Classifying and solving minimal structure and motion problems with missing data*

Magnus Oskarsson, Kalle Åström, Niels Chr. Overgaard
Centre for Mathematical Sciences, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden
{magnuso,kalle,nco}@maths.lth.se

Abstract

In this paper we investigate the structure and motion problem for calibrated one-dimensional projections of a two-dimensional environment. In a previous paper the structure and motion problem for all cases with non-missing data was classified and solved. Our aim is here to classify all structure and motion problems, even those with missing data, and to solve them. Although our focus here is on one-dimensional retina, the classification part works equally well for ordinary cameras, and we give some results for those as well.

1. Introduction

Understanding of one-dimensional cameras is important in several applications. In [12] it was shown that the structure and motion problem using line features in the special case of affine cameras can be reduced to the structure and motion problem for points in one dimension less, i.e. one-dimensional cameras. Thus solution to 1D structure and motion problems have been used to solve structure and motion problems for lines, [12, 3].

Another area of application is vision for planar motion. It is shown that ordinary vision (two-dimensional retina) can be reduced to that of one-dimensional cameras if the motion is planar, i.e. if the camera is rotating and translating in one specific plane only, cf. [7]. In another paper the planar motion is used for auto-calibration [1]. A typical example is the case where a camera is mounted on a vehicle that moves on a flat plane or flat road.

Our personal motivation, however, stems from the *autonomous guided vehicles*, called *AGV*, which are important components for factory automation. The navigation system uses strips of inexpensive reflector tape, *beacons*, which are put on walls or objects along the route of the vehicle, cf. [10]. The *laser scanner* measures the direction from the vehicle to the beacons, but not the distance. This is the information used to calculate the position of the vehicle.

One of the primary vision problems (both 1D and 2D retina) is the so called structure and motion problem. For AGV's this is the procedure to obtain a map of the unknown positions of the beacons using images at unknown positions and orientations. This is usually done off-line, once and for all, when the system is installed and then occasionally if there are changes in the environment. High-accuracy is needed, since the map has to be hard-coded in the system. The performance of the navigation routines depends on the precision of the surveyed map.

The overall goal of this work is to solve all solvable structure and motion problems. The purpose of this paper is twofold. Firstly, tools are developed to classify the minimal structure and motion problems with missing data. Tentative results are given both for 1D retina vision, but also for the ordinary camera. Secondly, solutions to the structure and motion problem are given for some of these minimal problems.

2. Problem formulation

One example of sensors with one-dimensional retina is a laser scanner as used in laser navigated vehicles. This sensor can be shown, cf. [4], to follow the well known projection equation:

$$\lambda \mathbf{u} = \mathbf{P}\mathbf{U} \,. \tag{1}$$

Here the camera matrix is calibrated, i.e. it has the following form:

$$\mathbf{P} = \begin{bmatrix} a & b & c \\ -b & a & d \end{bmatrix} . \tag{2}$$

It is sometimes useful to consider dual image coordinates, $\mathbf{v}$, so that $\mathbf{v}\mathbf{u} = 0$. With $\mathbf{p} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$ and

$$D(\mathbf{U}) = \begin{pmatrix} X & Y & Z & 0 \\ Y & -X & 0 & Z \end{pmatrix}$$

*This work has been supported by the Swedish Research Council (Vetenskapsrådet), project 221-2000-476.

equation (1) can be written as

$$0 = \mathbf{vPU} = \mathbf{v}D(\mathbf{U})\mathbf{p}\,. \qquad (3)$$

Motivated by the previous discussion the structure and motion problem will now be defined.

Problem 2.1. *Given some of the bearings to n beacons from m different positions $\mathbf{u}_{I,J}$, $(I,J) \in \mathbb{I}$, where $\mathbb{I}$ is an index set representing which beacons J are visible from image number I. The* **surveying problem** *is to find the depths $\lambda_{I,J} > 0$, the reconstructed points $\mathbf{U}_J$ and the camera matrices $\mathbf{P}_I$ such that*

$$\lambda_{I,J}\mathbf{u}_{I,J} = \mathbf{P}_I\mathbf{U}_J, \qquad \forall(I,J) \in \mathbb{I}\,.$$

In this paper the interest lies in classifying such problems. As such we will consider the problem with both beacons and cameras in general positions. The question whether a structure and motion problem is well-defined or perhaps even over-constrained depends on the structure of the index set $\mathbb{I}$.

In a previous paper [4] we considered only the cases where all beacons are visible in all views. The conclusion there is that the structure and motion problem is well-defined if and only if there are at least 3 views of at least 4 beacons, excluding the case of 3 views of 4 images.

If it is possible to solve a case with a subset of cameras and beacons, then it is relatively easy to extend that solution to other cameras and points by well known techniques called resection and intersection, [4]. Thus the solution of any well-defined case above are based on the only two minimal cases with non-missing data, i.e. 4 views of 4 beacons and 3 views of 5 beacons.

The goal of this paper is to repeat this for the case of missing data. Depending on the index set $\mathbb{I}$ a structure and motion problem can be either

- ill-defined, if there is not, in general, enough data to constrain all unknown variables.
- well-defined and minimal, if there is exactly enough data to constrain the unknown variables (up to a discrete number of solutions).
- well-defined but over-constrained, if there is more than enough data to constrain the unknown variables.

Some of the minimal cases contain a minimal case as a subproblem. An example of this is the case with four points seen in five images, but where the fourth point is missing from the fifth image. It is minimal, but contains a subproblem (the problem with the first four views only) which is well defined and minimal. We will use the notation **prime problem** for a minimal problem which does not contain a well defined minimal problem as a subproblem. A minimal but not prime problem may in some cases be solved by first solving an underlying prime subproblem and then extend this solution to the solution of the minimal problem using resection and intersection. In other cases the prime problem may be embedded in the minimal problem in a more complicated manner. We first observe that similar to the case of non-missing data a well-defined but over-constrained problem contains as a subset a problem which is well-defined but minimal. Thus by finding the minimal cases and solving them, we should be able to solve all well-defined problems.

As the classification is based on the index set $\mathbb{I}$ alone, it is interesting to study these sets. In this paper we consider these sets as binary matrices, visibility matrices, A of size $m \times n$ where black denotes missing data and white denotes a measurement beacon which is present. In the paper we will use the notation $|\mathbb{I}|$ to denote the number of elements in the set $\mathbb{I}$.

3. Conditions on prime problems

In this section we will give some of the conditions for prime problems, i.e. well defined minimal problems, which do not contain subproblems which are minimal.

A first characterization of such problems is that they contain exactly the same number of equations as unknowns. Each object point has two degrees of freedom and each camera state has three. The solution is only defined up to a similarity transformation. This manifold has dimension 4. Using n points and m cameras we thus have $2n + 3m - 4$ degrees of freedom in the parameters. Each measured bearing gives one constraint on the estimated parameters. Thus for a problem with visibility index set $\mathbb{I}$ we have $|\mathbb{I}|$ equations. Thus prime problems have $|\mathbb{I}| = 2n + 3m - 4$. Since the maximum number of equations with m views of n points is mn it is easy to see how many measurements l that have to be occluded to obtain prime problems, $l = mn - (2n + 3m - 4)$. This number is shown in Table 1. We will use the notation **germ** for a configuration whose index set fulfills $|\mathbb{I}| = 2n + 3m - 4$. Notice that for the case where all points are visible in all images there are only two prime problems.

Another necessary condition for prime problems is that each point must be visible in at least 3 views and that at least 4 points must be visible in each view. If this is not the case then a point or a camera can be removed and the problem is still well-defined. Thus if the original problem is minimal and well defined, then the subproblem is also minimal and well defined. Using this condition it is straightforward to show that no further prime problems for the case of 3 views and no further prime problems for the case of 4 points exist.

An empirical method for determining whether a problem is minimal and well defined is to calculate the Jacobian of the bundle adjustment problem and study its singular val-

Table 1. The number of excess constraints $l = mn - (2n + 3m - 4)$ for the structure and motion problem with m images of n points.

l	n				
m	4	5	6	7	8
3	-1	0	1	2	3
4	0	2	4	6	8
5	1	4	7	10	13
6	2	6	10	14	18
7	3	8	13	18	23

ues. We have used this technique to empirically single out interesting problems.

4. Equivalence classes of index sets

The labeling of the cameras and of the beacons are of no consequence to the structure of the problem under study. Two index sets are considered equivalent if one results from the other by suitable relabelings. The problem addressed in this section is to count the number of index sets which may be considered essentially different, that is, non–equivalent in the above sense.

Definition 4.1. An index set $\mathbb{I}$ is said to be of type (m, n, l) if it represents a situation with m images and n points, in which exactly l points are not visible in all of the images, that is, if $|\mathbb{I}| = mn - l$.

From this definition it is clear that an index set $\mathbb{I}$ of type (m, n, l) can be represented by a binary $m \times n$–matrix $A = (a_{IJ})$ with $a_{IJ} = 1$ if $(I, J) \in \mathbb{I}$, and $a_{IJ} = 0$ otherwise, and such that $\sum_{IJ} a_{IJ} = mn - l$. The possible index sets of type (m, n, l) are thus in one–to–one correspondence with the set

$$M(m, n, l) = \{A \in \mathrm{Mat}_{m\times n}(\mathbb{Z}_2) : \textstyle\sum_{IJ} a_{IJ} = mn - l\}.$$

Let S_k denote the group of permutations on k symbols. With each permutation $\sigma \in S_k$ is associated a $k \times k$–permutation matrix $(\delta_{i\sigma(j)})$, which will be denoted simply by σ.

Definition 4.2. Two $m \times n$–matrices A and B are said to be *permutation equivalent*, if there exist permutations $\sigma \in S_m$ and $\tau \in S_n$ such that $B = \sigma^T A \tau$. If A and B are permutation equivalent then we write $A \sim B$.

The notion of equivalence of index sets can now be given a formal definition

Definition 4.3. Two index sets $\mathbb{I}$ and $\mathbb{I}'$ are called *equivalent*, and we write $\mathbb{I} \sim \mathbb{I}'$, if their corresponding matrix representations are permutation equivalent.

The relation $\sim$ is easily seen to be an equivalence relation. It follows that $M(m, n, l)$ (or the corresponding index sets) can be partitioned into equivalence classes $M_1, \ldots, M_\omega$ of matrices (or index sets). The number of essentially different index sets is thus seen to be exactly the same as the number $\omega = \omega(m, n, l)$ of equivalence classes.

Let the type (m, n, l) be fixed throughout the remainder of the discussion. To compute $\omega = \omega(m, n, l)$, notions and results from group theory will be used. Our reference here is to Section 3.6 of Fraleigh's text [8].

First, denote the product group $S_m \times S_n$ by G. Secondly, if $g = (\sigma, \tau) \in G$ and $A \in M = M(m, n, l)$, then a group action of G on M is defined by the formula

$$g \cdot A = \sigma^T A \tau. \tag{4}$$

Thus two matrices $A, B \in M$ satisfy $A \sim B$ if and only if there exists $g \in G$ such that $g \cdot A = B$. The equivalence classes $M_1, \ldots, M_\omega$ of $\sim$ correspond to **the orbits in M under the action of G**. Therefore ω can be computed by the following formula of Burnside; For any $g \in G$ let $M_g = \{A \in M : g \cdot A = A\}$ denote the set of matrices which are fix-points under action by g. Then

$$\omega = \frac{1}{|G|} \sum_{g\in G} |M_g|. \tag{5}$$

While (5) solves our problem in theory, there are still some practical problems to overcome. First, given $g \in G$, how do we compute $|M_g|$? Secondly, the sum $\sum_{g\in G} |M_g|$ must be evaluated, but as $|G| = m!n!$ becomes very large very quickly, the shear size of G may become an obstacle, unless the evaluation is performed cleverly.

A permutation $g = (\sigma, \tau) \in G$ may be regarded as an element of S_{mn}, as $A \mapsto \sigma^T A \tau$ permutes the mn entries of A. Let $g = g_1 g_2 \cdots g_s$ be the factorization in S_{mn} of g into a product of commuting (or disjoint) cyclic permutations. It is now easy to see that $A \in M_g$ if and only if, the entries in A, which equal zero, are arranged in such a manner, that any cycle g_i is either completely occupied by entries equal to zero, or contains no such entry at all. It follows that $|M_g|$ equals the number of ways in which l zeros can be allocated to $m \times n$ entries, such that the condition just described is satisfied. It is clear from this discussion that $|M_g|$ only depends on g's cycle structure (the number of cycles and their lengths).

Definition 4.4. If $\sigma \in S_k$ is a permutation in k symbols, let $n_i(\sigma), i = 1, \ldots, k$, denote the number of i–cycles in the factorization of σ into commuting cycles. The *cycle index of* σ is the polynomial

$$P_\sigma(x_1, x_2, \ldots, x_k) = x_1^{n_1(\sigma)} x_2^{n_2(\sigma)} \cdots x_k^{n_k(\sigma)}. \tag{6}$$

If $H < S_k$ is a (sub–)group of permutations, then the *cycle index of* H is the polynomial $P_H(x_1, x_2, \ldots, x_k) = |H|^{-1} \sum_{h\in H} P_h(x_1, x_2, \ldots, x_k)$.

Table 2. The number ω of distinct germs for different $(m, n, mn-(2n+3m-4))$.

	n				
m	4	5	6	7	8
3	0	1	1	3	6
4	1	3	16	62	225
5	1	16	155	1402	10162
6	3	79	1799	34045	505609

It follows from the theory developed in [14] that $|M_g| = (l!)^{-1}(d/dx)^l P_g(1+x, 1+x^2, \ldots, 1+x^{mn})|_{x=0}$, for any $g \in G$. This formula solves the first of our two problems. Furthermore, it follows from Burnside's formula (5) that

$$\omega = \frac{1}{l!}\left(\frac{d}{dx}\right)^l P_G(1+x, 1+x^2, \ldots, 1+x^{mn})\ \Big|_{x=0} \quad (7)$$

It turns out that the cycle index P_H is reasonably easy to compute when H is all of S_k. Now, $G = S_m \times S_n$ is a proper subgroup of S_{mn}, so in view of (7) our second problem above becomes: How do we compute P_G when the cycle indices of S_m and S_n are known? Again the authors of [14] provide the answer; They introduce a new operation beside the usual addition and multiplication, denoted $*$, on the ring of polynomials in the infinitely many variables $x_1, x_2, x_3, \ldots$, and with rational coefficients. The "product" is associative, commutative and distributive over both $+$ and $\cdot$, so it suffices to describe $*$ on monomial factors x_i^m and x_j^n, in which case

$$x_i^m * x_j^n = x_{[i,j]}^{imjn/[i,j]}, \quad (8)$$

where $[i,j]$ is the least common multiple of i and j. The authors of [14] then proceed to prove the following beautiful result, which we have used to compute P_G:

Theorem 4.1 (Wei and Xu). *If $H < S_m$ and $K < S_n$ are (sub–)groups, then $H \times K < S_{mn}$, and $P_{H\times K} = P_H * P_K$.*

Example The cycle index of S_3 is $\frac{1}{6}(x_1^3 + 3x_1x_2 + 2x_3)$ so if $G = S_3 \times S_3$ then

$$\begin{aligned} P_G &= \tfrac{1}{6}(x_1^3 + 3x_1x_2 + 2x_3) * \tfrac{1}{6}(x_1^3 + 3x_1x_2 + 2x_3) \\ &= \tfrac{1}{36}(x_1^9 + 6x_1^3x_2^3 + 9x_1x_2^4 + 12x_3x_6 + 8x_3^3) \end{aligned}$$

and it follows from (7) that $\omega(3,3,3) = 6$.

The procedure for calculating ω, described above, was implemented in Maple. Using this program we are able to compute ω for any given type (m, n, l). Table 2 contains the number of distinct germs ω for the first few types (m, n, l), with l given by table 1.

5. Finding equivalence classes

When solving the different minimal cases it is important to have representatives for the different equivalence classes of one minimal case, in order to be able to determine which are prime.

When finding the equivalence classes for different (m, n, l) it is enough to investigate configurations of type (n, n, n) from which other configuration easily can be derived. In the light of this we will concentrate our efforts on such configurations.

If one were to generate all index sets of type (n, n, n) there are $\binom{n^2}{n}$ such configurations. This is a very large number as n increases. If one has already calculated the equivalence classes for the configurations of type $(n-1, n-1, n-1)$ the number of possible candidates for (n, n, n) can be reduced substantially.

Lemma 5.1. *Given A and B two $n \times n$ matrices that are permutation equivalent; If C is given by the $(n+1) \times (n+1)$-matrix that is A extended one row and one column with ones, and where one non-zero element is set to zero, then there exists a matrix D that is given by the $(n+1) \times (n+1)$-matrix that is B extended one row and one column with ones, and where one non-zero element is set to zero, such that C and D are permutation equivalent.*

Proof. The proof is obvious. ∎

This leads to algorithm 5.1.

Algorithm 5.1. (Finding possible candidates for equivalence classes).

1. *Given a representative of each equivalence class of type $(n-1, n-1, n-1)$, $X_i, i = 1\ldots N$.*
2. *For each X_i construct $Y_i, j = 1\ldots N$, where Y_i is the $n \times n$-matrix that is X_i extended one row and one column with ones.*
3. *In Y_i there are $n^2-(n-1)$ non-zero elements. Y_{ij} is given by setting the j-th nonzero element of Y_i to zero, $j = 1\ldots n^2-(n-1)$.*

This gives a number of possible candidates from which representatives of the equivalence classes can be selected. Using algorithm 5.1 the number of possible candidates has been reduced from $\binom{n^2}{n}$ to $(n^2-n+1)\cdot N$. For instance if $n = 10$, then $N = 1430$ and this leads to a reduction from $1.7 \cdot 10^{13}$ to $1.3 \cdot 10^5$.

Short of trying all permutations there is no easy way of establishing if two index sets are permutation equivalent. If one is to check all possible permutations of a $n \times n$ matrix there are $(n!)^2$ such possibilities. This is rather undoable even for moderate n. In this section we propose an algorithm that uses a different approach.

If we have a set of matrices we try to permute each matrix to a standard form. This is done by solving the minimization problem stated in (9) for each Y in our set of matrices.

$$\hat{X} = \arg \min_{X \sim Y} f(X). \tag{9}$$

Here f is given by

$$f(X) = \sum_{i,j} 2^{a_{ij}} x_{ij}, \quad a_{ij} = N(i-1) + j - 1, \tag{10}$$

and x_{ij} are the entries in X. For each $n \times n$-matrix, with zero and one entries, f is injective since $f(X)$ represents the binary number with the n^2 entries of X as its digits. This means that $\hat{X}$ in (9) exists uniquely for every Y.

The problematic part of the minimization is finding the global minimum. We have used a local search method when we do the minimization, and inevitably we end up with not finding the global minimum for every matrix in our list. One way of getting away from a local minimum is to start the minimization again from a position away from the starting position. This can be done by randomly permuting the given matrix and then use the minimization routines on the new matrix. The steps are described in algorithm 5.2.

Algorithm 5.2. (Comparing matrices).

1. *Given a number of matrices* $X_i, i = 1 \dots N$
2. *For each* X_i *create* M *random permutations* $X_{ij}, j = 1 \dots M$ *of* X_i.
3. *For every* X_{ij} *find* $\hat{X}_{ij}$ *by solving* $\hat{X}_{ij} = \arg\min_{X \sim X_{ij}} f(X)$.
4. $X_i \sim X_k$ *if* $\hat{X}_{ij} = \hat{X}_{kl}$ *for some* j *and* l

Since we for every given configuration (m, n, l) can calculate the number of equivalence classes according to section 4 we can use algorithm 5.2 until we have found the right number of distinct matrices.

The motivation for the success of algorithm 5.2 is the following. Let's say that we have two matrices that are permutation equivalent, but that have ended up in different minima after our minimization, and for simplicity assume that these are the only two minima that we may end up with, from different starting points, after our minimization. If minimum one attracts a of all starting matrices and minimum two attracts $(1-a)$ then the probability that the two matrices stay in their respective minima after the use of algorithm 5.2 is $a^M(1-a)^M$. This is maximized for $a = 0.5$ with probability equal to 2^{-2M} which tends rapidly towards zero as M increases.

Table 3. The number ω_p of prime problems for different (m, n, l).

n	5	6	5	5	7	6	8
m	4	4	5	6	4	5	4
l	2	4	4	6	6	7	8
ω_p	1	3	3	6	5	22	8

Returning to our original problem of finding the prime problem configurations, we are now able to find representatives for the different primes. For a given configuration (m, n, l) the equivalence classes can be chosen from the equivalence classes for (l, l, l) where configurations with less than 3 ones in each column and less than 4 ones in each row have been removed as well as the configurations that can not be contained in a $m \times n$ matrix.

We have calculated the equivalence classes for some of the first minimal cases using the algorithms described in this section. In table 3 the number of minimal configurations for these cases are given.

In figure 1 the prime problems for the configurations of type $(5, 5, 4)$ and $(4, 6, 4)$ are given. Configurations 1a-c seem to be connected to configurations 1d-f. The similarity can be explained using a technique that Carlsson developed in [5, 6].

Theorem 5.1. *The calibrated structure and motion problem with* n *points and* m *images is equivalent to the calibrated structure and motion problem with* $m+1$ *points and* $n-1$ *images.*

The theorem is based on singling out one special point, which must be visible in all views. By using this point as a reference point (origo). It is possible to exchange the cameras for beacons and vice versa. Thus a structure and motion problem with visibility matrix $\begin{bmatrix} A_{m \times n-1} & \mathbf{0}_{m \times 1} \end{bmatrix}$ is dual to one with visibility matrix $\begin{bmatrix} A^T_{n-1 \times m} & \mathbf{0}_{n-1 \times 1} \end{bmatrix}$. Using this duality one can see that the configurations in figure 1a-c are dual to the configurations in figure 1d-f. If one has the solution to one structure and motion problem the solution to its dual problem can easily be calculated.

6. Classification of 2D retina problems

The tools of Sections 4 and 5 work equally well for ordinary vision problems. For example for the case of 3 uncalibrated views of 8 points there are 6 distinct germs. Of these 6 germs, 4 are ill-defined, and of the two remaining minimal problems one is prime, cf. [11]. For 3 views of 9 points there are no minimal configurations. For 3 views of 10 points there are 33 distinct germs (30 are ill defined, two are minimal but not prime and one is prime, see figure 2. The 2D retina problems will be studied further in a near future.

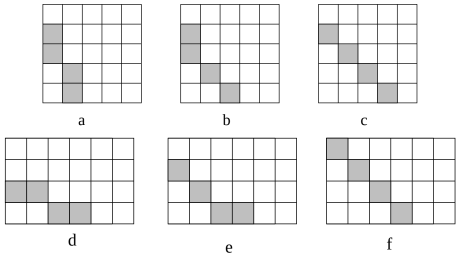

Figure 1. The 3 distinct configurations for prime problems of type $(5, 5, 4)$ (a-c) and of type $(4, 6, 4)$ (d-f).

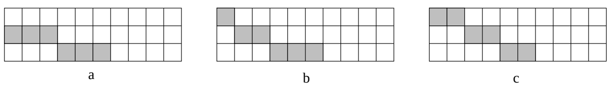

Figure 2. The 3 minimal cases of type $(3, 10, 6)$ for 2D retina. Of these (a) and (b) are minimal but not prime and (c) is prime.

7. Solution of some minimal cases

There is only one prime configuration for the case of five points in five images. This is the case where one sees five points in two images. In image three, one point is occluded and in image four another point is occluded. We will start by finding the solutions to this case.

Theorem 7.1. *The surveying problem with four views of five points*

$$\lambda_{I,J}\mathbf{u}_{I,J} = \mathbf{P}_I \mathbf{U}_J, \qquad \forall (I, J) \in \mathbb{I}\,.$$

with $\mathbb{I}$ such that point 1 is missing in view 3 and point 2 is missing in view 4 (see Table 4) has in general three solutions.

Proof. We introduce a coordinate system such that the first camera is given by

$$P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Then we can parameterize the structure with the depths in the first image,

$$\mathbf{U}_J = \begin{bmatrix} \lambda_{1,J}\cos(\alpha_{1,J}) & \lambda_{1,J}\sin(\alpha_{1,J}) & 1 \end{bmatrix}^T.$$

Using equation (3) we can write the projections in the remaining three images as

$$\mathbf{v}_{I,J} \begin{bmatrix} \lambda_{1,J}\cos(\alpha_{1,J}) & \lambda_{1,J}\sin(\alpha_{1,J}) & 1 & 0 \\ \lambda_{1,J}\sin(\alpha_{1,J}) & -\lambda_{1,J}\cos(\alpha_{1,J}) & 0 & 1 \end{bmatrix} \mathbf{p}_J = \mathbf{0}$$

If we write this as

$$M^2_{5\times 4}\mathbf{p}_2 = \mathbf{0},\ M^3_{4\times 4}\mathbf{p}_3 = \mathbf{0},\ M^4_{4\times 4}\mathbf{p}_4 = \mathbf{0}$$

we see that all 4×4-determinants of M^i have to be zero since $\mathbf{p_i} \neq \mathbf{0}$. This leads to seven two-degree polynomials in λ_{1i}. All seven are not linearly independent. Points are only determined up to scale so we can set $\lambda_{15} = 1$ in our calculations and then choose four of the polynomials and solve for $\lambda_i, \quad i = 1 \ldots 4$. If

$$\begin{aligned}
p_{11} &= a\lambda_2\lambda_3 + b\lambda_3\lambda_4 + c\lambda_4 + d\lambda_2\lambda_4 + e\lambda_3 + f\lambda_2 \\
p_{12} &= g\lambda_2\lambda_3 + h\lambda_3\lambda_4 + i\lambda_4 + j\lambda_2\lambda_4 + k\lambda_3 + l\lambda_2 \\
p_{13} &= m\lambda_1\lambda_3 + n\lambda_3\lambda_4 + o\lambda_4 + p\lambda_1\lambda_4 + q\lambda_3 + r\lambda_1 \\
p_{14} &= s\lambda_1\lambda_3 + t\lambda_3\lambda_4 + u\lambda_4 + v\lambda_1\lambda_4 + w\lambda_3 + x\lambda_1
\end{aligned}$$

then we want to solve the equations

$$p_{1i} = 0, \quad i = 1 \ldots 4 \tag{11}$$

Taking the resultant of p_1 and p_2 with respect to λ_2 and of p_3 and p_4 with respect to λ_1 gives respectively polynomials p_{21} and p_{22} of total degree three,

$$\begin{aligned}
p_{21} &= \hat{a}\lambda_3^2\lambda_4 + \hat{b}\lambda_3\lambda_4^2 + \hat{c}\lambda_3^2 + \hat{d}\lambda_4^2 + \hat{e}\lambda_3\lambda_4 + \hat{f}\lambda_3 + \hat{g}\lambda_4 \\
p_{22} &= \hat{h}\lambda_3^2\lambda_4 + \hat{i}\lambda_3\lambda_4^2 + \hat{j}\lambda_3^2 + \hat{k}\lambda_4^2 + \hat{l}\lambda_3\lambda_4 + \hat{m}\lambda_3 + \hat{n}\lambda_4
\end{aligned}$$

Taking the resultant of p_{21} and p_{22} with respect to λ_3 gives a homogeneous polynomial in λ_4 of degree seven. One of the seven solutions is created from the resultant calculations, and will not fulfill the original polynomial equations. This gives six solutions for $\lambda_{1i}, i = 1 \ldots 5$ including the trivial one. Of these, two are complex and are a result of our parameterization,

$$(\lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{14}, \lambda_{15}) \sim (e^{\pm i\alpha_{11}}, e^{\pm i\alpha_{12}}, e^{\pm i\alpha_{13}}, e^{\pm i\alpha_{14}}, e^{\pm i\alpha_{15}}).$$

This leaves three non-trivial solutions. ■

There are three prime problems for the case of five points in five images. We will now solve the surveying problem for the case of five points in five images with four measurements missing as shown in figure 1a.

Theorem 7.2. *The surveying problem for five images of five points,*

$$\lambda_{I,J}\mathbf{u}_{I,J} = \mathbf{P}_I \mathbf{U}_J, \qquad \forall (I, J) \in \mathbb{I}\,.$$

with $\mathbb{I}$ given by figure 1a has in general three solutions.

Proof. As in the previous case we can use the first image to parameterize the structure. The remaining four images are then used to solve the problem. This gives four second degree polynomial equations in $\lambda_{1i}, i = 1 \ldots 4$. These

Table 4. Some bearing measurements

0.6929	-0.7825	-1.9347	0.3263	-0.6421
0.3206	-0.9479	-1.8732	-0.0041	-0.8289
—	-2.5202	2.4474	-0.9746	-2.3323
2.3024	—	-1.0540	1.8991	0.6499

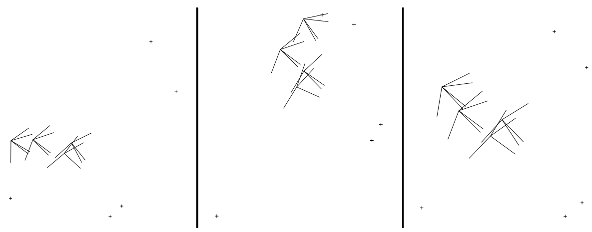

Figure 3. Three solutions to the minimal case of five points in four images. Beacons are indicated by '+'.

polynomials have the exact same structure as those in the case of five points in four images, and hence the solutions have the same structure. This leads to that the problem of five points in five views has three non-trivial solutions. ■

The dual to the case of five points in five images is the case of six points in four images. This means that there are three solutions to the case of six points in four images given by figure 1d.

Corollary 7.1. *The surveying problem for four images of six points,*

$$\lambda_{I,J}\mathbf{u}_{I,J} = \mathbf{P}_I\mathbf{U}_J, \qquad \forall (I,J) \in \mathbb{I}\,.$$

with $\mathbb{I}$ *given by figure 1d has in general three solutions.*

The methods described in the proof of theorem 7.1 can easily be implemented. In table 4 bearings for an example of the minimal case described in theorem 7.1 is shown. The resulting solutions are given in figure 3. In this case there were three real solutions with all depths positive.

8. Conclusions

In this paper we have begun to classify and solve structure and motion problems for calibrated 1D retina vision with missing data. We have introduced a notation on so called prime structure and motion problems. These are the problems that if solved will allow solutions to all structure and motion problems. Similar to the prime numbers it seems that there are infinitely many such prime problems. In the paper we have given methods for calculating the number of such problems with a given size and also methods for finding representatives of each such problem. We have also begun our work on actually designing algorithms that solve the structure and motion problems for some of these instances. More work is however needed in order to understand and solve these problems. We hope to be able to give more results in this direction in a near future.

References

[1] M. Armstrong, A. Zisserman, and R. Hartley. Self-calibration from image triplets. In *Proc. 4th European Conf. on Computer Vision, Cambridge, UK*, pages 3–16. Springer-Verlag, 1996.

[2] K. Åström. Multilinear constraints in two-dimensional vision and isogonal conjugacy. In *Proc. Symposium on Image Analysis, Lund, Sweden*, pages 34–38, 1996.

[3] K. Åström, A. Heyden, F. Kahl, and M. Oskarsson. Structure and motion from lines under affine projections. In *Proc. 7th Int. Conf. on Computer Vision, Kerkyra, Greece*, 1999.

[4] K. Åström and M. Oskarsson. Solutions and ambiguities of the structure and motion problem for 1d retinal vision. *Journal of Mathematical Imaging and Vision*, 12:121–135, 2000.

[5] S. Carlsson. Duality of reconstruction and positioning from projective views. In *IEEE Workshop on Representation of Visual Scenes*. IEEE, 1995.

[6] S. Carlsson and D. Weinshall. Dual computation of projective shape and camera positions from multiple images. *Int. Journal of Computer Vision*, 27(3):227–241, 1998.

[7] O. D. Faugeras, L. Quan, and P. Sturm. Self-calibration of a 1d projective camera and its application to the self-calibration of a 2d projective camera. In *Proc. 5th European Conf. on Computer Vision, Freiburg, Germany*, pages 36–52. Springer-Verlag, 1998.

[8] J.B. Fraleigh. *A first course in abstract algebra*. Addison-Wesley, 5th edition, 1994.

[9] R Gupta and R. I. Hartley. Linear pushbroom cameras. *Pattern Analysis and Machine Intelligence*, 19(9):963–975, sep 1997.

[10] K. Hyyppä. Optical navigation system using passive identical beacons. In *Proceedings intelligent autonomous systems, Amsterdam*, 1987.

[11] F. Kahl, A. Heyden, and L. Quan. Projective reconstruction from minimal missing data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(4), 2001.

[12] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(8):834–845, August 1997.

[13] B. Triggs. Matching constraints and the joint image. In *Proc. ICCV'95*, pages 338–343. IEEE Computer Society Press, 1995.

[14] W. Wei and J. Xu. Cycle index of direct product of permutation groups and number of equivalence classes of subsets of z_ν. *Discrete Mathematics*, 123:179–188, 1993.