooperative game". In: *2009 IEEE 12th International Conference on Computer Vision*. 2009, pp. 1319–1326. DOI: `10.1109/ICCV.2009.5459312`.

[3] Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello. "Fast and accurate surface alignment through an isometry-enforcing game". In: *Pattern Recognition* 48.7 (2015), pp. 2209–2226. ISSN: 0031-3203. DOI: `https://doi.org/10.1016/j.patcog.2015.01.020`. URL: `http://www.sciencedirect.com/science/article/pii/S0031320315000394`.

[4] Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello. "Imposing Semi-Local Geometric Constraints for Accurate Correspondences Selection in Structure from Motion: A Game-Theoretic Perspective". In: *International Journal of Computer Vision* 97 (Mar. 2012), pp. 36–53. DOI: `10.1007/s11263-011-0432-4`.

[5] Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello. "Loosely Distinctive Features for Robust Surface Alignment". In: Sept. 2010, pp. 519–532. ISBN: 978-3-642-15554-3. DOI: `10.1007/978-3-642-15555-0_38`.

[6] Alex M. Andrew. "Shape from Shading, edited by Berthold K.P. Horn and Michael J. Brooks MIT Press, Cambridge, Mass., 1989, 577pp. (£49.50)". In: *Robotica* 8.3 (1990), pp. 263–264. DOI: `10.1017/S0263574700000242`.

[7] M. Arie-Nachimson et al. "Global Motion Estimation from Point Matches". In: *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*. 2012, pp. 81–88. DOI: `10.1109/3DIMPVT.2012.46`.

[8] F. Arrigoni, A. Fusiello, and B. Rossi. "Camera Motion from Group Synchronization". In: *2016 Fourth International Conference on 3D Vision (3DV)*. 2016, pp. 546–555. DOI: 10.1109/3DV.2016.64.

[9] Federica Arrigoni, Eleonora Maset, and Andrea Fusiello. "Synchronization in the Symmetric Inverse Semigroup". In: Oct. 2017, pp. 70–81. ISBN: 978-3-319-68547-2. DOI: 10.1007/978-3-319-68548-9_7.

[10] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. "Spectral Synchronization of Multiple Views in SE(3)". In: *SIAM Journal on Imaging Sciences* 9 (Jan. 2016), pp. 1963–1990. DOI: 10.1137/16M1060248.

[11] Brown and Lowe. "Recognising panoramas". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. 2003, 1218–1225 vol.2. DOI: 10.1109/ICCV.2003.1238630.

[12] Matthew Brown and David Lowe. "Automatic Panoramic Image Stitching using Invariant Features". In: *International Journal of Computer Vision* 74 (Aug. 2007), pp. 59–73. DOI: 10.1007/s11263-006-0002-3.

[13] Peter J. Burt and Edward H. Adelson. "A Multiresolution Spline with Application to Image Mosaics". In: *ACM Trans. Graph.* 2.4 (Oct. 1983), pp. 217–236. ISSN: 0730-0301. DOI: 10.1145/245.247. URL: https://doi.org/10.1145/245.247.

[14] Yuxin Chen, Leonidas Guibas, and Qixing Huang. "Near-Optimal Joint Object Matching via Convex Relaxation". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, 2014, II–100–II–108.

[15] L. Cosmo et al. "A game-theoretical approach for joint matching of multiple feature throughout unordered images". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. Dec. 2016, pp. 3715–3720. DOI: 10.1109/ICPR.2016.7900212.

[16] Ingemar J. Cox et al. "A Maximum Likelihood Stereo Algorithm". In: *Computer Vision and Image Understanding* 63.3 (1996), pp. 542–567. ISSN: 1077-3142. DOI: https://doi.org/10.1006/cviu.1996.0040. URL: http://www.sciencedirect.com/science/article/pii/S1077314296900405.

[17] J. Davis. "Mosaics of scenes with moving objects". In: *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*. 1998, pp. 354–360. DOI: 10.1109/CVPR.1998.698630.

[18] Paul E. Debevec and Jitendra Malik. "Recovering High Dynamic Range Radiance Maps from Photographs". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 369–378. ISBN: 0897918967. DOI: 10.1145/258734.258884. URL: https://doi.org/10.1145/258734.258884.

[19] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://doi.org/10.1145/358669.358692.

[20] R. I. Hartley. "In defense of the eight-point algorithm". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.6 (1997), pp. 580–593. DOI: 10.1109/34.601246.

[21] Heung-Yeung Shum and R. Szeliski. "Construction and refinement of panoramic mosaics with global and local alignment". In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 953–956. DOI: 10.1109/ICCV.1998.710831.

[22] Berthold Horn. "Determining Lightness from Image". In: *Computer Graphics and Image Processing* 3 (Dec. 1974), pp. 277–299. DOI: 10.1016/0146-664X(74)90022-7.

[23] Berthold Klaus Paul Horn, ed. *Robot Vision*. Cambridge, MA, USA: MIT Press, 1986. ISBN: 0262081598.

[24] Qixing Huang and Leonidas Guibas. "Consistent Shape Maps via Semidefinite Programming". In: *Computer Graphics Forum* 32 (Aug. 2013). DOI: 10.1111/cgf.12184.

[25] B. Ibragimov et al. "A Game-Theoretic Framework for Landmark-Based Image Segmentation". In: *IEEE Transactions on Medical Imaging* 31.9 (2012), pp. 1761–1776. DOI: 10.1109/TMI.2012.2202915.

[26] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97. DOI: https://doi.org/10.1002/nav.3800020109. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

[27] L. Li et al. “Optimal seamline detection for multiple image mosaicking via graph cuts”. In: *Isprs Journal of Photogrammetry and Remote Sensing* 113 (2016), pp. 1–16.

[28] David Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60 (Nov. 2004), pp. 91–. DOI: 10.1023/B:VISI.0000029664.99615.94.

[29] Steve Mann and Rosalind Picard. “On being ’undigital’ with digital cameras: extending dynamic range by combining differently exposed pictures”. In: *Proc. IS&T’s 48th Annual Conference* 48 (Mar. 1996).

[30] R. Marzotto, A. Fusiello, and V. Murino. “High resolution video mosaicing with global alignment”. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1. 2004, pp. I–I. DOI: 10.1109/CVPR.2004.1315099.

[31] E. Maset, F. Arrigoni, and A. Fusiello. “Practical and Efficient Multiview Matching”. In: *2017 IEEE International Conference on Computer Vision (ICCV).* Oct. 2017, pp. 4578–4586. DOI: 10.1109/ICCV.2017.489.

[32] T. Mitsunaga and S. K. Nayar. “Radiometric self calibration”. In: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149).* Vol. 1. 1999, 374–380 Vol. 1. DOI: 10.1109/CVPR.1999.786966.

[33] M. Oliveira, A. D. Sappa, and V. Santos. “Unsupervised local color correction for coarsely registered images”. In: *CVPR 2011.* 2011, pp. 201–208. DOI: 10.1109/CVPR.2011.5995658.

[34] Deepti Pachauri, Risi Kondor, and Vikas Singh. “Solving the multiway matching problem by permutation synchronization”. In: *Advances in Neural Information Processing Systems.* Ed. by C. J. C. Burges et al. Vol. 26. Curran Associates, Inc., 2013, pp. 1860–1868. URL: https://proceedings.neurips.cc/paper/2013/file/3df1d4b96d8976ff5986393e8767f5b2-Paper.pdf.

[35] Patrick Pérez, Michel Gangnet, and Andrew Blake. “Poisson Image Editing”. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 313–318. ISSN: 0730-0301. DOI: 10.1145/882262.882269. URL: https://doi.org/10.1145/882262.882269.

[36] American Society for Photogrammetry et al. *Manual of Photogrammetry*. American Society of Photogrammetry. American Society of Photogrammetry, 1980. ISBN: 9780937294017. URL: https://books.google.it/books?id=1MoYAQAAIAAJ.

[37] Erik Reinhard et al. *High dynamic range imaging : acquisition, display, and image-based lighting*. Jan. 2006. ISBN: 978-0-12-585263-0.

[38] E. Rodolà et al. "A game-theoretic approach to deformable shape matching". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 182–189. DOI: 10.1109/CVPR.2012.6247674.

[39] Emanuele Rodolà et al. "A Scale Independent Selection Process for 3D Object Recognition in Cluttered Scenes". In: *International Journal of Computer Vision* 102 (Mar. 2013). DOI: 10.1007/s11263-012-0568-x.

[40] E. Santellani, Eleonora Maset, and Andrea Fusiello. "SEAMLESS IMAGE MOSAICKING VIA SYNCHRONIZATION". In: vol. IV-2. May 2018, pp. 247–254. DOI: 10.5194/isprs-annals-IV-2-247-2018.

[41] P. Schroeder et al. "Closed-form solutions to multiple-view homography estimation". In: *2011 IEEE Workshop on Applications of Computer Vision (WACV)*. Jan. 2011, pp. 650–657. DOI: 10.1109/WACV.2011.5711566.

[42] Guy L. Scott and H. Christopher Longuet-Higgins. "An Algorithm for Associating the Features of Two Images". In: *Proceedings: Biological Sciences* 244.1309 (1991), pp. 21–26. ISSN: 09628452. URL: http://www.jstor.org/stable/76644.

[43] A. Singer. "Angular synchronization by eigenvectors and semidefinite programming". In: *Applied and Computational Harmonic Analysis* 30.1 (2011), pp. 20–36. ISSN: 1063-5203. DOI: https://doi.org/10.1016/j.acha.2010.02.001. URL: http://www.sciencedirect.com/science/article/pii/S1063520310000205.

[44] R. Szeliski. "Image Alignment and Stitching". In: *Handbook of Mathematical Models in Computer Vision*. Ed. by Nikos Paragios, Yunmei Chen, and Olivier Faugeras. Boston, MA: Springer US, 2006, pp. 273–292. ISBN: 978-0-387-28831-4. DOI: 10.1007/0-387-28831-7_17. URL: https://doi.org/10.1007/0-387-28831-7_17.

[45] Richard Szeliski. "Image Alignment and Stitching: A Tutorial". In: *Foundations and Trends in Computer Graphics and Vision* 2 (Jan. 2006). DOI: 10.1561/0600000009.

[46] Quoc-Huy Tran et al. "In Defence of RANSAC for Outlier Rejection in Deformable Registration". In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 274–287. ISBN: 978-3-642-33765-9.

[47] B. Triggs et al. "Bundle Adjustment - A Modern Synthesis". In: *Workshop on Vision Algorithms*. 1999.

[48] Matt Uyttendaele, Ashley Eden, and Richard Szeliski. "Eliminating Ghosting and Exposure Artifacts in Image Mosaics." In: vol. 2. Jan. 2001, pp. 509–516. DOI: 10.1109/CVPR.2001.991005.

[49] Jörgen W. Weibull. *Evolutionary Game Theory*. MIT press, 1995.

[50] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. "Local color transfer via probabilistic segmentation by expectation-maximization". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 747–754 vol. 1. DOI: 10.1109/CVPR.2005.215.

[51] Robert J. Woodham. "Analysing images of curved surfaces". In: *Artificial Intelligence* 17.1 (1981), pp. 117–140. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(81)90022-9. URL: http://www.sciencedirect.com/science/article/pii/0004370281900229.

[52] W. Xu and J. Mulligan. "Performance evaluation of color correction approaches for automatic multi-view image and video stitching". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 263–270. DOI: 10.1109/CVPR.2010.5540202.

[53] J. Zaragoza et al. "As-Projective-As-Possible Image Stitching with Moving DLT". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. June 2013, pp. 2339–2346. DOI: 10.1109/CVPR.2013.303.

[54] Zhengyou Zhang. "Parameter estimation techniques: a tutorial with application to conic fitting". In: *Image and Vision Computing* 15.1 (1997), pp. 59–76. ISSN: 0262-8856. DOI: https://doi.org/10.1016/S0262-8856(96)01112-2. URL: http://www.sciencedirect.com/science/article/pii/S0262885696011122.

[55] X. Zhou, M. Zhu, and K. Daniilidis. "Multi-image Matching via Fast Alternating Minimization". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 4032–4040. DOI: 10.1109/ICCV.2015.459.

Appendix A

2D Projective Warping

Let and $\mathbf{x} = [\,x\ \ y\,]^T$ and $\mathbf{x}' = [\,x'\ \ y'\,]^T$ be matching coordinate points across overlapping images I and I'. A projective warp transforms $\mathbf{x}$ to $\mathbf{x}'$ following the relation

$$\tilde{\mathbf{x}}' \propto H\tilde{\mathbf{x}}\,, \tag{A.1}$$

where $\tilde{\mathbf{x}} = [\,\mathbf{x}^T\ \ 1\,]$ is $\mathbf{x}$ in homogeneous coordinates, and $\propto$ indicates equality up to scale. The 3×3 invertible matrix H is called the *homography.* In inhomogeneous coordinates,

$$x' = \frac{\mathbf{r}_1[\,x\ \ y\ \ 1\,]^T}{\mathbf{r}_3[\,x\ \ y\ \ 1\,]^T} \quad \text{and} \quad y' = \frac{\mathbf{r}_2[\,x\ \ y\ \ 1\,]^T}{\mathbf{r}_3[\,x\ \ y\ \ 1\,]^T}\,, \tag{A.2}$$

where $\mathbf{r}_j$ is the j-th row of H. The division in (A.2) cause the 2D function to be non-linear, which is crucial to allow a fully perspective warp.

Direct Linear Transformation (DLT) [54] is the baseline method to estimate H from a set of noisy point matches $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ across I and I' (e.g., established using SIFT matching [28]). First, (A.1) is rewritten as the implicit condition $\mathbf{0}_{3\times1} = \tilde{\mathbf{x}}' \times H\tilde{\mathbf{x}}$ and then linearized as

$$\mathbf{0}_{3\times1} = \begin{bmatrix} \mathbf{0}_{1\times3} & -\tilde{\mathbf{x}}^T & y'\tilde{\mathbf{x}}^T \\ \tilde{\mathbf{x}}^T & \mathbf{0}_{1\times3} & -x'\tilde{\mathbf{x}}^T \\ -y'\tilde{\mathbf{x}}^T & x'\tilde{\mathbf{x}}^T & \mathbf{0}_{1\times3} \end{bmatrix} \mathbf{h}, \quad \mathbf{h} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}\,, \tag{A.3}$$

where $\mathbf{h}$ is obtained by vectorizing H into a vector. Only two of the rows in (A.3) are linearly independent. Let $\mathbf{a}_i$ be the first-two rows of the LHS matrix in (A.3) computed for the i-th point match $\{\mathbf{x}_i, \mathbf{x}'_i\}$. The quantity $\|\mathbf{a}_i\mathbf{h}\|$ is the *algebraic error* of the i-th datum from an estimate of $\mathbf{h}$. DLT minimizes the sum of squared algebraic errors

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\arg\min} \sum_{i=1}^{N} \|\mathbf{a}_i\mathbf{h}\|^2 \quad \text{s.t.} \quad \|\mathbf{h}\| = 1\,,$$

where the norm constraint prevents the trivial solution. DLT is thus also referred to as algebraic least squares [54]. Stacking vertically $\mathbf{a}_i$ for all i into matrix $A \in \mathbb{R}^{2N\times 9}$, the problem can be rewritten as

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} \|A\mathbf{h}\|^2 \quad \text{s.t.} \quad \|\mathbf{h}\| = 1\,.$$

The solution is the least significant right singular vector of A. Given the estimated H (reconstructed from $\hat{\mathbf{h}}$), to align the images, an arbitrary pixel $\mathbf{x}_*$ in the source image I is warped to the target image I' by

$$\tilde{\mathbf{x}}'_* \propto H\tilde{\mathbf{x}}_*\,, \tag{A.4}$$

while the position $\mathbf{x}'_*$ can be obtained by converting $\tilde{\mathbf{x}}'_*$ in cartesian coordinates. To avoid issues with numerical precision, prior to DLT the data can first be normalized in the manner of [20], with the estimated H then denormalized before executing (A.4).

Appendix B

RANSAC

The *RANdom SAmple Consensus* (RANSAC) algorithm proposed by Fischler and Bolles [19] is a general parameter estimation approach designed to cope with a large proportion of outliers in the input data. Unlike many of the common robust estimation techniques such as M-estimators and least-median squares that have been adopted by the computer vision community from the statistics literature, RANSAC was developed from within the computer vision community.

RANSAC is a resampling technique that generates candidate solutions by using the minimum number observations (data points) required to estimate the underlying model parameters. As pointed out by Fischler and Bolles [19], unlike conventional sampling techniques that use as much of the data as possible to obtain an initial solution and then proceed to prune outliers, RANSAC uses the smallest set possible and proceeds to enlarge this set with consistent data points [19]. Formally we introduce the *consensus set* given an inlier threshold ϵ and a parameter vector θ as

$$\mathrm{CS}_\epsilon(\theta) = \{x \in X \,|\, r(x,\theta) \leq \epsilon\} \,,$$

where $r(x,\theta)$ is the residual error of the model obtained from θ at point x. In conclusion the larger the consensus set $\mathrm{CS}_\epsilon(\theta)$ the better the model obtained from θ. This process is repeated through different iterations by randomly sampling from the complete dataset the minimum number of points required to determine the model parameters, aiming to maximize the number of consensus points.

The number of iterations, N, is chosen high enough to ensure that the probability p (usually set to 0.99) that at least one of the sets of random

Algorithm 1: RANSAC

```
Input: $X$ data, $\epsilon$ inlier threshold
Output: $\theta^*$ model parameter estimate
$i = 0$, $\mathrm{CS}^* = \{\}$, $N = +\infty$
repeat
    Select randomly a minimal sample set $S \subset X$ of size $m$
    Estimate parameters $\theta$ on $S$
    Evaluate consensus set $\mathrm{CS}_\epsilon(\theta)$
    if $|CS_\epsilon(\theta)| > |CS^*|$ then
        $\theta^* = \theta$
        $\mathrm{CS}^* = \mathrm{CS}_\epsilon(\theta)$
        $\hat{v} = 1 - \frac{|\mathrm{CS}^*|}{|X|}$
        $N = \frac{\log(1-p)}{\log(1-(1-\hat{v})^m)}$
    $i$++
until $i < N$;
Re-estimate $\theta$ on $\mathrm{CS}^*$ (through Ordinary Least Squares)
```

samples does not include an outlier. Let u represent the probability of selecting an inlier and $v = 1 - u$ the probability of observing an outlier. In this case, N iterations of the minimum number of points denoted m are required, where

$$1 - p = (1 - u^m)^N$$

and thus with some manipulation,

$$N = \frac{\log(1-p)}{\log(1-(1-v)^m)} \; .$$

Appendix C

SIFT Features

The *scale-invariant feature transform* (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images [28]. For any object in an image, interesting points on the object can be extracted to provide a “feature description” of the object. Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving.

The detection and description of local image features can help in object recognition. The SIFT features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint. In addition to these properties, they are highly distinctive, relatively easy to extract and allow for correct object identification with low probability of mismatch. Such points usually lie on high-contrast regions of the image, such as object edges. They are relatively easy to match against a (large) database of local features but, however, the high dimensionality can be an issue, and generally probabilistic algorithms such as *k-d trees* with best bin first search are used. Object description by set of SIFT features is also robust to partial occlusion; as few as 3 SIFT features from an object are enough to compute its location and pose. Recognition can be performed in close-to-real time, at least for small databases and on modern computer hardware.

The procedure computes at the end a descriptor vector of 128 elements for each keypoint such that the descriptor is highly distinctive and partially invariant to the remaining variations such as illumination, 3D viewpoint, etc.

SIFT feature matching can be used in image stitching for fully automated panorama reconstruction from non-panoramic images. The SIFT features extracted from the input images are matched against each other to find k nearest-neighbors for each feature. These correspondences are then used to find m candidate matching images for each image. Homographies between pairs of images are then computed using RANSAC and a probabilistic model is used for verification. Because there is no restriction on the input images, graph search is applied to find connected components of image matches such that each connected component will correspond to a panorama. Finally the panorama is rendered using multi-band blending. Because of the SIFT-inspired object recognition approach to panorama stitching, the resulting system is insensitive to the ordering, orientation, scale and illumination of the images. The input images can contain multiple panoramas and noise images (some of which may not even be part of the composite image), and panoramic sequences are recognized and rendered as output.

Appendix D

Permutations

Consider a set of n *nodes.* A set of m_i *objects* out of d is attached to node i (we say that the node "sees" these m_i objects) in a random order, i.e., each node has its own local labeling of the objects with integers in the range $\{1, \dots, n\}$. Let us also denote the set of d objects as *universe* set.

Pairs of nodes can *match* these objects, establishing which objects are the same in the two nodes, despite the different naming. The goal is to infer a global labeling of the objects, such that the same object receives the same label in all the nodes.

A more concrete problem statement can be given in terms of feature matching, where nodes are *images* and objects are *features.* A set of matches between pairs of images is given, and the goal is to combine them in a multi-view matching, such that each feature has a unique label in all the images.

Each matching is a *bijection* between (different) subsets of objects, which is also known as *partial permutation* (if the subsets are improper then the permutation is *total*). Total and partial permutations admit a matrix representation through permutation and partial permutation matrices, respectively.

A matrix P is said to be a permutation matrix if exactly one entry in each row and column is equal to 1 and all other entries are 0. A matrix P is said to be a partial permutation matrix if it has at most one nonzero entry in each row and column, and these nonzero entries are all 1. Specifically, the partial permutation matrix P representing the matching between node B and node A is constructed as follows: $[P]_{h,k} = 1$ if object k in node B is matched with object h in node A; $[P]_{h,k} = 0$ otherwise. If row $[P]_{h,\cdot}$ is a row of zeros, then object h in node A does not have a matching object in node B. If column $[P]_{\cdot,k}$ is a column of zeros, then object k in node B does not have a matching object in node A.

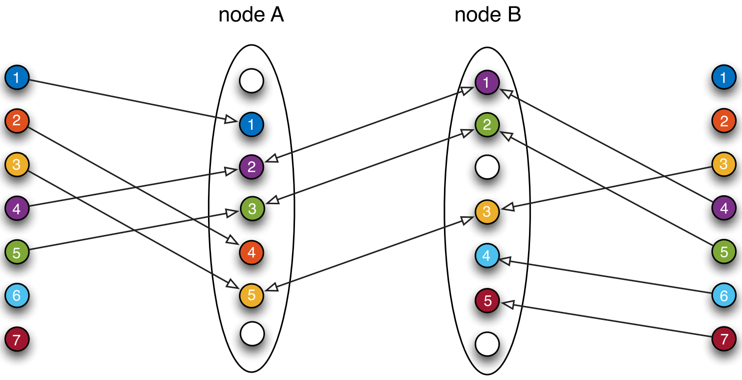


Figure D.1: In the center, two nodes with partial visibility match their three common objects. At the extrema the ground truth ordering of the objects. Each node sees some of the objects (white circles are missing objects) and puts them in a different order, i.e., it gives them different numeric labels.

The set of all $d \times d$ permutation matrices forms a group with respect to matrix multiplication, where the inverse is matrix transposition, which is called the *symmetric group* $\mathcal{S}_d$. The set of all $d \times d$ partial permutation matrices forms an inverse monoid with respect to the same operation, where the inverse is again matrix transposition, which is called the *symmetric inverse semigroup* $\mathcal{I}_d$.

Let $P_{ij} \in \mathcal{I}_d$ denote the partial permutation representing the matching between node j and node i, and let $P_i \in \mathcal{I}_d$ (resp. $P_j \in \mathcal{I}_d$) denote the unknown partial permutation that reveals the true identity of the objects in node i (resp. j) in the universe set. The matrix P_{ij} is called the *relative* permutation of the pair (i, j), and the matrix P_i (resp. P_j) is called the *absolute* permutation of node i (resp. j). It can be easily verified that

$$P_{ij} = P_i P_j^T \,. \tag{D.1}$$

Thus the problem of finding the global labeling can be modeled as finding n absolute permutations, assuming that a set of relative permutations is known, where the link between relative and absolute permutations is given by Eq. (D.1).

Appendix E

Group Synchronization

In a network of nodes, suppose that each node has an unknown state and that (noisy) measures of differences (or ratios) of states are available. The goal is to infer the unknown states from the available measures. This is a general statement of the *synchronization problem* [43]. Typically, states are represented by group elements, that is why the problem is actually referred to as *group synchronization.* Several instances of synchronization have been studied in the literature, which correspond to different instantiations of the considered group. Among them, it is worth citing SE(d) for *rigid-motion synchronization* [10], SL(d) for *homography synchronization* [41] and Aff(d) for *affine matrix synchronization.* Please note that SE(d), SL(d) and Aff(d) are all subgroups of GL(d). Here the attention is focused on synchronization over SL(3), that will be applied for image registration, over Aff(1) for color correction and over $\mathcal{I}_d$ for multi-view matching.

In order to formally define the problem and its solution, let Σ be a group and let $*$ denote its operation. Suppose that the pairwise relations between the index pairs $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$ are known, and refer to them as z_{ij}. *Synchronization* can be formulated as the problem of recovering $x_i \in \Sigma$ for $i = 1, \ldots, n$ such that the following *consistency constraint* is satisfied

$$z_{ij} = x_i * x_j^{-1} \,. \tag{E.1}$$

The solution is defined up to a global (right) product with any group element, i.e., if $x_i \in \Sigma$ satisfies (E.1) then also $x_i * y$ does for any (fixed) $y \in \Sigma$.

If the known pairwise measures are noisy, the consistency constraint cannot be satisfied exactly. Thus, as shown in Fig. 1, the searched solution is the one that minimizes the *consistency error*:

$$\epsilon(x_1, x_2, \ldots, x_n) = \sum_{(i,j)} \delta(z_{ij}, x_i * x_j^{-1}) \,,$$

where $\delta\colon \Sigma \times \Sigma \to \mathbb{R}^+$ is a metric function for Σ [8].

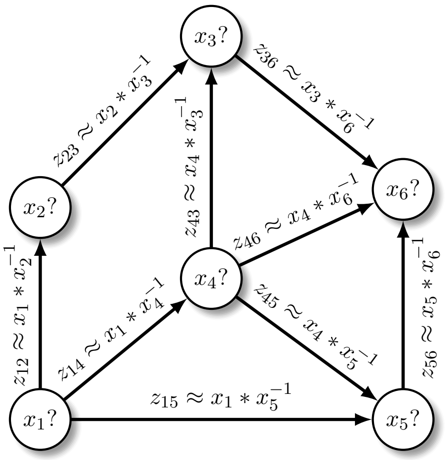

$x_3?$
$x_2?$
$x_6?$
$x_4?$
$x_1?$
$x_5?$
$z_{23} \approx x_2 * x_3^{-1}$
$z_{43} \approx x_4 * x_3^{-1}$
$z_{36} \approx x_3 * x_6^{-1}$
$z_{46} \approx x_4 * x_6^{-1}$
$z_{12} \approx x_1 * x_2^{-1}$
$z_{14} \approx x_1 * x_4^{-1}$
$z_{45} \approx x_4 * x_5^{-1}$
$z_{56} \approx x_5 * x_6^{-1}$
$z_{15} \approx x_1 * x_5^{-1}$

Figure E.1: The synchronization problem. Each node is characterized by an unknown state and measures on the edges are ratios of states. The goal is to compute the states that best agree with the measures.

E.1 Synchronization over $(\mathrm{GL}(d), \cdot)$

In this section we consider the synchronization problem over the General Linear Group $\mathrm{GL}(d)$, which is the set of all $d \times d$ invertible matrices, where the group operation $*$ is matrix multiplication and $1_\Sigma = I_d$. Let $X_i \in \mathbb{R}^{d \times d}$ and $Z_{ij} \in \mathbb{R}^{d \times d}$ denote the matrix representations of $x_i \in \Sigma$ and $z_{ij} \in \Sigma$, respectively. Using this notation, (E.1) rewrites $Z_{ij} = X_i X_j^{-1}$.

Let us collect the unknown group elements and all the measures in two matrices $X \in \mathbb{R}^{dn \times d}$ and $Z \in \mathbb{R}^{dn \times dn}$ respectively, which are composed of $d \times d$ blocks, namely

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix}, \quad Z = \begin{bmatrix} I_d & Z_{12} & \dots & Z_{1n} \\ Z_{21} & I_d & \dots & Z_{2n} \\ \dots & & & \dots \\ Z_{n1} & Z_{n2} & \dots & I_d \end{bmatrix}.$$

If not all the pairwise measures Z_{ij} are available, the input matrix becomes $Z_A := Z \circ (A \otimes \mathbf{1}_{d \times d})$, where $\circ$ denotes the Hadamard product, A is the adjacency matrix and the Kronecker product with $\mathbf{1}_{d \times d}$ is required to match the block structure of the measures. The $n \times n$ adjacency matrix is constructed as follows: $A_{ij} = 1$ if the pairwise measure Z_{ij} exists, $A_{ij} = 0$ otherwise. Accordingly, the consistency constraint writes

$$Z_A = (X X^{-\flat}) \circ (A \otimes \mathbf{1}_{d \times d}), \tag{E.2}$$

where $X^{-\flat} \in \mathbb{R}^{d \times dn}$ denotes the block-matrix containing the inverse of each $d \times d$ block of X.

It can be shown [7] that

$$Z_A X = (D \otimes I_d) X \,,$$

thus an estimate of X is represented by the eigenvectors of $(D \otimes I_d)^{-1} Z_A$ corresponding to the d largest eigenvalues, where D is the degree matrix defined as $D = \operatorname{diag}(A\mathbf{1}_{n\times 1})$. This is also called the *spectral* solution.

E.2 Synchronization over $\mathrm{SL}(d)$

Consider now the Special Linear Group $\mathrm{SL}(d)$, that is the set of $d \times d$ matrices with unit determinant

$$\mathrm{SL}(d) = \{ R \in \mathbb{R}^{d\times d} \quad \text{s.t.} \quad \det(R) = 1 \} \,.$$

Synchronization over SL(3) corresponds to the homography synchronization problem. Since $\mathrm{SL}(d)$ is a subgroup of $\mathrm{GL}(d)$, the problem can be addressed via the spectral solution, which computes the top d eigenvectors of $(D \otimes I_d)^{-1} Z_A$, that are collected in a $dn \times d$ matrix U. In order to obtain elements of $\mathrm{SL}(d)$ from U, each $d \times d$ block in U, denoted by U_i, must be scaled to unit determinant [41], which can be done by dividing U_i by $\sqrt[d]{\det(U_i)}$. However, if $\det(U_i)$ is negative and d is even, real roots do not exist; in this case the determinant can be always made positive by exchanging two columns of U.

E.3 Synchronization over $\mathrm{Aff}(d)$

Let us consider the Affine Group $\mathrm{Aff}(d)$, that is the set of invertible affine transformations in d-space, which admits a matrix representation through $(d+1) \times (d+1)$ matrices

$$\mathrm{Aff}(d) = \left\{ \begin{bmatrix} M & v \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \text{s.t.} \quad M \in \mathbb{R}^{d\times d}, \mathbf{v} \in \mathbb{R}^d \right\} \,.$$

$\mathrm{Aff}(d)$ is a subgroup of $\mathrm{GL}(d+1)$, therefore the synchronization problem can be solved by computing the top $d+1$ eigenvectors of $(D \otimes I_{d+1})^{-1} Z_A$. Since this approach leads to an algebraic solution, it does not enforce constraints that matrices in $\mathrm{Aff}(d)$ should satisfy.

Specifically, the output matrix U will not have vector $[\,\mathbf{0}_{1\times d} \;\, 1\,]$ in rows multiple of $d+1$. In order to recover X from U it is sufficient to choose a different basis for the resulting eigenvectors that satisfies such constraint, which can be found by taking a suitable linear combination of the columns of U, as explained in [10].

E.4 Synchronization over $\mathcal{S}_d$ and $\mathcal{I}_d$

Let us describe the synchronization problem over $\Sigma = \mathcal{S}_d$. Since $\mathcal{S}_d$ is a subgroup of $\mathrm{O}(d)$ and thus a subgroup of $\mathrm{GL}(d)$, permutation synchronization can be addressed with the matrix notation shown earlier. As observed in [34, 24, 55], the consistency constraint (D.1) can be expressed in a compact matrix form if all the absolute and relative permutations are collected in two block-matrices $X \in \{0,1\}^{m \times d}$ and $Z \in \{0,1\}^{m \times m}$ respectively, where $m = \sum_{i=1}^{n} m_i$, namely

$$X = \begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_n \end{bmatrix}, \quad Z = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \dots & & & \dots \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{bmatrix}.$$

For practical reasons we defined $P_{ij} \in \{0,1\}^{m_i \times m_j}$ for relative permutations and $P_i \in \{0,1\}^{m_i \times d}$ for absolute permutations. Note that Z may contain zero blocks: if all the features in image i do not match with any feature in image j, then $P_{ij} = 0$. Using this notation, Eq. (D.1) becomes

$$Z = XX^T. \tag{E.3}$$

Since Z has rank d, the matrix $V = X^T X$ contains the *largest* eigenvalues of Z and all the other eigenvalues are zero. Thus, in the presence of noise, we can take the eigenvectors of Z corresponding to the d largest eigenvalues as an estimate of X [31].

If permutations were total, Eqs. (D.1) and (E.3) would be recognized as the consistency constraint of a synchronization problem over $\mathcal{S}_d$ [34]. However, in all practical settings, permutations are partial, so in [9] they address the synchronization problem over the inverse monoid $\mathcal{I}_d$.

Consider now the synchronization problem over $\Sigma = \mathcal{I}_d$. Despite the group structure is missing, [31] shows that a spectral solution can be derived in an analogous way, which can be seen as the extension of [34] to the case of partial permutations. Moreover, the authors of [31] also propose an alternative method in terms of an optimization problem for multi-view matching.

E.4.1 Optimization problem

In practice, pairwise correspondences contain errors, hence what is being measured is an estimate $\hat{P}_{ij}$ of the relative permutation between image i and image j (here we use the hat accent to denote approximate quantities). The goal is to compute a set of partial permutation matrices $\{P_{ij}\}_{i,j=1}^{n}$ such that the consistency constraint is satisfied and P_{ij} is as close as possible to its measure $\hat{P}_{ij}$, namely $P_{ij} \approx \hat{P}_{ij}$ for all $i, j \in \{1, \dots, n\}$. A possible approach consists in considering the following optimization problem

$$\max_{\{P_{ij}\}_{i,j=1}^{n}} \sum_{i,j=1}^{n} \langle \hat{P}_{ij}, P_{ij} \rangle \quad \text{s.t.} \quad P_{ij} = P_i P_j^T \,, \tag{E.4}$$

where each optimization variable is constrained to be a partial permutation matrix. Here $\langle \cdot, \cdot \rangle$ denotes the matrix inner product, i.e. $\langle A, B \rangle = \operatorname{trace}(AB^T)$. The cost function in (E.4) counts, for each image pair (i, j), the number of features equally matched by permutations P_{ij} and $\hat{P}_{ij}$.

If $\hat{Z}$ denotes the block-matrix containing the measured relative permutations $\hat{P}_{ij}$ then Eq. (E.4) rewrites

$$\max_{Z} \langle \hat{Z}, Z \rangle = \max_{Z} \operatorname{trace}(\hat{Z} Z^T) \quad \text{s.t.} \quad Z = XX^T \tag{E.5}$$

$$\iff \max_{X} \langle \hat{Z}, XX^T \rangle = \max_{Z} \operatorname{trace}(X^T \hat{Z} X) \,, \tag{E.6}$$

where X is constrained to be composed of partial permutation matrices. Maximizing the objective function in Eq. (E.6) is a challenging task since the feasible set consists of binary variables which makes the problem combinatorially NP-hard. Moreover, optimizing with respect to multiple permutation matrices simultaneously increases the difficulty of the problem. For these reasons, it is common practice to relax some constraints on the optimization variables, thus providing tractable approaches that solve the multi-view matching problem approximately but efficiently. Some examples include the *semidefinite* relaxation [14], the *low-rank* relaxation [55] and the *spectral* relaxation [34].

Appendix F

Evolutionary Game Theory

Evolutionary game theory [49] considers a scenario where pairs of individuals, each pre-programmed with a given strategy, are repeatedly drawn from a large population to play a game, and a selection process allows "fit" individuals (i.e., those selecting strategies with high support) to thrive, while driving "unfit" ones to extinction. The general idea is to model each hypothesis as a strategy and let them be played one against the other according to a fixed payoff function until a stable population emerges. These notions of *hypothesis*, *payoff*, *population* are here described:

Hypothesis

A fact, derived from observed data, that is assumed to be produced by the phenomenon to be characterized. We define $H = \{1, \dots, n\}$ be the set of n available hypotheses derived from data.

Payoff

A measure of the degree of reciprocal support between two hypotheses. The payoff is usually expressed as a function $\pi \colon H \times H \to \mathbb{R}_{\geq 0}$. Since payoffs are defined between all the pairs, an alternative notation is the payoff matrix $\Pi = (\pi_{ij})$ where $\pi_{ij} = \pi(i, j)$, and i and j are hypotheses.

Population

A probability distribution $\mathbf{x} = (x_1, \dots, x_n)^T$ over the strategies H. Any population vector is bound to lie within the n-dimensional standard simplex $\Delta^n = \{\mathbf{x} \in \mathbb{R}^n : x_i \geq 0 \text{ for all } i = 1, \dots, n,\ \sum_{i=1}^n x_i = 1\}$. The support of a population $\mathbf{x} \in \Delta^n$, denoted by $\sigma(\mathbf{x})$, is defined as the set of elements chosen with non-zero probability: $\sigma(\mathbf{x}) = \{i \in H : x_i > 0\}$.

In order to find a set of mutually coherent hypotheses, we are interested in finding configurations of the population maximizing the average payoff.

Since the total payoff obtained by hypothesis i within a given population $\mathbf{x}$ is $(\Pi\mathbf{x})_i = \sum_j \pi_{ij}x_j$, the (weighted) average payoff over all the considered hypotheses is exactly $\mathbf{x}^T\Pi\mathbf{x}$.

Unfortunately, it is not immediate to find the global maximum for $\mathbf{x}^T\Pi\mathbf{x}$, however local maxima, called Evolutionary Stable State (ESS), can be obtained by letting an initial population vector $\mathbf{x}$ evolve by means of a rather wide class of evolutionary dynamics called *Payoff-Monotonic Dynamics.* In particular we use replicator dynamics which are governed by the following equation:

$$x_i(t+1) = x_i(t)\frac{(\Pi\mathbf{x}(t))_i}{\mathbf{x}(t)^T\Pi\mathbf{x}(t)}\,.$$