Structure from Motion with Missing Data is NP-Hard

David Nistér
Microsoft Live Labs
Microsoft Research, USA
dnister@microsoft.com

Fredrik Kahl
Centre for Mathematical Sciences
Lund University, Sweden
fredrik@maths.lth.se

Henrik Stewénius
Center for Visualization and Virtual Environments
University of Kentucky, USA
hstewenius@gmail.com

Abstract

This paper shows that structure from motion is NP-hard for most sensible cost functions when missing data is allowed. The result provides a fundamental limitation of what is possible to achieve with any structure from motion algorithm. Even though there are recent, promising attempts to compute globally optimal solutions, there is no hope of obtaining a polynomial time algorithm unless P=NP.

The proof proceeds by encoding an arbitrary Boolean formula as a structure from motion problem of polynomial size, such that the structure from motion problem has a zero cost solution if and only if the Boolean formula is satisfiable. Hence, if there was a guaranteed way to minimize the error of the relevant family of structure from motion problems in polynomial time, the NP-complete problem 3SAT could be solved in polynomial time, which would imply that P=NP. The proof relies heavily on results from both structure from motion and complexity theory.

1. Introduction

Recently there has been a surge of interest in provably optimal Structure from Motion (SfM), see [1, 9, 13]. In recent developments, it has been shown that other cost-functions such as max-norm for the reprojection error can lead to tractable global optimization [15, 8, 10, 12]. For example, in [10] it was shown that when camera rotations are known, the camera translations and the 3D points can be recovered in a globally optimal manner using quasi-convex optimization. A more classical example of globally optimal structure from motion is the Tomasi-Kanade algorithm [18], which can recover least squares optimal structure from motion under the affine camera model.

These results raise the question of under which conditions, such as camera model, error function or observation model, it is possible to find guaranteed algorithms for optimal structure from motion. This paper shows that when missing data is allowed, optimal structure from motion is NP-hard.

Most of the proven NP-complete problems have a combinatorial flavor. For example, it has been shown that global minimization of even the simplest discontinuity-preserving energy functions for stereo is NP-hard [3]. Structure from motion appears different in nature since the optimization is over a continuous domain. Yet, we show that when missing data [13] is allowed, structure from motion is NP-hard for most sensible cost-functions, and camera models. Here, missing data refers to that 3D points are not always observed in all cameras. The proof relies on exhibiting combinatorical structure within the structure from motion problem, which allows expressing Boolean formulas in terms of structure from motion problems.

Due to space constraints, we will have to assume that the reader is quite familiar with Structure from Motion as well as complexity theory. We direct readers to [7, 16] and [17, 6, 2], respectively, for background material on these topics. As is customary, we will keep the proofs on a more abstract level than that of programming Turing machines, trying to strike a balance between rigor and readability.

2. Preliminaries

In the language of complexity theory, a 'problem' actually intends a family of instances of decision problems with a yes/no answer. The size of an instance is measured in terms of its description length. The class P is the class of decision problems that can be solved in polynomial time by a deterministic Turing machine. The class NP is the class of decision problems that can be solved in polynomial time by a non-deterministic Turing machine. The class NP-complete is the class of decision problems in NP that are such that all problems in NP can be reduced to them in polynomial time. SATISFIABILITY (or SAT for short) was the first NP-complete problem established. It is the problem of deciding whether there is a variable assignment such that a Boolean expression is satisfied. It was proven by Cook [4] to be NP-complete. The proof establishes that it can be decided by a Boolean expression of polynomial size whether a string encodes an accepting computation path of a non-deterministic Turing machine. The class NP-hard applies to more general problems than decision problems. It is the class of problems such that all problems in NP can be reduced to them in polynomial time. To show that a problem is NP-hard, one only needs to reduce a known NP-hard problem to it in polynomial time. Karp [11] used the reduction technique to show that a number of decision problems, including the travelling salesman, are NP-complete. A plethora of problems has since then been proven NP-complete [6] including optimization problems over a continuous domain, for example, [5].

In this paper we will consider families of Structure from Motion problems, described in terms of reprojected image points. The reprojected image points are described in homogeneous coordinates [16], where each coordinate is a rational number. The problem size is then measured in terms of the encoding length of the rational coordinates. Note that it is reasonable to assume that only 3D points and cameras that are involved in at least one of the reprojections are part of the problem.

3. Main Result

The SfM problem consists of determining a set of 3D points and cameras such that the residuals between the reprojected and the (given) measured points are as small as possible. We will consider several different error norms on the residuals for this optimization problem. More formally, the problem can be stated as follows.

Problem (Structure from Motion). *Let m denote the number of point features and n the number of images. Assume that a set of feature correspondences $\{ x_{ij} \}_{(i,j)\in I}$ is given, where each x_{ij} denote the coordinates of the projection of point i in image j and $I \subset \mathbb{Z}^2$ denotes an index set. Some points may be missing (occluded) in some images. The Structure from Motion problem consists of finding 3D points X_i, $i = 1, \dots, m$ and camera matrices P_j, $j = 1, \dots, n$ such that the norm of all the residuals*

$$\|[\dots, r_{ij}, \dots]^\top\|$$

is minimized. Here r_{ij} is the residual vector of image differences between point x_{ij} and its reprojection $P_j X_i$.

Most often the 2-norm is used since it results in a standard nonlinear least-squares problem. Under the assumption of independent, Gaussian noise on the measured image points, it also gives an ML-estimate. However, our results are true for any norm. We will make use of the following norm properties: $||x|| = 0$ if and only if $x = 0$ and $||x|| \geq 0$ throughout the paper.

Our main result of the paper may be stated as follows.

Theorem 1 *There exists no polynomial time algorithm for solving the Structure from Motion problem with missing data based on a norm of reprojection residuals unless $P = NP$.*

The idea of our proof is quite simple: We show that any Boolean formula can be coded by a SfM problem instance and that the Boolean formula is satisfiable if and only if the globally optimal solution to the SfM problem has zero reprojection error. Thus, our strategy is similar to other approaches for showing NP-hardness; we apply polynomial reduction (or transformation) from a known NP-complete problem - in this case, the SAT problem - to the SfM problem. Hence, if somebody was able to come up with polynomial time algorithm for globally solving the SfM problem, then one could apply the same algorithm for solving the SAT problem in polynomial time. Since SAT is known to be NP-complete, all problems in NP could be solved in polynomial time if this were the case. Hence, this leads to a contradiction unless $P = NP$.

In order to code a Boolean formula using a SfM problem instance, several building blocks are needed. In the next four sections (Sections 4 to 7), we will develop these components, and then in Section 8 we will show how to put all the pieces together.

4. Anchoring Frame

All the sub-configurations that represent variables, truth-values, gates and variable transfers (all components of a boolean formula) are anchored to a common anchor frame in order to make sure that their relation is fixed and that the whole configuration stays fixed up to a common overall projective transformation. Almost any configuration will do as anchor frame, as long as it is general enough to ensure a *unique* projective reconstruction. For example, we may use the 3D points with homogeneous coordinates as the columns

$$X_{anchor} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad (1)$$

and the camera matrices

$$P^1_{anchor} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P^2_{anchor} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$P^3_{anchor} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \quad (2)$$

For the moment, we will ignore the problem of handling image points at the line at infinity with a standard image metric. From the image projections, the 3D structure and the camera motion can be reconstructed uniquely modulo projective coordinate system. The first five 3D points consist of the canonical projective coordinates in space, which can be thought of as fixing the coordinate system for the whole reconstruction. The 3D points also include the canonical projective coordinates within the plane at infinity (points one, two, three and nine), which will be used to fix the leading 3×3 sub-matrix of a camera matrix. In a similar way, points one, two, four and ten provide a projective basis for the xy-plane. Note that the three cameras in the anchor frame are enough to triangulate all points in space uniquely, and the ten 3D points are enough to determine the pose of any new camera.

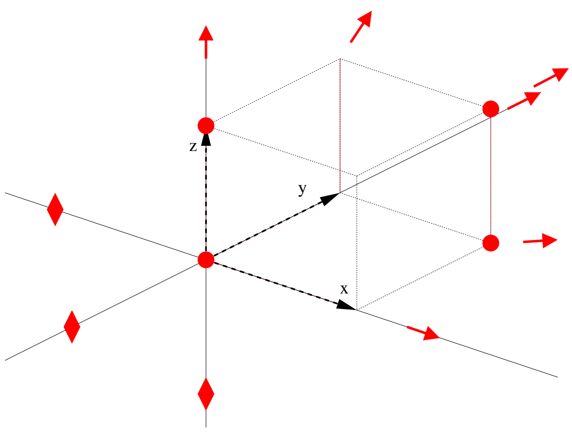


Figure 1. The anchoring frame with cameras ♦, points at finite coordinates ● and points at infinity ⟶. All cameras observe all points and the solution is unique up to a projective transformation. Points (cameras) not belonging to the anchoring frame can be locked down by being observed by (observing) cameras (points) beloning to the frame. In particulat we use the points at the plane at infinity to lock the rotation of cameras without restricting their position.

5. A Two-State Configuration

Our goal in this section is to create a SfM problem that has two and only two perfect solution configurations (up to projective ambiguity). Moreover, we require that all the solutions can be described with integer points and cameras. This will be used as a vehicle that enables us to create 'binary variables' as well as 'NAND-gates' in our SfM encoding of a Boolean formula.

We will, however, start by deriving a tri-state configuration from the well known problem of projective reconstruction from two uncalibrated views seeing seven points. This is known to have exactly three solutions up to projective ambiguity, and we will take care to construct a case where all the solution configurations can be described entirely in integers. Our motivation for this is to ensure that our solutions remain in the parameter space even when we restrict the parameter space to the rationals $\mathbb{Q}$ or integers $\mathbb{Z}$.

We start by noting that the three solutions for fundamental matrices come from a two-dimensional nullspace (created after putting seven linear point constraints on the fundamental matrix). This is really the only constraint, so if we have two distinct fundamental matrices and create the third as a linear combination of the first two, we obtain three fundamental matrices that can arise as solutions to a seven-point problem. For example, we may choose the three diagonal matrices

$$F_A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, F_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, F_C = F_A - F_B.$$

It can be seen that the point correspondences $x \leftrightarrow x'$ that agree with these fundamental matrices are those for which

$$-x_1 x_1' = x_2 x_2' = x_3 x_3', \quad (3)$$

or in other words

$$x = \begin{bmatrix} a & b & c \end{bmatrix}^\top \leftrightarrow x' = \begin{bmatrix} -bc & ac & ab \end{bmatrix}^\top \quad (4)$$

for some choices of a, b, c. We choose the following seven image point correspondences

$$\begin{matrix} \begin{bmatrix} -1 & 1 & 1 \end{bmatrix}^\top & \leftrightarrow & \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top \\ \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}^\top & \leftrightarrow & \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}^\top \\ \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}^\top & \leftrightarrow & \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}^\top \\ \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top & \leftrightarrow & \begin{bmatrix} -1 & 1 & 1 \end{bmatrix}^\top \\ \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^\top & \leftrightarrow & \begin{bmatrix} -6 & 3 & 2 \end{bmatrix}^\top \\ \begin{bmatrix} 2 & 1 & 1 \end{bmatrix}^\top & \leftrightarrow & \begin{bmatrix} -1 & 2 & 2 \end{bmatrix}^\top \\ \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^\top & \leftrightarrow & \begin{bmatrix} -2 & 1 & 2 \end{bmatrix}^\top \end{matrix}. \quad (5)$$

It is straightforward to verify that these point correspondences result in exactly the solutions F_A, F_B and F_C for the fundamental matrix. A projective reconstruction for a particular choice of projective coordinate frame is given in Table 1. Other frames can simply be generated by applying a projective transformation.

Solution 1:

$$P_A^1 = \begin{bmatrix} 2 & 3 & 2 & 0 \\ 1 & 1 & 6 & 0 \\ 1 & 1 & 4 & 0 \end{bmatrix}, \quad P_A^2 = \begin{bmatrix} -1 & -1 & -6 & 0 \\ 2 & 3 & 2 & 0 \\ 3 & 4 & 4 & -1 \end{bmatrix}, \quad X_A = \begin{bmatrix} -4 & 12 & -14 & 2 & -78 & 1 & -6 \\ 3 & -7 & 9 & -1 & 48 & 0 & 4 \\ 0 & -1 & 1 & 0 & 3 & 0 & 1 \\ -1 & 5 & -1 & 1 & -26 & 1 & -2 \end{bmatrix}.$$

Solution 2:

$$P_B^1 = \begin{bmatrix} 10 & 15 & 18 & 1 \\ 5 & 6 & 30 & 2 \\ 10 & 11 & 26 & -3 \end{bmatrix}, \quad P_B^2 = \begin{bmatrix} -10 & -11 & -26 & 3 \\ 17 & 22 & 22 & -6 \\ 10 & 15 & 18 & 1 \end{bmatrix}, \quad X_B = \begin{bmatrix} -4 & 12 & -14 & 2 & -78 & 0 & -14 \\ 3 & -7 & 9 & -1 & 48 & 1 & 8 \\ 0 & -1 & 1 & 0 & 3 & 0 & 3 \\ -1 & 5 & -1 & 1 & -26 & 1 & -2 \end{bmatrix}.$$

Solution 3:

$$P_C^1 = \begin{bmatrix} 2 & 3 & 18 & 2 \\ 1 & 3 & 6 & 4 \\ 11 & 13 & 16 & -6 \end{bmatrix}, \quad P_C^2 = \begin{bmatrix} -7 & -8 & -8 & 3 \\ 11 & 13 & 16 & -6 \\ 1 & 3 & 6 & 4 \end{bmatrix}, \quad X_C = \begin{bmatrix} -4 & 12 & -14 & 2 & -78 & 0 & -22 \\ 3 & -7 & 9 & -1 & 48 & 0 & 12 \\ 0 & -1 & 1 & 0 & 3 & 1 & 1 \\ -1 & 5 & -1 & 1 & -26 & 1 & -10 \end{bmatrix}.$$

Table 1. *A projective realization of a tri-state configuration for the image correspondences in (5).*

The first five 3D points can now be used to fix the coordinate system by making them visible in the three cameras of the anchor frame (2), while the three different states

$$S_A = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^\top \tag{6}$$

$$S_B = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^\top \tag{7}$$

$$S_C = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^\top, \tag{8}$$

of the sixth point can be used to represent a variable.

Restricting the tri-state configuration in Table 1 in order to obtain a two-state configuration is straightforward. Introduce another camera with its camera centre somewhere on the line spanned by S_A and S_B, and attach it to the anchoring frame by making the anchor points (1) visible. Finally, add the projection point x_{AB} of S_A and S_B, which is, by construction, a single image point to this view. This invalidates the third state since the projection of the sixth point S_C in the C-configuration results in non-zero reprojection cost. This constrains X to

$$S_A = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^\top, \tag{9}$$

$$S_B = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^\top. \tag{10}$$

Various variables' coordinate systems can be fused together, simply by making sure that the cameras from the anchor frame triangulate the first five 3D points of each coordinate system uniquely. When this is done, it ensures that all sub-configurations are joined together uniquely in a zero-cost reconstruction, up to a common projective transformation.

6. Transferring Variables

Suppose we wish to transfer variables, that is, we have a variable representation in terms of a point X residing in one of the two positions X_a, X_b, and we need to constrain a point Y, which is already known to be either in position Y_a or Y_b, in such a way that Y is at Y_a if and only if X is in X_a and Y is at Y_b if and only if X is in X_b. We assume that neither of the four point positions is at infinity (none of their fourth coordinates is zero). Moreover, we assume that the four points are not coplanar (if they were, the transfer could easily be done in two steps).

To accomplish the transfer, we simply need to introduce an additional camera that sees the points X and Y at carefully selected image points. We introduce a new camera P and fix its leading 3×3 sub-matrix to identity by observing the points at infinity from the anchoring frame (1) at the same image directions as their respective directions at infinity. We also observe the projection of point X in P at the image point x, which must be distinct from the direction of the line between X_a and X_b (because we wish to avoid the possibility of them both projecting to x at the same time). Note that this constrains the camera centre of P to lie on a line with direction x through either X_a or X_b. In a similar way, we also observe the 3D point Y in P at the image point y, which must be distinct from the direction of the line between Y_a and Y_b. This constrains the camera centre of P to a line with direction y through either Y_a or Y_b. A configuration is now possible if and only if the directions between X and Y is coplanar with the directions x and y. We wish for this to be the case when and only when we have $X = X_a, Y = Y_a$ *or* $X = X_b, Y = Y_b$. Let the directions (represented by 3-vectors) between X_a, Y_a and X_b, Y_b be denoted by D_a and D_b, respectively. To accomplish our mission, we may select the homogenous image coordinates x and y as

$$x = D_a + D_b, \tag{11}$$

$$y = D_a - D_b. \tag{12}$$

By construction, there are two possible camera matrices P_a and P_b for the camera matrix P. Both solutions have the leading 3×3 sub-matrix equal to identity and the properties

$$P_aX_a \sim P_bX_b \sim x \text{ and } P_aY_a \sim P_bY_b \sim y.$$

An illustration is given in Figure 2. For this two-state construction to work, one needs to check that we do not have $PX_a \sim x$, $PX_b \sim x$ for neither $P = P_a$ nor $P = P_b$.

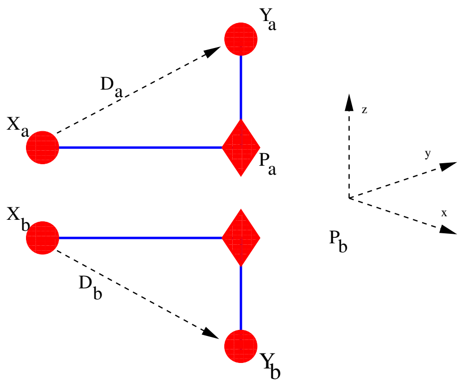
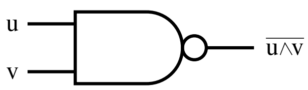


Figure 2. An illustration of the transfer component. Circles are possible point locations and diamonds are possible camera locations. See text for details. By construction, X_a has a truth-value if and only Y_a has a truth-value, similarly for X_b and Y_b. Either configuration a or b yields a zero cost reconstruction, but not both.

7. NAND-Gate



	0	1
0	1	1
1	1	0

Figure 3. Circuit board symbol for NAND gate along with tabulated response

Our construction of a NAND-gate is illustrated in Figure 4. We assume that we have an in-variable u represented by two possible positions for a point X_u, such that $X_u = \begin{bmatrix}-2 & -1 & 1 & 1\end{bmatrix}^\top$ when u is true, and $X_u = \begin{bmatrix}-2 & -1 & 0 & 1\end{bmatrix}^\top$ otherwise. We also assume that we have an in-variable v represented by two possible positions for a point X_v, such that $X_v = \begin{bmatrix}-1 & -1 & 2 & 1\end{bmatrix}^\top$ when v is true, and $X_v = \begin{bmatrix}0 & -1 & 2 & 1\end{bmatrix}^\top$ otherwise. This can be accomplished with the variable transfer method described above.

Furthermore, two cameras P_u and P_v are introduced. We fix the leading 3×3 sub-matrix of P_u to identity by observing the points one, two, three and nine - all at infinity - in the anchor frame (1). Camera P_v is also constrained, but in a slightly different way. The procedure of observing four points at infinity from the anchor frame at corresponding image locations makes sure that the backprojection from a particular image point always pierces the plane at infinity at the same location, regardless of what the remaining parameters of the camera are. We wish to constrain P_v in a similar way, but instead use the xy-plane. This will make sure that the backprojection from a particular image point will pierce the xy-plane in a fixed point. To this end, we make P_v observe the points one, two, four and ten of the anchor frame (1) all lying in the xy-plane. This constrains P_v to the form

$$\begin{bmatrix}1 & 0 & t_1 & 0\\ 0 & 1 & t_2 & 0\\ 0 & 0 & t_3 & 1\end{bmatrix}. \tag{13}$$

The camera P_u also observes X_u at the image point $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^\top$, as well as the additional point $\begin{bmatrix}-2 & 0 & 2 & 1\end{bmatrix}^\top$ at $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$. Note that this constrains P_u to one of two locations (determined by X_u), with camera center at $\begin{bmatrix}-2 & 0 & 1 & 1\end{bmatrix}^\top$ or $\begin{bmatrix}-2 & 0 & 0 & 1\end{bmatrix}^\top$. We also make P_u observe a new point X_G at $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top$, which constrains X_G to two parallel lines in the direction of the x-axis.

The camera P_v observes X_v at the image point $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^\top$ as well as the additional point $\begin{bmatrix}1 & 0 & 2 & 1\end{bmatrix}^\top$ at $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top$. This constrains P_v to one of two locations (determined by X_v), with camera center at $\begin{bmatrix}-1 & 0 & 2 & 1\end{bmatrix}^\top$ or $\begin{bmatrix}0 & 0 & 2 & 1\end{bmatrix}^\top$. We also make P_v observe the point X_G at $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$. This constrains X_G to two lines, but they are not parallel. Instead they converge on the origin because of the way we constrained P_v.

The point X_G is now constrained to one of the three locations, namely: $\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^\top$, $\begin{bmatrix}0 & 0 & 1 & 1\end{bmatrix}^\top$, or $\begin{bmatrix}-1 & 0 & 2 & 2\end{bmatrix}^\top$ where the last location occurs if and only if $u \wedge v$ is true. We now tap this information into the location of a camera $P_{\overline{u\wedge v}}$. This is accomplished by observing points at infinity of the anchor points (1), and hence making the 3×3 sub-matrix of $P_{\overline{u\wedge v}}$ equal to identity, as well as letting $P_{\overline{u\wedge v}}$ observe X_G at $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$ and an additional point $\begin{bmatrix}-1 & 0 & -1 & 1\end{bmatrix}^\top$ at $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top$. This constrains the projection center of $P_{\overline{u\wedge v}}$ to either $\begin{bmatrix}0 & 0 & -1 & 1\end{bmatrix}^\top$ or $\begin{bmatrix}-1 & 0 & -2 & 2\end{bmatrix}^\top$, depending on whether $\overline{u\wedge v}$ is true or false, respectively. Finally, we tap the output into a point $X_{\overline{u\wedge v}}$, by letting $P_{\overline{u\wedge v}}$ observe $X_{\overline{u\wedge v}}$ at $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^\top$, as well as adding the following fixed camera (relative the anchor frame)

$$\begin{bmatrix}1 & 0 & 0 & 1\\ 0 & 1 & 0 & -1\\ 0 & 0 & 1 & 1\end{bmatrix}, \tag{14}$$

which observes $X_{\overline{u \wedge v}}$ at $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top$. This results in $X_{\overline{u \wedge v}} = \begin{bmatrix}0 & 1 & -1 & 1\end{bmatrix}^\top$ or $X_{\overline{u \wedge v}} = \begin{bmatrix}-1 & 2 & -2 & 2\end{bmatrix}^\top$ depending on whether $\overline{u \wedge v}$ is true or false.

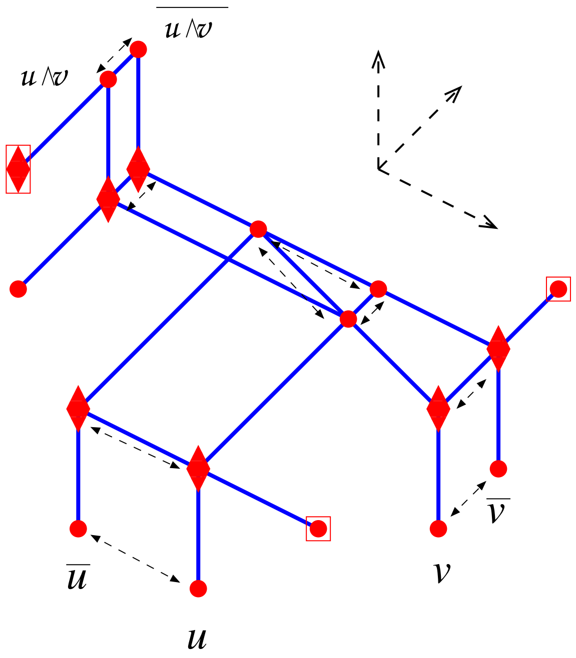


Figure 4. An illustration of the NAND-gate. Circles are possible point locations and diamonds are possible camera locations, a box means that this specific camera or point only has one possible location. Variables come in as two point position pairs $u \leftrightarrow \overline{u}$ and $v \leftrightarrow \overline{v}$. They constrain two cameras to two locations each. The cameras in their turn constrain a point to one of three possible locations in the middle of the NAND-gate. This point is seen by another camera, which is constrained to one of two locations, representing $\overline{u \wedge v}$ and its negation. This out-signal from the NAND-gate is finally transfered onto the location of a point. Pairs of point positions can be positioned correctly for input and output to the NAND-gate using variable transfer.

8. Collecting the Components Into a Boolean Formula

We now have all the components necessary to encode a Boolean formula in terms of a SfM problem.

For each formula, we use a single anchor frame to which all the other components are anchored. All the Boolean variables are represented with a two-state configuration each. The two-state configurations are linked into the NAND-gates with variable transfer components. Note that the NOT operation is most efficiently implemented by just switching the two point locations in the variable transfer. The outputs from the NAND-gates can also be transferred in a similar manner and be used as input to other NAND-gates.

Note also that by switching the two output point locations, a NAND-gate can just as easily be used as AND-gate. As is well known, all Boolean expressions can be written solely in terms of NAND-gates. Hence we can encode any Boolean expression by connecting together the various components (although it is important to notice that the only thing actually encoded are the reprojected image points and which camera and 3D point they relate to). The output from the final NAND-gate representing the whole expression can be required to be TRUE by observing the corresponding point location with one of the anchor cameras (2). Our labor results in:

Theorem 2 *Assume that a SfM problem is built according to a Boolean formula with the components described above for anchoring the coordinate frame, variable construction, variable transfer and the NAND-gate. If the formula is satisfiable, the SfM problem has a non-zero but finite number of exact zero-cost solutions. More precisely, the structure from motion problem has exactly as many zero-cost solutions as there are satisfying variable assignments of the formula. Moreover, these solutions can be described with integer parameters. If the formula is not satisfiable, the structure from motion problem has no exact zero-cost solutions.*

9. Proving the NP-Hardness

We have shown that we can transfer any formula from the SAT problem into the SfM problem. However, we wish to limit the signal depth of the formula, that is the number of times an output signal is used again as input to a new NAND-gate. Luckily, we can do just that without any loss of generality, by using a reduction from 3SAT. 3SAT is also known to be NP-complete [6] and consists of the problem of satisfying a Boolean formula in conjunctive normal form with only three literals per clause, i.e. a formula such as

$$(\overline{u_i} \vee u_j \vee \overline{u_k}) \wedge \ldots \wedge (\overline{u_w} \vee \overline{u_j} \vee u_k). \qquad (15)$$

Each clause $(u_i \vee u_j \vee u_k)$ can be encoded according to

$$\overline{(\overline{u_i} \wedge \overline{u_j}) \wedge \overline{u_k}}, \qquad (16)$$

which is as simple as two gates (note again that any required negations come for free in the variable transfers). Each clause has an output-gate that is required to return TRUE, and rather than collecting all these signals together, we simply require each one to be TRUE by observing the representing 3D point with a camera in the anchor frame. The signal depth is then limited to two gates with associated transfers. Hence, the polynomial reduction to the SfM problem is indeed polynomial.

There is one detail we have ignored so far that may cause some problems (or least, irritation by a punctilious reader).

In our construction of a SfM problem instance, there are image points at the line at infinity of the image plane. Our image metric cannot actually measure these points since they are not part of the real plane. But this is easily corrected for by changing the image coordinate system (by a homography) such that no points will appear at infinity, hence pushing back all image points to the standard real plane. Another detail - which we will continue to ignore - is that our SfM construction makes no guarantees that a zero-cost reconstruction actually obeys all chirality conditions, that is, that all points are in front of all cameras.

This concludes our proof that the SfM problem with missing data is NP-hard.

10. Conclusions

Strictly speaking, we have proved that projective structure from motion with missing data is NP-hard for any sensible cost function. We leave proofs for other settings to further study, but conjecture that similar proofs can be created for most camera models and settings, for example using the four-fold or eight-fold ambiguity in calibrated pose (depending on whether orientation is enforced), or the reflection symmetries arising in affine structure from motion. Other relevant questions that we leave for further research are whether structure from motion remains NP-hard even if we demand that a unique solution exists, or whether structure from motion under perspective is NP-hard even without missing data.

Acknowledgments

This work has been supported by the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMErobot^{TM}, Swedish Foundation for Strategic Research (SSF) through the programme Vision in Cognitive Systems II (VISCOS II), Swedish Research Council through grants no. 2004-4579 'Image-Based Localisation and Recognition of Scenes' and no. 2005-3230 'Geometry of multi-camera systems'.

References

[1] S. Agarwal, M. Chandraker, F. Kahl, D. Kriegman and S. Belongie, Practical Global Optimization for Multiview Geometry, *European Conference on Computer Vision (ECCV 2006)*, Graz, Austria, 2006.

[2] L. Blum, F. Cucker, M. Shub, S. Smale, *Complexity and Real Computation*, Springer-Verlag, New York, ISBN 0-387-98281-7, 1998.

[3] Y. Boykov, O. Veksler, R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 11, pp. 1222-1239, 2001.

[4] S. Cook, The Complexity of Theorem Proving Procedures, *Proceedings of the third annual ACM symposium on Theory of computing*, 151-158, 1971.

[5] R. W. Freund and F. Jarre, Solving the Sum-of-Ratios Problem by an Interior-Point Method *Journal of Global Optimization*, vol. 19, pp. 83-102, 2001.

[6] M. Garey and D. Johnson, *Computers and Intractability*, New York, Freeman, 1979.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN 0-521-62304-9, 2000.

[8] R. Hartley and F. Schaffalitzky, L_∞ Minimization in Geometric Reconstruction Problems, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, Washington DC, June, 2004.

[9] F. Kahl, D. Henrion, Globally Optimal Estimates for Geometric Reconstruction Problems, *International Conference on Computer Vision (ICCV 2005)*, Beijing, China, 2005.

[10] F. Kahl, Multiple View Geometry and the L_∞-Norm, *International Conference on Computer Vision (ICCV 2005)*, Beijing, China, 2005.

[11] R.M. Karp, Reducibility Among Combinatorial Problems, *In Complexity of Computer Computations*, Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y.. New York: Plenum, p.85-103. 1972.

[12] Q. Ke and T. Kanade, Quasiconvex Optimization for Robust Geometric Reconstruction, *IEEE International Conference on Computer Vision (ICCV 2005)*, Beijing, China, October 2005.

[13] Q. Ke and T. Kanade, Robust L1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, CA, June 2005.

[14] J. Munkres, *Topology*, 2nd Edition, Prentice Hall, ISBN 0-13-181629-2, 2000.

[15] D. Nistér, *Automatic Dense Reconstruction from Uncalibrated Video Sequences*, PhD Thesis, Royal Institute of Technology KTH, Stockholm, Sweden, ISBN 91-7283-053-0, March 2001.

[16] J. Semple and G. Kneebone, *Algebraic Projective Geometry*, ISBN 0-19-8503636, Oxford University Press, 1952.

[17] M. Sipser, *Introduction to the Theory of Computation*, 2nd Edition, Thomson Course Technology, ISBN 0-534-95097-3, 1996.

[18] C. Tomasi and T. Kanade, Shape and Motion from Image Streams Under Orthography: a Factorization Method, *International Journal of Computer Vision (IJCV)*, 9(2):137-154, November 1992.