A Direct Approach to Viewing Graph Solvability

Federica Arrigoni[1], Andrea Fusiello[2], and Tomas Pajdla[3]

[1] DEIB - Politecnico di Milano, Italy
[2] DPIA - University of Udine, Italy
[3] CIIRC - Czech Technical University in Prague, Czechia

Abstract. The viewing graph is a useful way to represent uncalibrated cameras and their geometric relationships: nodes correspond to cameras and edges represent fundamental matrices. By analyzing this graph, it is possible to establish if the problem is "solvable" in the sense that there exists a unique (up to a single projective transformation) set of cameras that are compliant with the given fundamental matrices. In this paper, we take several steps forward in the study of viewing graph solvability: we propose a new formulation of the problem that is more direct than previous literature, based on a formula that explicitly links pairs of cameras via their fundamental matrix; we introduce the new concept of "infinitesimal solvability", demonstrating its usefulness in understanding real structure from motion graphs; we propose an algorithm for testing infinitesimal solvability and extracting components of unsolvable cases, that is more efficient than previous work; we set up an open question on the connection between infinitesimal solvability and solvability.

Keywords: Viewing graph · Fundamental Matrix · Uncalibrated camera · Structure from motion · Multi-view geometry

1 Introduction

Reconstructing a 3D scene from multiple images, as well as recovering the cameras that captured those images, is a fundamental task in Computer Vision, spanning a variety of applications such as cultural heritage and augmented/virtual reality. This problem is also referred to as *structure from motion* [23]. The difficulty of the task as well as the geometric richness of the scenario inspired both practical methods (e.g., [6, 7, 18, 25, 26, 34]) as well as theoretical results (e.g., [5, 11, 19, 20, 33]) over the last years. We refer the reader to the comprehensive survey [23] for additional references on structure from motion.

In this context, important theoretical properties can be derived by encoding the problem as a *viewing graph* [17]. Such a graph is constructed as follows: each node corresponds to one image/camera; an edge is present between two nodes if and only if it is possible to estimate the geometric relationship between the two cameras, encoded in the *fundamental matrix*. The viewing graph can be studied to establish whether it is theoretically possible to get a *unique* reconstruction of uncalibrated cameras starting from fundamental matrices. Henceforth, as customary, uniqueness is intended up to a single projective transformation.

It is well known that a *single* fundamental matrix uniquely determines the two cameras up to a projective transformation [12]. However, this may not be true when considering *multiple* fundamental matrices attached to the edges of a viewing graph. There may be cases where there are many solutions or no solution at all. The latter case leads to the problem – not dealt with in this paper – of *compatibility of fundamental matrices* [5], namely the question *whether* there is a configuration of cameras that is compatible with given fundamental matrices.

Assuming that the fundamental matrices are compatible and generic (to exclude pathological cases), the multiplicity of solutions depends on the structure of the viewing graph (i.e., the number and disposition of the edges), regardless of the actual fundamental matrices attached to the edges. In other terms, when the fundamental matrices are not sufficient to uniquely recover the cameras, there are several (typically infinitely many) transformations that can be applied to the cameras without changing the fundamental matrices (see, e.g., [17] for some examples). This leads to the notion of *solvability* of the viewing graph, which has been studied by several authors [2, 4, 17, 24, 30, 31]. A viewing graph is called *solvable*[4] when the set of (generic) cameras that yields the given fundamental matrices is unique (assuming that it exists). Otherwise it is called unsolvable. Note that solvability is a theoretical analysis that is different from the practical task of retrieving cameras from fundamental matrices [14, 26, 28]. However, the two problems are related: testing solvability serves as a pre-processing step for reconstruction, in order to ensure that the latter is well-posed.

The analogous of solvability in the calibrated case (where essential matrices replace fundamental ones) is well understood: it is equivalent to parallel rigidity [22], and there exists a fair amount of theoretical results about parallel rigidity [1] as well as effective algorithms for extracting the maximal rigid component of an unsolvable graph [13,15,32]. In this paper we are interested in considering viewing graph solvability in the *uncalibrated case*, which still presents open questions and room for improvement, as will be discussed in the next section.

1.1 Related Work

The concept of *solving* viewing graph (later called *solvable* by [31]) was first introduced in [17] where small incomplete graphs (up to six cameras) were manually analyzed, by reasoning in terms of how to uniquely recover the missing fundamental matrices starting from the available ones. The authors of [17] also derived a *necessary condition* for solvability, namely the property that all the vertices have degree at least two and no two adjacent vertices have degree two. Later, additional necessary conditions were developed: a solvable graph is biconnected [31]; it has at least $(11n - 15)/7$ edges, with n being the number of nodes [31]; it is parallel rigid [3]. The latter means that, as expected, a graph that is solvable with unknown intrinsic parameters is also solvable when they

[4] However misleading it may be, we are compelled to use the term "solvable" to refer to a property that should more appropriately be named something as rigid/tight/iso-constrained [10, 29], as it has been previously defined so in other related fields.

are known. *Sufficient conditions* are also available: in [30] it is proved that those graphs which are constructed from a 3-cycle by adding vertices of degree 2 one at a time are solvable; [31] introduces specific “moves” which can be applied to a graph, possibly transforming it into a complete one, in which case the graph is solvable; a sufficient condition based on a linear system is proposed in [24].

In fact, necessary or sufficient conditions alone are not enough to classify all possible graphs so a *characterization* of the problem is required. In this respect, the authors of [2, 31] study solvability using principles from algebraic geometry, therefore falling under the emerging field of Algebraic Vision [16]. Specifically, a polynomial system of equations was derived in [31], such that solvability can be tested by counting the number of solutions of the system with algebraic geometry tools (e.g., Gröbner basis computation). Building on [31], the authors of [2] improve efficiency by deriving a simplified polynomial system with fewer unknowns. Still, the largest example that was tested in [2] is a graph with 90 nodes, which is far from the size of typical structure from motion datasets.

The main drawback of [2, 31] is that solving polynomial equations is computationally highly demanding, therefore limiting the practical usage of this characterization of solvability. For this reason, the related concept of *finite solvability* has been explored [4, 31]. Specifically, a graph is called finite solvable if there is a *finite* number of camera configurations that are compliant with the given fundamental matrices. This concept represents a proxy for (unique) solvability for it does not exclude the presence of more than one solution (e.g., two distinct solutions), however, it has been shown to be more practical since it admits a *linear* formulation [31]: indeed, finite solvability can be deduced from the rank of a suitable matrix. Later, the authors of [4] improved the efficiency of such a formulation, and developed an algorithm for partitioning an unsolvable graph into maximal components that are finite solvable. Nevertheless, the approach from [4, 31], has some disadvantages:

- the number of unknowns depends on the number of edges in the graph (instead of nodes), leading to reduced efficiency (see Tab. 2 and Fig. 3);
- the formulation is hard to interpret, for it does not explicitly involve cameras and fundamental matrices but unknown projective transformations that represent all the possible ambiguities of the problem (see Remark 1);
- the methodology heavily depends on proving results with algebraic geometry tools that are specific for the chosen equations, but can not be easily adapted to other (possibly new) formulations (see Remark 3).

In the following sections we will address these drawbacks and propose a solution.

1.2 Contribution

In this paper we advance the understanding of the hard task of viewing graph solvability via the following contributions:

- We derive a new formulation of the problem (i.e., a new set of polynomial equations) that directly employs the definition of solvability, i.e., a graph

is called solvable if and only if the available fundamental matrices uniquely determine the cameras (up to a single projective transformation). Our formulation is more *intuitive* than previous work, as it directly involves cameras and fundamental matrices (see Sec. 3).
- We show that, by evaluating the rank of the Jacobian matrix of the polynomial equations in a fabricated solution (see Sec. 4.1), we can test the presence of a *finite* number of solutions in the neighborhood of that solution, leading to the new concept of "infinitesimal solvability".
- For unsolvable cases, we develop a method for partitioning the graph into the maximal components that are infinitesimally solvable (see Sec. 4.2). In our formulation the number of unknowns depends on the number of nodes in the graph, that are typically much inferior to the number of edges used by previous work, therefore setting the state of the art in terms of *efficiency*.
- We demonstrate the usefulness of infinitesimal solvability through experiments on synthetic and real graphs in Sec. 5.
- While conjecturing that they are equivalent, we left open the research question on the connection between infinitesimal solvability and finite solvability, in the hope to stimulate further research on this topic (see Sec. 5.3 and 6).

In order to derive our contributions, some preliminary notions are required, which are introduced in the next section.

2 Theoretical Framework

We review here the concept of solvability, previously studied by many authors. Given a set of fundamental matrices stored as edges in a viewing graph, following [31] we say that the graph is *solvable* if and only if there exists a unique set of uncalibrated cameras (associated to the nodes) that produce the input fundamental matrices. The problem can be framed in terms of a polynomial system of equations and solvability can be tested by counting the number of solutions:

$$\mathbf{g}(\mathbf{p}, \mathbf{x}) = \mathbf{0} \text{ has a unique solution} \iff \text{the graph is solvable} \tag{1}$$

where $\mathbf{g} : \mathbb{R}^p \times \mathbb{R}^v \mapsto \mathbb{R}^e$ is a polynomial map with p parameters, v unknowns and e equations, therefore $\mathbf{g}(\mathbf{p}, \mathbf{x}) = \mathbf{0}$ represents a polynomial system in the unknown $\mathbf{x}$. The form of the system, the meaning of the parameters/unknowns as well as the number of parameters/equations/unknowns are not universal, but they depend on the specific formulation that is being considered. Differently from previous work [2, 31], in this paper the parameters $\mathbf{p} \in \mathbb{R}^p$ are the input fundamental matrices (properly stacked in a vector) and the unknowns $\mathbf{x} \in \mathbb{R}^v$ are the cameras, as it will be clarified later.

Since solvability is a generic property of the graph (i.e., it does not depend on the actual fundamental matrices), it is typically tested on a *random set of parameters*: as a result, the number of solutions will always be the same with probability one. Counting the number of solutions uses Gröbner basis, whose worst-case complexity is exponential in the number of variables [8], so current research focuses primarily on the extraction of new formulations or a reduced set of unknowns to improve efficiency.

3 Proposed Formulation

The starting point of our derivations is a new set of polynomial equations that characterize solvability. Specifically, we derive here a polynomial system where the only unknowns are the camera matrices by using the *definition of solvability*, that links fundamental matrices to cameras. In other terms, we start with a graph with compatible fundamental matrices on the edges[5], so a solution always exists, and the task is to check whether it is unique or not (recall that a solution is a camera configuration that yields the fundamental matrices).

Formally, let $P_1, \dots, P_n$ denote 3×4 matrices representing uncalibrated cameras and let F_{ij} denote the 3×3 fundamental matrix between cameras P_i and P_j. Using this notation, the viewing graph is an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$, such that node $i \in \mathcal{V}$ is associated with (unknown) camera P_i and edge $(i, j) \in \mathcal{E}$ is associated with (known) fundamental matrix F_{ij}. The number of nodes is denoted by n, whereas the number of edges is denoted by m. There exist many formulae that connect a fundamental matrix to the corresponding camera pair. We concentrate on this result from [12]:

Lemma 1. *A non-zero matrix F_{ij} is the fundamental matrix corresponding to a pair of cameras P_i and P_j if and only if $S = P_j^\top F_{ij} P_i$ is skew-symmetric.*

Please note than any scaling of each of the three terms of the product would clearly leave the result skew-symmetric. The above formula can be rewritten as:

$$S + S^\top = 0 \quad \Longleftrightarrow \quad P_j^\top F_{ij} P_i + P_i^\top F_{ij}^\top P_j = 0 \tag{2}$$

which translates in 10 quadratic equations when considered entry-wise. In the supplementary material we show that the Jacobian of (2) with respect to vectorized cameras $\operatorname{vec}(P_j)$ and $\operatorname{vec}(P_i)$ is (this will be used in Sec. 4.1):

$$J = (K_{4,4} + I_{16}) \left[(I_4 \otimes (F_{ij} P_i)^\top) \,|\, (I_4 \otimes (P_j^\top F_{ij})) \right] \tag{3}$$

where $\otimes$ is the Kronecker product, I denotes the identity matrix, and $K_{4,4}$ is the permutation matrix such that $\operatorname{vec}(A) = K_{n,q} \operatorname{vec}(A^\top)$ for any $A_{n \times q}$.

Note that Eq. (2) holds for a single edge $(i, j) \in \mathcal{E}$. By collecting equations coming from all the edges in the graph, it results in a *polynomial system* with $10m$ equations and $12n$ unknowns. Recall that m is the number of edges/fundamental matrices and n is the number of nodes/cameras. Extra care must be taken to ensure that Eq. (2) has a unique solution for a solvable graph. Specifically, *additional equations* must be included in order to manage the global projective ambiguity (which is inherent to the problem) and the fact that each matrix is defined up to scale. Furthermore, we have to avoid degeneracies arising from cameras that are not full-rank.

[5] From the implementation point of view, fundamental matrices are obtained from a set of random cameras associated to the vertices.

The global *projective ambiguity* is fixed, without loss of generality, by arbitrarily choosing the first camera[6] and the first row in the second camera:

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} P_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

resulting in 16 additional equations. Note that $[1\ 0\ 0]P_2$ is equivalent to selecting the first row in P_2. Concerning the *scale ambiguity*, the scale of each camera can be arbitrarily set, e.g., by fixing the sum of its entries to 1:

$$\mathbf{1}_{12}^{\mathsf{T}} \mathrm{vec}(P_i) = 1 \tag{5}$$

where $\mathbf{1}_{12}$ denotes a vector of ones of length 12. This results in a linear equation for each node (except the first camera used to fix the global ambiguity). Finally, we should also enforce that each camera has *full-rank*. To this end, we exploit the following lemma.

Lemma 2. *Let W_i be a 4×4 matrix obtained by vertically concatenating P_i with four random values α, β, γ, δ:*

$$W_i = \begin{bmatrix} & P_i & & \\ \alpha & \beta & \gamma & \delta \end{bmatrix}. \tag{6}$$

Let $P_i^{(1)}$, $P_i^{(2)}$, $P_i^{(3)}$, $P_i^{(4)}$ be the 3×3 submatrices of the 3×4 camera matrix P_i obtained by removing the first, second, third and fourth columns, respectively. Then almost surely:

$$\det(W_i) \neq 0 \iff \det(P_i^{(1)}) \neq 0 \vee \det(P_i^{(2)}) \neq 0 \vee \det(P_i^{(3)}) \neq 0 \vee \det(P_i^{(4)}) \neq 0.$$

Proof. Note that $\det(W_i) = -\alpha \det(P_i^{(1)}) + \beta \det(P_i^{(2)}) - \gamma \det(P_i^{(3)}) + \delta \det(P_i^{(4)})$. In one direction ($\Longrightarrow$): by contradiction let us assume that the thesis is false, i.e., $\det(P_i^{(1)}) = \det(P_i^{(2)}) = \det(P_i^{(3)}) = \det(P_i^{(4)}) = 0$, hence it follows immediately that $\det(W_i) = 0$, which is a contradiction. In the other direction ($\Longleftarrow$): by assumption, at least one submatrix has nonzero determinant, hence let us suppose (wlog) that $\det(P_i^{(1)}) \neq 0$; we have to prove that $\det(W_i) \neq 0$, which is equivalent to $\alpha \neq (\beta \det(P_i^{(2)}) - \gamma \det(P_i^{(3)}) + \delta \det(P_i^{(4)})) / \det(P_i^{(1)})$; since α has been sampled at random, there is zero probability that $\alpha = (\beta \det(P_i^{(2)}) - \gamma \det(P_i^{(3)}) + \delta \det(P_i^{(4)})) / \det(P_i^{(1)})$, hence we get the thesis. $\square$

Observe that the right term in Eq. (2) is tantamount to saying that at least one 3×3 submatrix of P_i has non-zero determinant, which in turn is equivalent to impose that the camera is indeed rank-3 and not less, as required. In order to convert this constraint into a polynomial equation we consider the following:

$$z_i \det(W_i) + 1 = 0 \tag{7}$$

[6] In fact, any pair of cameras can be chosen to fix the projective ambiguity. In practice, we will use two nodes that are endpoint of an edge in the graph (see Sec. 4.2).

where $z_i \in \mathbb{R}$ is an auxiliary variable[7]. The idea is that, if the determinant is zero, then such equation cannot be satisfied.

To summarize, equations of the form (2), (4), (5) and (7) are all collected in a polynomial system, for a total of $10m + 2(n - 1) + 16 = 10m + 2n + 14$ equations. Specifically, equations of the form (2) are derived from Lemma 1 combined with the definition of solvability that explicitly links cameras and fundamental matrices: for this reason we call our approach *direct*. Although Lemma 1 is a well-known result in multiview geometry, it has never been used for solvability so far. Equations of the form (4), (5) and (7) have been introduced to manage the projective/scale ambiguity as well as the requirement of full-rank cameras: this has the effect that the solution will be *exactly one* for a solvable graph. The unknowns of our polynomial system are the camera matrices plus the auxiliary variables defined above, for a total of $13n - 1$ unknowns: they *depend on the number of nodes*, as desired.

Remark 1. Using the terminology from parallel rigidity theory [1, 32], our formulation can be regarded as *node-based*, whereas [2, 31] can be regarded as *edge-based*. The latter indeed reason in terms of some unknown transformations (one for each edge in the graph) that can be ideally applied to the cameras without changing the fundamental matrices. Our approach, instead, works directly with unknown cameras (one for each node in the graph).

4 Proposed Method

In this section we introduce a new concept named *infinitesimal solvability*, motivated by the fact that checking solvability is highly demanding and far from being practical. We also show how to partition an unsolvable graph into maximal components that are infinitesimally solvable. We will clarify the connection with the related concept of finite solvability in Remark 3.

4.1 Testing infinitesimal solvability

As explained in Sec. 3, our direct formulation of viewing graph solvability collects polynomial equations of the form (2), (4), (5) and (7), taking into account both the relation between a fundamental matrix and the corresponding camera pair, as well as additional constraints to enforce cameras of full-rank and manage the projective and scale ambiguities. Note that the input graph comes into play in Eq. (2), as only available edges contribute to equations of this form. Using the terminology from Sec. 2, we have a polynomial system of the form $\mathbf{g}(\mathbf{p}, \mathbf{x}) = \mathbf{0}$ where $\mathbf{g} : \mathbb{R}^p \times \mathbb{R}^v \mapsto \mathbb{R}^e$, $p = 9m$, $v = 13n-1$ and $e = 10m+2n+14$. Specifically, the vector of parameters $\mathbf{p}$ concatenates the entries of all fundamental matrices,

[7] The total number of auxiliary variables is $n - 1$ since we do not need to enforce the full-rank constraint for the first camera which is completely determined.

while the vector of unknowns $\mathbf{x}$ concatenates the entries of all cameras and auxiliary variables:

$$\mathbf{p} = \begin{bmatrix} \text{vec}(F_{12}) \\ \dots \\ \text{vec}(F_{ij}) \\ \dots \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \text{vec}(P_1) \\ \dots \\ \text{vec}(P_n) \\ z_2 \\ \dots \\ z_n \end{bmatrix} \tag{8}$$

where vec( ) denotes the vectorization of a matrix.

In order to study the number of solutions of $\mathbf{g}(\mathbf{p}, \mathbf{x}) = \mathbf{0}$, we exploit the so-called "Jacobian check" [16, 27], that is reviewed here. It is a general technique from Algebraic Geometry that can be applied to any polynomial system (not necessarily equations describing solvability), for example, in the context of minimal problems in Computer Vision [9]. Given a polynomial system of the form $\mathbf{g}(\mathbf{p}, \mathbf{x}) = \mathbf{0}$ in the unknown $\mathbf{x}$, the derivatives of equations with respect to unknowns are computed:

$$J_{ij}(\mathbf{p}, \mathbf{x}) = \frac{\partial g_i(\mathbf{p}, \mathbf{x})}{\partial x_j} \tag{9}$$

and they are collected in an $e \times v$ matrix $J(\mathbf{p}, \mathbf{x})$ called the *Jacobian* of the system. The Jacobian is then evaluated at a given solution $\widetilde{\mathbf{x}}$ associated to a specific set of parameters $\widetilde{\mathbf{p}}$. Such a matrix reveals important properties:

$$\begin{aligned} \mathbf{g}(\widetilde{\mathbf{p}}, \mathbf{x}) = \mathbf{0} \ \text{ has a finite number of solutions around } \widetilde{\mathbf{x}} \\ \iff J(\widetilde{\mathbf{p}}, \widetilde{\mathbf{x}}) \ \text{ has full rank.} \end{aligned} \tag{10}$$

In other terms, information about the local structure of the problem can be deduced from the rank of the Jacobian matrix.

Applying this procedure to our formulation, we build the Jacobian by using m blocks like (3) plus the derivatives of the $2n + 14$ additional equations, and compute its rank at a given solution (which always exists by construction): if the rank is full we say that the graph is *infinitesimally solvable* (Fig. 1). Our new definition has two main advantages:

- it is easy to implement and computationally feasible to test, being based on computing the rank of a matrix;
- it is general, i.e., it holds for any formulation of solvability.

To complete the description of the algorithm for testing infinitesimal solvability, as mentioned before, we need a solution $\widetilde{\mathbf{x}}$ associated to a specific set of parameters $\widetilde{\mathbf{p}}$. Concerning $\widetilde{\mathbf{x}}$, we sample at random n cameras $\widetilde{P}_1, \dots, \widetilde{P}_n$ ensuring that they are full-rank, and we set $\widetilde{z}_i = -1/\det(\widetilde{W}_i)$ in order to satisfy Eq. (7), with $\widetilde{W}_i$ constructed by concatenating the camera $\widetilde{P}_i$ and a random vector as in Eq. (6). Then, the parameter vector $\widetilde{\mathbf{p}}$ is simply obtained by computing, for each edge in the graph, the fundamental matrix associated to the corresponding cameras, for which there are closed-form formulas [12].

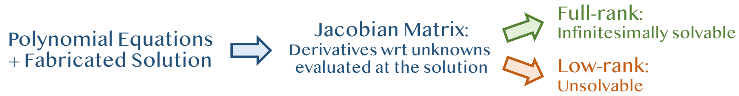


Fig. 1: Our approach for analyzing the solvability of viewing graphs.

Our polynomial equations and this *fabricated* solution are then used to construct the Jacobian matrix J, which contains the derivatives of equations with respect to unknowns, and is evaluated at the given solution (see Fig. 1). The exact form of such derivatives is given in the supplementary material. Hence:

– if J has full rank then the graph is infinitesimally solvable;
– if J has low-rank then the graph is unsolvable (see Remark 2).

The rank can be checked by computing the smallest singular value of J, which in turn is equivalent to computing the smallest eigenvalue of $J^{\mathsf{T}}J$. Note that the Jacobian matrix is sparse, since only two cameras are involved in each equation of the form (2). In Sec. 5 we will provide experimental results on the performance of our method, as well as a comparison between the size of our Jacobian matrix and the one used by previous work [4].

Remark 2. In an infinitesimally solvable graph, there are no guarantees that the number of solutions (i.e., the number of camera configurations compatible with the given fundamental matrices) will be finite (or even unique) overall. However, if a graph is not infinitesimally solvable then it has an infinite number of solutions in a neighborhood of a particular one, so it has an infinite number of solutions *overall*. In other terms:

$$\text{not infinitesimally solvable} \implies \text{unsolvable} \tag{11}$$

Remark 3. In [4, 31] the notion of *finite solvability* is introduced. In that case, the solution set of the considered polynomial equations has special properties, namely it is an "algebraic group", hence its dimension can be obtained as the dimension of its tangent space, which is a linear space, and thus reduces to determining the rank of a linear system of equations. Specifically, finite solvability corresponds to the zero-dimensional case (see the supplementary material of [31] for more details). This local analysis on the tangent space allows to predicate on the overall number of solutions, *thanks to the algebraic group structure* of the solutions set of the equations of [31]. It is unknown whether the same holds in our formulation, that does not enjoy the same algebraic structure (our unknowns are the camera matrices, that are not invertible, hence they do not form a group). It is straightforward to see that the following implication holds:

$$\text{finite solvable} \implies \text{infinitesimally solvable} \tag{12}$$

See also Fig. 4 for a visualization on the relation between different concepts.

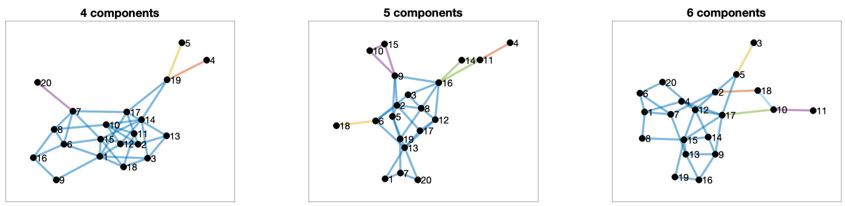


Fig. 2: Examples of maximal components on synthetic viewing graphs, where each component is represented with a different color.

4.2 Finding Maximal Components

If a viewing graph is established to be unsolvable, then the challenge is to find maximal subgraphs (called *components*) that are infinitesimally solvable. It is easy to see that Prop. 3 from [4] also holds for infinitesimal solvability, namely such components form a *partition* of the edges. In other terms, each edge belongs to exactly one component. However, it is important to remark that a node can belong to more components. For example, with reference to Fig. 2 (Right), node 5 belongs to two different components: the yellow one and the blue one.

For this reason, we can not trivially use the *edge-based* methodology from [4], for our formulation is *node-based*. Indeed, the null space of the Jacobian matrix J, which is nontrivial for an unsolvable graph, is such that any block of 12 rows corresponds to a camera/node in the viewing graph (whereas in [4] there was a correspondence between rows in the null space and edges in the graph). Luckily, it is still possible to identify components from the null space of J with proper modifications. In this context, an important observation is that we need two nodes to fix the global projective ambiguity: in particular, for the purpose of identifying components, it is useful to select two adjacent nodes (i.e., an edge).

Lemma 3. *Let J be the Jacobian matrix defining infinitesimal solvability of a viewing graph, constructed as explained above, and let N be the null space of J. A node is in the same component as the edge used to fix the global projective ambiguity if and only if the corresponding rows in N are zero.*

Proof. Following the reasoning from [4], we can prove the thesis based on this observation: if we focus on the component containing the edge used to fix the global projective ambiguity, then the fact that all ambiguities have been fixed in that component, it is equivalent to say that there are no degrees of freedom, i.e., the null space is trivial on that component. □

According to the above result, we can define an iterative approach to identify components: first, we use two adjacent nodes to fix the global projective ambiguity and identify all nodes within such component by selecting the ones corresponding to the zero rows in N; then, we repeatedly apply the same procedure to the remaining part of the graph until there are no more edges to be assigned. Observe that, although the null space is computed several times (equal

Table 1: Results of our synthetic experiments. On all graphs we tested for finite solvability [4] and infinitesimal solvability (our method) and counted the number of graphs that passed the test. Since in all cases both methods gave the same results, we only report one column named "ns".

(a) Solvability for all the biconnected graphs with up to 10 nodes and $\lceil(11n-15)/7\rceil$ edges, called "candidates".

#nodes	#candidates	ns
3	1	1
4	1	1
5	2	1
6	9	4
7	20	3
8	161	36
9	433	27
10	5898	756

(b) Solvability for 1000 random graphs with 20 nodes and varying density. The last column reports the [min max] number of components.

%density	ns	#comp.
5	0	[10, 25]
10	2	[1, 27]
20	253	[1, 24]
30	826	[1, 5]
40	977	[1, 3]
50	999	[1, 2]
60	1000	[1, 1]
70	1000	[1, 1]

to the number of components), the Jacobian matrix has a size that gets smaller and smaller, for the procedure is not re-applied to the whole graph but only to the subgraph containing the remaining edges. Some examples are given in Fig. 2.

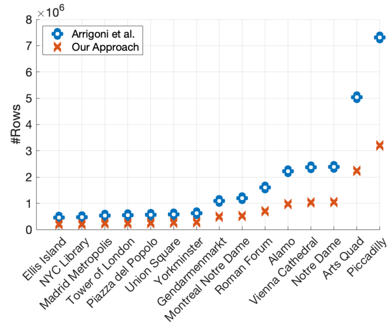
5 Experiments

In this section we report results on synthetic/real data and we also discuss limitations of our approach. We implemented our method in MATLAB R2023b – the code is publicly available[8] – and used a MacMini M1 (2020) with 16Gb RAM for our experiments. The closest method to ours is the one by Arrigoni et al. [4], that addresses finite solvability (the code is available online). See Remark 3 and Fig. 4 for information on the relation between infinitesimal solvability – addressed by our approach – and finite solvability. We did not consider the method by Trager et al. [31] in our comparisons since it was subsumed by [4].

5.1 Synthetic Data

In this experiment, we exhaustively generated all the graphs with a given number of nodes (up to ten) having $\lceil(11n-15)/7\rceil$ edges, that satisfy a necessary condition for being solvable, namely being biconnected [31]. Among these candidates, we tested for finite solvability (using [4]) and infinitesimal solvability (using our method). The results are reported in Tab. 1a. In all cases, the two approaches gave the same output: note that this property is not granted a priori, as infinitesimal solvability and finite solvability are not proven to be equivalent.

[8] `https://github.com/federica-arrigoni/finite-solvability`



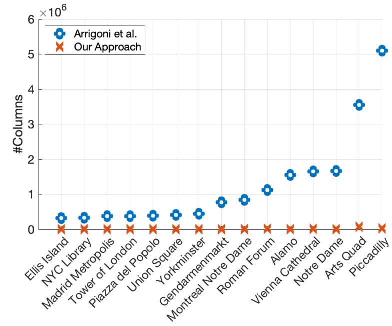


Fig. 3: Number of rows and columns of the matrix used by Arrigoni et al. [4] and the one from our formulation on large-scale datasets [7, 34]. Note that the difference in the number of columns even surpasses one order of magnitude.

We then analyzed synthetic graphs with 20 nodes[9] generated by randomly selecting a fixed percentage of edges from the complete graph (named density); disconnected graphs were discarded. We considered increasing density values ranging from 5% to 70%: for each value, 1000 graphs were sampled, for a total of 8000 samples. Both our method and [4] were applied to each graph and they always gave the same output. Results are collected in Tab. 1b: as expected, when the percentage of edges decreases, the graphs are more likely to be unsolvable and the number of components increases. Some visualizations of the components are depicted in Fig. 2, showing established cases of unsolvable graphs, such as the "square" topology or the presence of pending edges and articulation points.

5.2 Real Data

As done in [4], we consider viewing graphs taken from typical structure from motion datasets: the Cornell Arts Quad dataset [7], the 1DSfM dataset [34], and image sequences from [21]. Some statistics about these graphs are reported in Tab. 2, namely: the number of nodes; the number of edges; the density (i.e., the percentage of available edges with respect to the complete graph). Table 2 also reports the outcome of our real experiments, namely the number of components and the execution times of the competing methods. Observe that #components=1 is equivalent to say that the graph is infinitesimally solvable (for our approach) or finite solvable (for [4]). See also Fig. 3 for information on the number of rows and columns of the matrix used by Arrigoni et al. [4] and the one from our formulation (formulas are given in the supplementary material).

Results show that there are only five unsolvable cases among the analyzed datasets, all exhibiting four components, in agreement with previous work. It is worth observing that our direct method and the one by Arrigoni et al. [4] always give the same result on all the graphs, in agreement with the outcome of our

[9] We also tested other values for the number of nodes, obtaining comparable results.

Table 2: Results of our experiments on real datasets [7,21,34]. "Time" is the total time (in seconds) spent building the solvability matrix (Arrigoni et al. [4]) or the Jacobian matrix (our approach), testing for finite solvability and infinitesimal solvability respectively, and computing the #*comp.* components (only on non-solvable cases).

Dataset				Arrigoni et al. [4]		Our Method	
Name	#nodes	%density	#edges	#comp.	Time	#comp.	Time
Gustav Vasa	18	72	110	1	0.10	1	0.37
Dino 319	36	37	230	1	0.07	1	0.15
Dino 4983	36	37	231	1	0.06	1	0.06
Folke Filbyter	40	32	250	1	0.06	1	0.05
Jonas Ahls	40	41	321	1	0.09	1	0.20
Park Gate	34	94	529	1	0.13	1	0.12
Toronto University	77	33	974	1	0.30	1	0.18
Sphinx	70	55	1330	1	0.53	1	0.23
Cherub	65	64	1332	1	0.53	1	0.24
Tsar Nikolai I	98	52	2486	1	1.32	1	0.46
Skansen Kronan	131	88	7490	1	10.08	1	2.07
Alcatraz Courtyard	133	92	8058	1	11.59	1	2.30
Buddah Tooth	162	73	9546	1	15.47	1	2.93
Pumpkin	195	65	12276	1	26.16	1	4.63
Ellis Island	240	71	20290	1	74.22	1	11.54
NYC Library	358	32	20662	1	74.44	1	12.10
Madrid Metropolis	370	35	23755	1	99.45	1	15.25
Tower of London	489	20	23844	4	103.83	4	19.53
Piazza del Popolo	345	42	24701	4	108.57	4	18.35
Union Square	853	7	25478	4	124.94	4	27.64
Yorkminster	448	28	27719	1	126.89	1	19.31
Gendarmenmarkt	722	18	48124	4	411.67	4	69.81
Montreal N. Dame	467	48	52417	1	462.81	1	65.53
Roman Forum	1102	12	70153	4	913.57	4	158.16
Alamo	606	53	97184	1	2335.94	1	222.35
Vienna Cathedral	898	26	103530	1	2565.31	1	253.84
Notre Dame	553	68	103932	1	2631.00	1	249.12
Arts Quad	5460	1	221929	1	10979.35	1	898.56
Piccadilly	2446	11	319195	1	25889.20	1	2361.20

synthetic experiments. Table 2 also shows that our method is significantly faster than the state of the art, underlying the advantage of a node-based approach with respect to an edge-based one. Indeed, the matrix employed by our formulation is significantly smaller than the one used by the authors of [4], as can be seen in Fig. 3. In particular, our direct formulation takes less than 10% of the total running time of [4] on the largest datasets (from "Alamo" to "Piccadilly"). This figure becomes 20% for medium size datasets (from "Skansen Kronan" to "Roman Forum"). For the smallest ones the running time is less than a second and the comparison becomes meaningless.

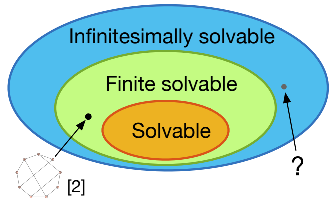


Fig. 4: Relation between different notions. While there exist examples of graphs that are finite solvable but not solvable [2], it is unknown if there are graphs that are infinitesimally solvable but not finite solvable. We conjecture that the answer is "no", i.e., blue and green sets coincide, as supported by empirical evidence.

5.3 Limitations

The concept of infinitesimal solvability, while valuable, is only one step towards characterizing the solvability of a viewing graph in a computationally tractable way. However, it has two inherent limitations: i) it asserts the existence of a finite number of solutions rather than guaranteeing a single one and ii) it only addresses a neighbourhood of a particular solution. It remains an open question whether this property extends to the whole solution set or not (see Fig. 4).

6 Conclusion

This paper highlighted the usefulness of the viewing graph as an effective representation of uncalibrated cameras and their geometric relationships. In particular, the solvability of the graph is equivalent to the existence of a unique set of conforming cameras, up to a single projective transformation, based on the given fundamental matrices. We presented a novel formulation of the problem that provides a more direct approach than previous literature. We based our approach on a formula that explicitly establishes links between pairs of cameras through their fundamental matrices, as suggested by the definition of solvability.

In addition, we introduced the new concept of "infinitesimal solvability" and showed its practical importance by analyzing structure-from-motion graphs. We presented an algorithm designed to test infinitesimal solvability and extract components of unsolvable cases, surpassing the efficiency of previous methods. Furthermore, our framework has the advantage of being general: the core methodology only requires computing the derivatives of polynomial equations with respect to their unknowns, and can therefore be easily applied to new formulations of viewing graph solvability to be developed in the future.

Finally, we raise a research question about the connection between infinitesimal solvability and finite solvability. Although empirical evidence supports the conjecture that the two notions are equivalent, this remains an open problem. We hope that our results will inspire further research on this fascinating topic.

Acknowledgements. This paper is supported by PNRR-PE-AI FAIR project funded by the NextGeneration EU program. Tomas Pajdla was supported by the OPJAK CZ.02.01.01/00/22 008/0004590 Roboprox Project.

References

1. Arrigoni, F., Fusiello, A.: Bearing-based network localizability: A unifying view. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(9), 2049 – 2069 (2019)
2. Arrigoni, F., Fusiello, A., Ricci, E., Pajdla, T.: Viewing graph solvability via cycle consistency. In: Proceedings of the International Conference on Computer Vision. pp. 5540 – 5549 (2021)
3. Arrigoni, F., Fusiello, A., Rizzi, R., Ricci, E., Pajdla, T.: Revisiting viewing graph solvability: an effective approach based on cycle consistency. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–14 (2022)
4. Arrigoni, F., Pajdla, T., Fusiello, A.: Viewing graph solvability in practice. In: Proceedings of the International Conference on Computer Vision. pp. 8147–8155 (2023)
5. Bratelund, M., Rydell, F.: Compatibility of fundamental matrices for complete viewing graphs. In: Proceedings of the International Conference on Computer Vision. pp. 3305 – 3313 (2023)
6. Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: Proceedings of the International Conference on Computer Vision (2013)
7. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.P.: Discrete-continuous optimization for large-scale structure from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3001–3008 (2011)
8. Dubé, T.W.: The structure of polynomial ideals and Gröbner bases. SIAM Journal on Computing **19**(4), 750 – 773 (1990)
9. Duff, T., Kohn, K., Leykin, A., Pajdla, T.: Plmp – point-line minimal problems in complete multi-view visibility. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(1), 421–435 (2024)
10. Graver, J., Servatius, B., Servatius, H.: Combinatorial Rigidity. Graduate studies in mathematics, American Mathematical Soc. (1993)
11. Hartley, R.I., Trumpf, J., Dai, Y., Li, H.: Rotation averaging. International Journal of Computer Vision (2013)
12. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, second edn. (2004)
13. Karimian, A., Tron, R.: Theory and methods for bearing rigidity recovery. In: Proceedings of the IEEE Conference on Decision and Control. p. 2228–2235 (2017)
14. Kasten, Y., Geifman, A., Galun, M., Basri, R.: GPSfM: Global projective SFM using algebraic constraints on multi-view fundamental matrices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3259–3267 (2019)
15. Kennedy, R., Daniilidis, K., Naroditsky, O., Taylor, C.J.: Identifying maximal rigid components in bearing-based localization. In: Proceedings of the International Conference on Intelligent Robots and Systems. pp. 194 – 201 (2012)
16. Kileel, J., Kohn, K.: Snapshot of algebraic vision. arXiv (2210.11443) (2023)
17. Levi, N., Werman, M.: The viewing graph. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 518 – 522 (2003)

18. Magerand, L., Del Bue, A.: Revisiting projective structure from motion: A robust and efficient incremental solution. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(2), 430–443 (2020)
19. Manam, L., Govindu, V.M.: Sensitivity in translation averaging. In: Neural Information Processing Systems (NeurIPS) (2023)
20. Nasihatkon, B., Hartley, R., Trumpf, J.: A generalized projective reconstruction theorem and depth constraints for projective factorization. International Journal of Computer Vision **115**, 87 – 115 (2015)
21. Olsson, C., Enqvist, O.: Stable structure from motion for unordered image collections. In: Proceedings of the 17th Scandinavian conference on Image analysis (SCIA'11). pp. 524–535. Springer-Verlag (2011)
22. Ozyesil, O., Singer, A.: Robust camera location estimation by convex programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2674 – 2683 (2015)
23. Ozyesil, O., Voroninski, V., Basri, R., Singer, A.: A survey of structure from motion. Acta Numerica **26**, 305 – 364 (2017)
24. Rudi, A., Pizzoli, M., Pirri, F.: Linear solvability in the viewing graph. In: Proceedings of the Asian Conference on Computer Vision. pp. 369–381 (2011)
25. Sarlin, P.E., Lindenberger, P., Larsson, V., Pollefeys, M.: Pixel-perfect structure-from-motion with featuremetric refinement. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
26. Sengupta, S., Amir, T., Galun, M., Goldstein, T., Jacobs, D.W., Singer, A., Basri, R.: A new rank constraint on multi-view fundamental matrices, and its application to camera location recovery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2413–2421 (2017)
27. Shafarevich, I., Hirsch, K.: Basic algebraic geometry, vol. 1. Springer (1994)
28. Sinha, S., Pollefeys, M., McMillan, L.: Camera network calibration from dynamic silhouettes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. I–I (2004)
29. Sitharam, M., St. John, A., Sidman, J.: Handbook of Geometric Constraint Systems Principles. Discrete Mathematics and Its Applications, CRC Press (2018)
30. Trager, M., Hebert, M., Ponce, J.: The joint image handbook. In: Proceedings of the International Conference on Computer Vision. pp. 909–917 (2015)
31. Trager, M., Osserman, B., Ponce, J.: On the solvability of viewing graphs. In: Proceedings of the European Conference on Computer Vision. pp. 335–350 (2018)
32. Tron, R., Carlone, L., Dellaert, F., Daniilidis, K.: Rigid components identification and rigidity enforcement in bearing-only localization using the graph cycle basis. In: IEEE American Control Conference. pp. 3911–3918 (2015)
33. Wilson, K., Bindel, D.: On the distribution of minima in intrinsic-metric rotation averaging. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6030–6038 (2020)
34. Wilson, K., Snavely, N.: Robust global translations with 1DSfM. In: Proceedings of the European Conference on Computer Vision. pp. 61–75 (2014)