Projective Synchronization
IACV Project

Carlo Sgaravatti
Politecnico di Milano
carlo.sgaravatti@mail.polimi.it

Abstract

This project deals with the projective synchronization problem. The solution to the problem is obtained using a general approach called synchronization, whose goal is to estimate the values of some unknown states related to a network, given some measurements between pairs of nodes. The proposed methods extend the approaches present in the literature for the so-called homography synchronization problem.

1. Introduction

Given a set of projective frames, from which some local transformations between pairs of projective frames are known, in the projective synchronization the goal is to find for each of them a global transformation that represents it in a certain common reference frame.

In general, the problem of synchronization ([1]) is characterized by the presence of a network, in which each node has an unknown state that belongs to a group $(\Sigma, *)$. A set of measurements, typically corrupted by noise, between some pair of nodes is known, from which the goal is to find the value of the unknown states. Several methods have been proposed in the literature in order to solve synchronization problems for many settings. The main goal of this project is to study if it is possible to extend the methods used to perform homography synchronization ([3, 1, 2]) to the projective synchronization problem.

1.1. Homography synchronization

In the homography synchronization problem ([3]), the elements of the group $(\Sigma, *)$ are 2-dimensional homographies, namely 3x3 invertible matrices defined up to a scale.

A typical application of homography synchronization is the image mosaicing problem ([2]), described in Figure 1. Given a set of images, from which pairs of relative homographies are computed, the goal is to align the images into a single mosaic. In particular:

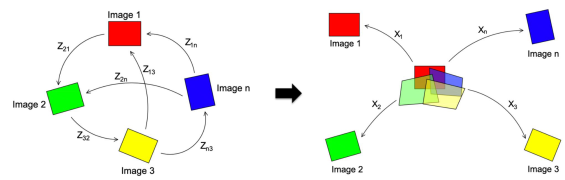

Figure 1. The image mosaicing problem.

- The edges of the graph are the local pairwise homographies between images
- The nodes of the graph are the global homographies that permit to align all the images into the mosaic by applying the inverse transformation

The synchronization approach permit solving the image mosaicing problem in a simple way. Indeed, it is possible to easily compute the pairwise homographies by using point correspondences with robust fitting algorithms (e.g. RANSAC). However, it is more challenging to compute the global transformations without reasoning in a broader way, as the synchronization approach does by considering the whole graph.

This project considers the projective synchronization problem, where the elements of the group $(\Sigma, *)$ are 3-dimensional projective transformations, namely 4x4 matrices defined up to a scale.

2. Related work

2.1. The synchronization framework

Formally, a synchronization problem ([1]) is represented by the notion of group-labelled graph, characterized by:

- a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which the nodes represent the unknown states and the edges represent the measurements between the nodes
- a group $(\Sigma, *)$ that represents the states of the graph

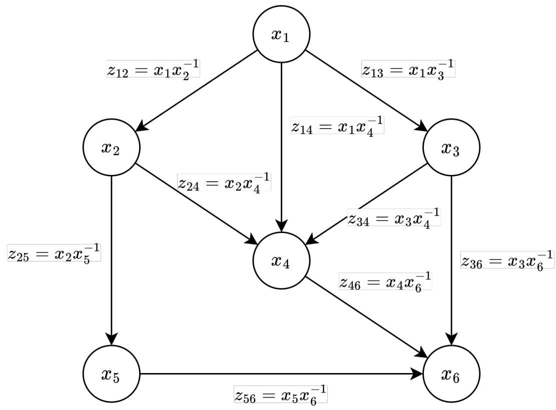

Figure 2. An example of consistent labelling.

- a labelling function $z : \mathcal{E} \rightarrow \Sigma$, mapping each edge of the graph to an element of the group, that represents the measurements between the nodes connected by the edge

In order to have a well-defined problem, the graph must be connected.

A solution to the problem is given by a vertex labelling, i.e. a function $x : \mathcal{V} \rightarrow \Sigma$ that maps each node of the graph to an element of the group, representing the state of the node. We say that vertex labelling is consistent when

$$z(e) = x(i) * x(j)^{-1}, \forall e = (i, j) \in \mathcal{E}. \tag{1}$$

Figure 2 shows an example of Σ-labelled graph with consistent labelling. This definition of consistent labelling is strictly connected to the null-cycle property, which states that the composition of the edge labels along a circuit $\{(i_1, i_2), (i_2, i_3), \dots, (i_l, i_1)\}$ should return the identity element 1_Σ:

$$z(i_1, i_2) * z(i_2, i_3) * ... * z(i_l, i_1) = 1_\Sigma. \tag{2}$$

In practice, the labels of the edges are affected by noise, thus it is not possible to ensure a consistent labelling of the nodes. Thus, the goal of the synchronization problem is to find a vertex labelling $\widetilde{x}$ that minimizes the following cost function, called consistency error:

$$\epsilon(\widetilde{x}) = \sum_{(i,j) \in \mathcal{E}} \rho(\delta(\widetilde{z}(i, j), z(i, j))) \tag{3}$$

where $\rho : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is a non-negative and non-decreasing function, $\delta : \Sigma \times \Sigma \rightarrow \mathbb{R}^+$ is a metric function, and $\widetilde{z}(i, j) = \widetilde{x}(i) * \widetilde{x}(j)^{-1}$ is the consistent labelling function induced by $\widetilde{x}$. It is possible to notice that, if x is a consistent labelling of the vertices, we can obtain another consistent labelling by defining a global right product with an arbitrary element $s \in \Sigma$, i.e. the vertex labelling y defined as $y(i) = x(i) * s$ is consistent. Thus, any solution to the problem has an ambiguity, i.e. it is defined up to the right product with an element of the group.

2.2. The spanning tree approach

An algorithm that can be used to solve the synchronization problem, in any setting (i.e. for any possible group $(\Sigma, *)$), is based on the computation of a minimum spanning tree from the starting graph. The algorithm performs the following steps:

1. Compute a minimum spanning tree $\mathcal{G}_{mst} = (\mathcal{V}, \mathcal{T})$ of the graph using a certain node $r \in \mathcal{V}$ as a root of the tree. Figure 3 shows an example of a minimum spanning tree of the previous graph, using node 1 as the root of the tree.

2. Assign an arbitrary element of the group to $x(r)$, e.g. $x(r) = 1_\Sigma$

3. Visit the tree in order starting from the root and exploit the consistency property in order to compute the other unknown states: $\forall (i, j) \in \mathcal{T}$

$$x(i) = x(j) * z(i, j)^{-1} = x(j) * z(j, i) \tag{4}$$

Considering that the graph is unweighted, the minimum spanning tree can be computed using a breadth-first tree approach.

This approach permits obtaining consistent labelling of the nodes in the noise-free case. On the other hand, if we have some noise, the consistency error might depend on the choice of the root node. More in detail, the propagation of the consistency property through the tree might result in bigger errors if the depth of the tree is high. This is why a very common choice for the root node is the node with the highest degree.

2.3. Synchronization over SL(3): Spectral Solution

The homography synchronization problem can be solved in closed form by recasting it into an eigenvector problem ([1], [3]). Let X_i be the unknown global homography representing node (image) $i \in \mathcal{V}$, and Z_{ij} the pairwise homography between images $i, j \in \mathcal{V}$. In order to solve the problem using the spectral solution, it is required to solve the scaling of the matrices. Thus, the pairwise homographies are normalized to have a unitary determinant; this is achieved by dividing each Z_{ij} matrix by the 3rd root of the determinant, $\sqrt[3]{det(Z_{ij})}$. In this way, the labels of the edges are now 3x3 invertible matrices with unitary determinant, namely $(\Sigma, *)$ is the Special Linear Group, SL(3). The next

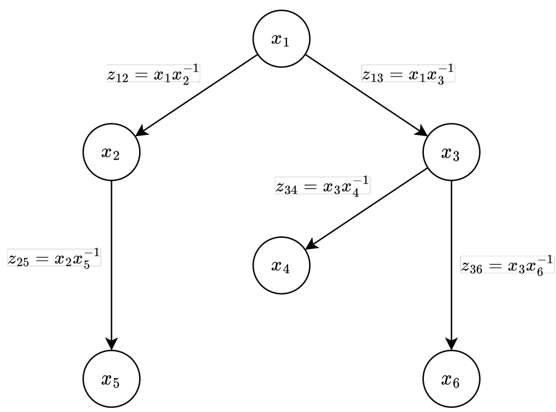

Figure 3. An example of minimum spanning tree.

step of the method is to collect all the labels in two matrices $X \in \mathbb{R}^{3n \times 3}$ and $Z \in \mathbb{R}^{3n \times 3n}$ (where $n = |\mathcal{V}|$ is the number of nodes of the graph), defined as follows:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_n \end{bmatrix}, \quad Z = \begin{bmatrix} I_3 Z_{12} \cdots Z_{1n} \\ Z_{21} I_3 \cdots Z_{2n} \\ \ldots\ldots \ddots \ldots \\ Z_{n1} Z_{n2} \ldots I_3 \end{bmatrix}. \tag{5}$$

When the graph is complete, the following equation represents the consistency constraint induced by the vertex labelling X:

$$Z = XX^{-b} \tag{6}$$

where

$$X^{-b} = \left[X_1^{-1} X_2^{-1} \cdots X_n^{-1} \right]. \tag{7}$$

If the graph is not complete, the consistency constraint can be defined using the matrix $Z_A = Z \circ (A \otimes 1_{3\times3})$ (where A is the adjacency matrix of the graph), which contains the available measurements, in the following way:

$$Z_A = (XX^{-b}) \circ (A \otimes 1_{3\times3}) \tag{8}$$

where $\circ$ is the Hadamard product and $\otimes$ is the Kronecker product. Thus, we can derive a consistency error in the following way:

$$\epsilon(X) = \| Z_A - (XX^{-b}) \circ (A \otimes 1_{3\times3}) \|_F^2 \tag{9}$$

It is possible to prove in the noise-free case that the columns of X are the eigenvectors of the matrix $(D \otimes I_3)^{-1} Z_A$ associated with the eigenvalue 1, where D is the degree matrix of the graph. In addition, matrix $(D \otimes I_3)^{-1} Z_A$ has real eigenvalues and the largest one is 1, with multiplicity 3. In the noisy case, an estimate of X can be computed from the 3 eigenvectors of the matrix $(D \otimes I_3)^{-1} Z_A$ associated with the largest eigenvalues. However, these may be complex and so the imaginary part needs to be removed.

As pointed out in Section 2.1, the obtained solution $U \in \mathbb{R}^{3n \times 3}$ has an ambiguity. This can be solved by scaling with respect to one of the nodes $v \in V$ of the graph, i.e. by multiplying each block U_i by U_v^{-1}. Due to the removal of the complex values and this last step, in order to obtain elements of the SL(3) group, the matrices U_i need to be divided by the 3rd root of their determinant.

3. Proposed approach

3.1. Spectral Solution Extension

Similar to the homography synchronization problem, for the projective synchronization problem we can define the matrices $X \in \mathbb{R}^{4n \times 4}$ and $Z \in \mathbb{R}^{4n \times 4n}$. One issue with respect to homography synchronization is represented by the fact that the measurements Z_{ij} are 4x4 matrices thus it is not possible to obtain a definition of the problem in the SL(4) group since dividing by the 4th root their determinant can yield complex matrices, since the determinant can be negative. However, it is still possible to extend the method by working with complex matrices.

The pairwise projective transformations are normalized to have a determinant equal to 1, but now $Z_{ij} \in \mathbb{C}^{4 \times 4}$. These matrices are arranged into a new matrix $Z \in \mathbb{C}^{4n \times 4n}$. Thus, it is possible to solve the problem in the same way by computing the 4 eigenvectors associated with the highest 4 eigenvalues of the matrix $(D \otimes I_3)^{-1} Z \circ (A \otimes 1_{3\times3})$. The obtained solution is a complex block matrix $X \in \mathbb{C}^{4n \times 4}$. In order to obtain a valid solution to the original problem, a real matrix is needed. Two possible alternatives have been considered:

1. Removing the imaginary values after computing the eigenvectors, but before scaling with respect to a reference node, in the same way, that the homography synchronization spectral solution does.

2. Removing the imaginary values at the very end of the computation, i.e. after scaling with respect to the reference node.

The results of these two alternatives are discussed in Section 4.3.1, where it turns out that only the second approach is feasible.

3.2. Multi Source Propagation

The Multi-Source Propagation approach (inspired by the method discussed in [4]), is a method that tries to extend the spanning tree approach. In particular, the method tries to make the spanning tree approach solution more robust by computing multiple solutions with different root nodes for the tree and then averaging the results. More in detail, the method performs the following steps:

1. Select the top 10 nodes with the highest degree: $n_1, n_2, \ldots, n_{10} \in V$
2. For $i = 1, \ldots, 10$:
   (a) Compute a minimum spanning tree $\mathcal{G}_i = (\mathcal{V}, \mathcal{T}_i)$ using n_i as the root node of the tree
   (b) Compute a solution from $\mathcal{G}_i$ using the approach discussed in Section 2.2, obtaining

$$X^{(i)} = \begin{bmatrix} X_1^{(i)} \\ X_2^{(i)} \\ \cdots \\ X_n^{(i)} \end{bmatrix} \tag{10}$$

   where $X_{n_i}^{(i)} = I_4$
3. Align each solution to the same reference node n_1, that is the node with the highest degree, by scaling each solution with respect to $X_{n_1}^{(i)}$:

$$X_j^{(i)} \leftarrow X_j^{(i)} X_{n_1}^{(i)^{-1}}, \ \forall j \in \mathcal{V}, i = 1, \ldots, 10 \tag{11}$$

4. Normalize each matrix of each solution using sum of the elements of the matrix and then the norm of the vectorized form of the matrix: $\forall j \in \mathcal{V}, i = 1, \ldots, 10$

$$X_j^{(i)'} \leftarrow \frac{X_j^{(i)}}{\sum_{k=1}^{4} \sum_{l=1}^{4} X_j^{(i)}(k,l)} \tag{12}$$

$$X_j^{(i)} \leftarrow \frac{X_j^{(i)'}}{\|x_j^{(i)'}\|_2} \tag{13}$$

   where $X_j^{(i)}(k,l)$ is the element in row k and column l of matrix $X_j^{(i)}$, and

$$x_j^{(i)'} = [X_j^{(i)'}(1,1) \ \ X_j^{(i)'}(1,2) \ \ \ldots \ \ X_j^{(i)'}(4,4)]^T \tag{14}$$

   is the vectorized form of the matrix $X_j^{(i)'}$.
5. Aggregate the results by computing an average of the 10 solutions:

$$X_j = \frac{1}{10} \sum_{i=1}^{10} X_j^{(i)}, \ \forall j \in \mathcal{V} \tag{15}$$

4. Experiments

4.1. Datasets generation

Some synthetic experiments were performed to evaluate the performance of the methods discussed in the previous sections for both homography synchronization and projective synchronization. Given a number n representing the number of nodes (transformations) in the graph, a number $\alpha \in (0, 1]$ representing the percentage of missing edges in the graph, and a parameter σ representing the standard deviation of the noise, these tests were built using some random synthetic datasets generated in the following way:

1. The matrix $X \in \mathbb{R}^{dn \times d}$ was generated from some random transformations $X_i \in \mathbb{R}^{d \times d}$, where d is 3 for homography synchronization or 4 for projective synchronization.
2. The matrix $Z \in \mathbb{R}^{dn \times dn}$ was computed using the definition of consistent labelling of the vertex: $Z = XX^{-b}$
3. A gaussian noise with standard deviation equal to σ was added to the previous matrix Z
4. The adjacency matrix A was built in this way:
   (a) Initially, the matrix is composed of all ones, except the diagonal that contains zeros
   (b) A number of transformations (i, j), with $i > j$, equal to $\alpha \frac{n(n-1)}{2}$ was selected, where $\frac{n(n-1)}{2}$ is the number of possible transformations (not counting the inverse transformations from j to i) in the complete graph
   (c) The selected transformations were removed from the adjacency matrix by setting $A_{ij} = 0$ and $A_{ji} = 0$

   The matrix produced by this step needs to represent a connected graph, thus this process is repeated until this property did not hold.

According to the considered method, the final matrix Z, which will be used as an argument of the function implementing the method, can be normalized to obtain unitary determinants.

Given a solution U found by one of the methods and by knowing the ground-truth matrix X, that is assumed to be scaled with respect to the reference node of U, the performance index of the methods was computed in the following way:

1. Each matrix X_i and U_i is vectorized as

$$x_i = [X_i(1,1) \ \ X_i(1,2) \ \ \ldots \ \ X_i(d,d)]^T \tag{16}$$

$$u_i = [U_i(1,1) \ \ U_i(1,2) \ \ \ldots \ \ U_i(d,d)]^T \tag{17}$$

2. The angle between the two vectors is computed:

$$\delta(x_i, u_i) = arccos\left(\frac{x_i^T u_i}{\|x_i\|_2 \|u_i\|_2}\right) \tag{18}$$

3. The total error is the sample mean of all the angles:

$$\epsilon(X, U) = \frac{1}{n} \sum_{i=1}^{n} \delta(x_i, u_i) \tag{19}$$

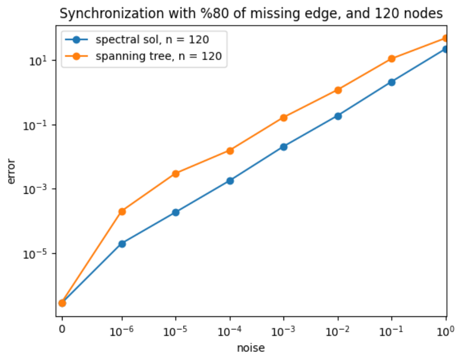

Figure 4. Homography synchronization with variable noise and 120 nodes.

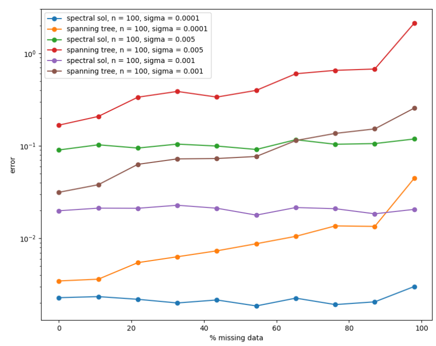

Figure 5. Homography synchronization with a variable number of missing edges.

4.2. Homography Synchronization

Two types of experiments were run for the homography synchronization problem:

- Tests with different values of the noise standard deviation
- Tests with different percentages of missing edges in the graph

Figure 4 shows how the spanning tree approach and the spectral solution perform with the increase of the noise. It turns out that the spectral solution is much more robust to the presence of noise. These results are in line with the ones obtained in [3]. For each value of the noise, 20 repeated runs were performed. Figure 5 shows how the two methods perform when the percentage of missing edges varies. In particular, it is possible to observe how the spectral solution is almost independent with respect to the percentage of missing edges, while it depends only on the value of the noise. On the other hand, the spanning tree solution has a high error when the percentage of missing edges increases.

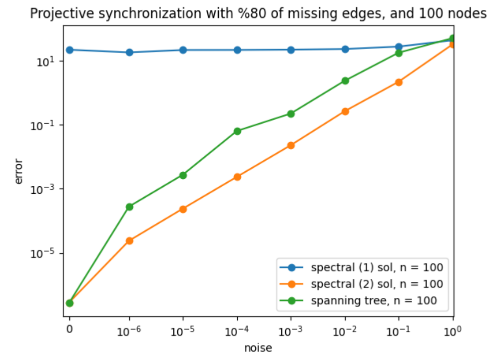

Figure 6. Projective synchronization, spectral solution, with variable noise and 100 nodes, where (1) denotes the first approach for the removal of the imaginary parts before scaling with respect to the reference node, and (2) denotes the second approach in which the removal is done after all the computations.

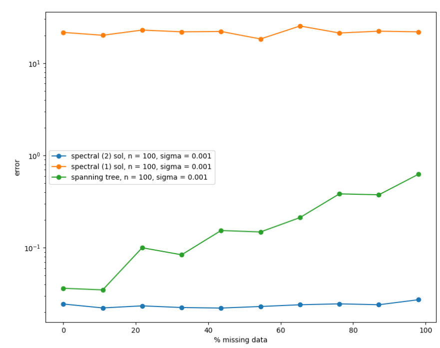

Figure 7. Projective synchronization, spectral solution, with a variable number of missing edges.

4.3. Projective Synchronization

4.3.1 Spectral solution

Similar tests with respect to the homography synchronization ones were performed to test the spectral solution also in the projective synchronization problem. Both the approaches described in Section 3.1 were tested. The result is that the first approach, in which the imaginary values are removed before scaling with respect to the reference node, is not feasible since it causes a loss of information that was given by the imaginary values. On the other hand, the second approach permits obtaining similar results to homography synchronization both when the noise is variable (Figure 6) and when the percentage of missing edges is variable (Figure 7). The reason is that the scaling with respect to the reference node yields matrices that are aligned in the same direction in the complex plane, thus removing the imaginary part is equivalent to rotating all the matrices in the complex plane by the same angle.

4.3.2 Multi Source Propagation

Figure 8 and 9 shows how the Multi-Source Propagation approach performs when the value of the noise standard deviation is varying. In particular, several tests were performed using different values for the percentage of the missing edges (the case with 20% and 80% missing edges are reported). The result is that the method is performing worse than the spectral solution, especially when the number of missing edges is high. Indeed, when the number of missing edges in the graph is high, the minimum spanning tree (and the shortest path tree) are deeper, thus computing some labels of the vertices might require many consecutive matrix multiplications that can cause numerical errors. Moreover, the method is performing worse than the spanning tree approach. The main reason why this happens is explained in Figures 10 and 11:

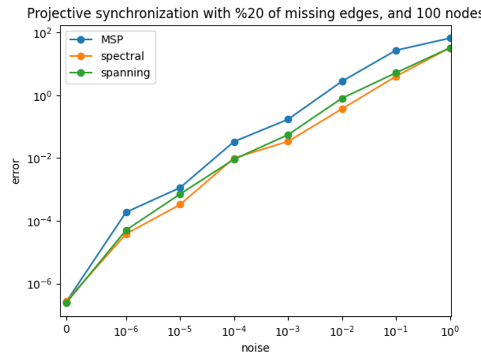

Figure 8. Projective synchronization, multi-source propagation, with variable noise and 20% missing edges.

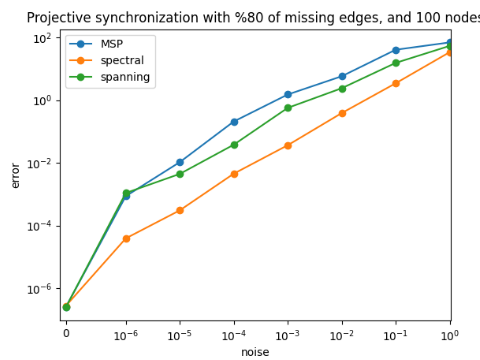

Figure 9. Projective synchronization, multi-source propagation, with variable noise and 80% missing edges.

- Figure 10 is reporting 4 different spanning tree solutions, computed using as root nodes the node with the highest degree, the node with the second highest degree, the node with the median degree and the node with the minimum degree. These solutions are all aligned with respect to the node with the highest degree, in order to simulate the behaviour of the Multi-Source Propagation method

- Figure 11 is reporting the same solutions, but each aligned with respect to their corresponding node used as a root of the tree (i.e. the solutions are not aligned).

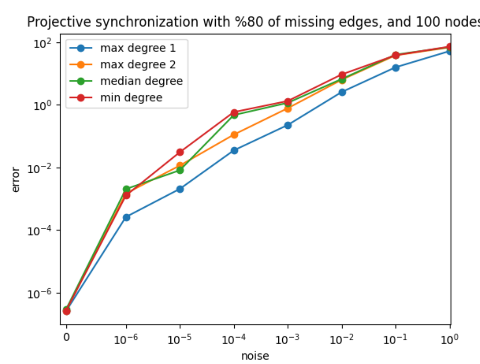

Figure 10. Projective synchronization, spanning tree approach with different root nodes and aligned solutions.

It is possible to notice that aligning the different solutions with respect to the same node is the reason why the Multi-Source Propagation is not improving the spanning tree solution. This is because this step is requiring the computation of a matrix multiplication between the solutions found using the spanning tree approach and the inverse of the matrix of the reference node, which gives additional numerical errors. On the other hand in the case with not aligned solutions this step is not performed since the solution is implicitly aligned with respect to the root node of the spanning tree by imposing its matrix equal to the identity.

4.4. Tests with outliers

Finally, some tests were performed in the presence of outliers. In this case, the generation of the synthetic datasets required some additional steps with respect to the ones previously described. In particular, given a percentage α of missing edges, a percentage β of outliers, the matrices Z and A obtained after the steps described in Section 4.1, the datasets were generated by performing the following steps:

1. Select a number of transformations (i, j) equal to $\beta(1-\alpha)\frac{n(n-1)}{2}$, where the $\frac{n(n-1)}{2}$ gives the number of transformations in a complete graph (not considering self edges) and the multiplication by $1-\alpha$ gives the number of edges in the graph.

2. For each selected edge (i, j), compute a completely random transformation R_{ij} and set $Z_{ij} = R_{ij}$ and $Z_{ji} = R_{ji}$. This random transformation can be considered an outlier since it probably does not respect

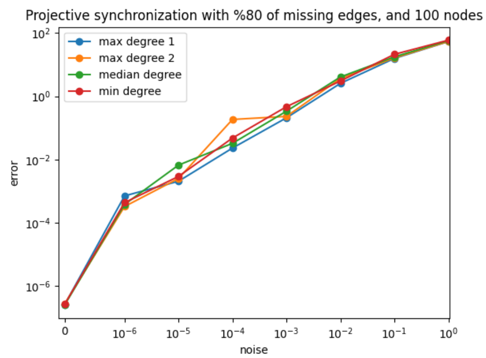

Figure 11. Projective synchronization, spanning tree approach with different root nodes and not aligned solutions.

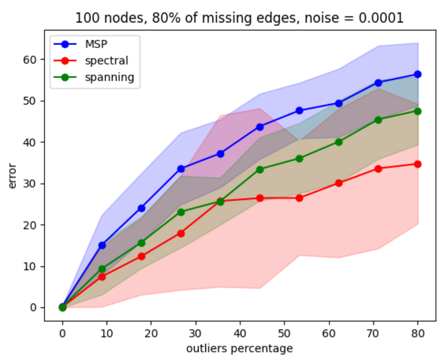

Figure 12. Projective synchronization, tests with variable percentage of outliers and 80% of missing edges. For each value of the percentage of outliers, 30 tests were performed for each method: the region highlighted between the medium of the curves is obtained by considering the standard deviation of the experiments.

the consistency constraint between nodes i and j.

As Figures 12 and 13 are showing, it is difficult to state which of the tree methods between the spanning tree, the multi-source propagation and the spectral solution is more robust to the presence of the outliers. In particular, it is possible to notice that the spectral solution is performing worse than the other methods when the number of missing edges is lower. On the other hand, the performance of the multi-source propagation and the spanning tree solution is in line with the previous results, since the multi-source propagation is always performing slightly worse than the spanning tree. Moreover, it is possible to notice that the spectral solution has a much more high variance than the other two methods, thus it is possible to say that the method is less robust to the presence of the outliers.

5. Conclusion

This project aims in investigating if it is possible to generalize the methods used to solve homography synchronization to the problem of projective synchronization. The main feature of the considered methods is that they are very easy to implement and can be run efficiently in a short time. The results have proved that is possible to obtain results similar to the homography synchronization ones. In particular, the extension of the spectral solution to the complex case permit addressing the issue of dividing by the 4th root of the determinant of the matrices, without having a degradation in the performance with respect to the 3x3 case. On the other hand, the other considered methods suffer more in terms of numerical errors due to the computation of many matrix multiplications.

Since the spectral solution gives the best performances when there are no outliers, but it is not robust when they are present, future work is needed to understand and fix this behaviour.

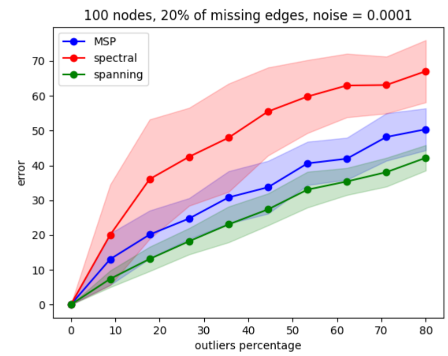

Figure 13. Projective synchronization, tests with variable percentage of outliers and 20% of missing edges.

References

[1] F. Arrigoni and A. Fusiello. Synchronization problems in computer vision with closed-form solutions. *International Journal of Computer Vision*, 128, 01 2020. 1, 2

[2] E. Santellani, E. Maset, and A. Fusiello. Seamless image mosaicking via synchronization. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2:247–254, 2018. 1

[3] P. Schroeder, A. Bartoli, P. Georgel, and N. Navab. Closed-form solutions to multiple-view homography estimation. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 650–657, 2011. 1, 2, 5

[4] L. Yang, H. Li, J. A. Rahim, Z. Cui, and P. Tan. End-to-end rotation averaging with multi-source propagation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11769–11778, 2021. 3